SIAM J. OPTIM.
Vol. 30, No. 3, pp. 2197–2220
© 2020 Society for Industrial and Applied Mathematics

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

A PROXIMAL POINT DUAL NEWTON ALGORITHM FOR SOLVING GROUP GRAPHICAL LASSO PROBLEMS*

YANGJING ZHANG†, NING ZHANG‡, DEFENG SUN§, AND KIM-CHUAN TOH¶

Abstract. Undirected graphical models have been especially popular for learning the conditional independence structure among a large number of variables where the observations are drawn independently and identically from the same distribution. However, many modern statistical problems would involve categorical data or time-varying data, which might follow different but related underlying distributions. In order to learn a collection of related graphical models simultaneously, various joint graphical models inducing sparsity in graphs and similarity across graphs have been proposed. In this paper, we aim to propose an implementable proximal point dual Newton algorithm (PPDNA) for solving the group graphical Lasso model, which encourages a shared pattern of sparsity across graphs. Though the group graphical Lasso regularizer is nonpolyhedral, the asymptotic superlinear convergence of our proposed method PPDNA can be obtained by leveraging on the local Lipschitz continuity of the Karush–Kuhn–Tucker solution mapping associated with the group graphical Lasso model. A variety of numerical experiments on real data sets illustrates that the PPDNA for solving the group graphical Lasso model can be highly efficient and robust.

Key words. group graphical Lasso, proximal point algorithm, semismooth Newton method, Lipschitz continuity

AMS subject classifications. 90C22, 90C25, 90C31, 62J10

DOI. 10.1137/19M1267830

1. Introduction.

Let $w^{(k)} \in \mathbb{R}^{n_k \times p}$, $k = 1, 2, \ldots, K$, be K given data matrices. For each $k = 1, 2, \ldots, K$, the rows of $w^{(k)}$ are observations drawn independently from a Gaussian distribution with mean zero, and the empirical covariance matrix for $w^{(k)}$ is given by $S^{(k)} = (1/n_k)(w^{(k)})^T w^{(k)}$. In this paper, we consider the following joint graphical model:

$$\min_{\Theta} \ \sum_{k=1}^{K} \Big(-\log\det \Theta^{(k)} + \langle S^{(k)}, \Theta^{(k)} \rangle \Big) + \mathcal{P}(\Theta), \tag{1.1}$$

where $\Theta = \big(\Theta^{(1)}, \Theta^{(2)}, \ldots, \Theta^{(K)}\big) \in \mathbb{S}^p \times \mathbb{S}^p \times \cdots \times \mathbb{S}^p$ is the decision variable, and $\mathcal{P}$ is a convex penalty term that can promote certain desired structure in the decision variable Θ. Throughout this paper, we assume that the solution set to problem (1.1) is nonempty.

*Received by the editors June 11, 2019; accepted for publication (in revised form) May 18, 2020; published electronically August 12, 2020.

https://doi.org/10.1137/19M1267830

Funding: The research of the second author was supported in part by the National Natural Science Foundation of China under grant 11901083. The research of the third author was supported in part by Hong Kong Research Grant Council under grant PolyU 153014/18P. The research of the fourth author was supported in part by the Academic Research Fund of the Ministry of Education of Singapore under grant R-146-000-257-112.

†Department of Mathematics, National University of Singapore, 119076, Singapore (zhangyangjing@u.nus.edu).

‡Corresponding author. School of Computer Science and Technology, Dongguan University of Technology, Dongguan 523808, China (ningzhang_2008@yeah.net).

§Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong, China (defeng.sun@polyu.edu.hk).

¶Department of Mathematics, and Institute of Operations Research and Analytics, National University of Singapore, 119076, Singapore (mattohkc@nus.edu.sg).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

If $K = 1$ and $\mathcal{P}(\cdot) = \lambda\|\cdot\|_1$, problem (1.1) reduces to the well-known sparse Gaussian graphical model which has been studied by various researchers (e.g., [1, 2, 10, 14, 27, 33, 36]). In many applications, a single Gaussian graphical model is typically enough to capture the conditional independence structure of the random variables. However, in some situations it is more reasonable to fit a collection of such models jointly, due to the similarity or heterogeneity of the data involved. These models for estimating multiple precision matrices jointly are referred to as joint graphical models in [8]. A scenario where joint graphical models are more suitable than a single graphical model is when the data comes from several distinct but closely related classes, which share the same collection of variables but differ in terms of the dependency structures. Their dependency graphs can have common edges across a portion of all classes and unique edges restricted to only certain classes. In this case, fitting separate graphical models for distinct classes does not exploit the similarity among the dependency graphs. In contrast, joint estimation of these models could exploit information across different but related classes. In addition to the data from different classes, another scenario that would favor joint graphical models over a single graphical model is when the data contains sequences of multivariate time-stamped observations. Such data might correspond to a series of dependency graphs over time. Next, we give two practical applications of joint graphical models, which will also be used in our numerical experiments:

- The inference of words' relationships from webpages or newsgroups: the webpages from the computer science departments of various universities are classified into several classes: Student, Faculty, Course, Project, etc. The 20 newsgroups are grouped into various topics.
- The inference of time-varying dependency structures of stocks: the dependency structures among the Standard & Poor's 500 component stocks might change smoothly over time.

In summary, there are two major applications of the joint graphical models: (i) estimating multiple precision matrices jointly for a collection of variables across distinct classes; (ii) inferring the time-varying networks and finding the change-points.

For solving problem (1.1) with different forms of penalty terms, the alternating direction method of multipliers (ADMM) has been extensively used; see, e.g., [8, 12, 13]. As we know, the ADMM could be a fast first order method for finding approximate solutions of low or moderate accuracy. However, for attaining superlinear convergence to compute highly accurate solutions, one has to incorporate at least in part the second order information of the problem. Yang et al. [34] proposed a proximal Newton-type method, where the subproblem in each iteration can be solved by the nonmonotone spectral projected gradient method [19, 32], and an active set identification scheme was applied to reduce the cost. Another notable contribution is that a screening rule, which can be combined with any method to reduce the computational cost, was proposed in [34]. However, the second order method in [34] is not without drawbacks. Each of its subproblems is a complicated quadratic approximation problem, which generally requires expensive computations. Besides, the inexact proximal Newton-type method proposed in [34] has no guarantee of local linear convergence. It is worth noting that, in a recent paper related to [34], Yue, Zhou, and So [37] studied the local convergence rate of a family of inexact proximal Newton-type methods for solving a class of nonsmooth convex composite optimization problems based on an error bound condition. However, it is not clear to us whether the convergence analysis in [37] can be directly applied to problem (1.1), as the Hessian

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

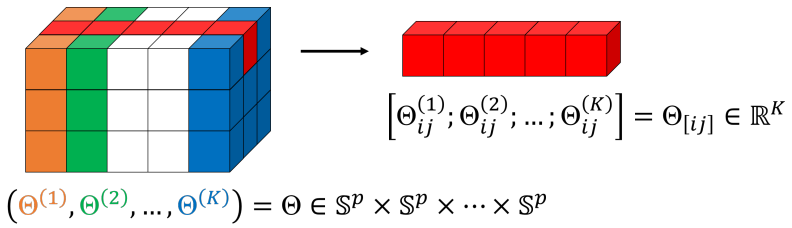


FIG. 1. *Illustration of* Θ *and* $\Theta_{[ij]}$. *One cube stands for one entry.*

of the first function in the objective of the problem is not uniformly bounded on its effective domain. More recently, Zhang et al. [38] applied a regularized proximal point algorithm (rPPA) to solve a fused multiple graphical Lasso (FGL) model and heavily exploited the underlying second order information through the semismooth Newton method when solving the subproblems of the rPPA. Due to the polyhedral property of the FGL regularizer, the rPPA for solving the FGL problem is proven to have an arbitrary linear convergence rate in [38].

Our goal in this paper is to design and analyze an efficient second order information based algorithm with economical implementations and a fast convergence rate for solving problem (1.1) with the following nonpolyhedral regularizer, which was referred to as the group graphical Lasso (GGL) regularizer in [8]:

$$\mathcal{P}(\Theta) = \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\Theta_{ij}^{(k)}| + \lambda_2 \sum_{i \neq j} \Big(\sum_{k=1}^{K} |\Theta_{ij}^{(k)}|^2 \Big)^{1/2}, \tag{1.2}$$

where λ_1 and λ_2 are positive parameters. We refer to model (1.1) with the regularizer (1.2) as the GGL model. In fact, the GGL regularizer acting on a collection of matrices can be viewed as an extension of the sparse group Lasso regularizer [11, 28] acting on a vector. The former can be regarded as the latter if the (i,j)th elements across all K precision matrices are assigned into one group. For $1 \le i, j \le p$, we let $\Theta_{[ij]} := [\Theta_{ij}^{(1)}; \ldots; \Theta_{ij}^{(K)}] \in \mathbb{R}^K$ be the column vector obtained by taking out the (i,j)th elements across all K matrices $\Theta^{(k)}$, $k = 1, 2, \ldots, K$. We can observe that

$$\mathcal{P}(\Theta) = \sum_{i \neq j} \varphi(\Theta_{[ij]}) \text{ with } \varphi(x) = \lambda_1 \|x\|_1 + \lambda_2 \|x\| \ \forall\, x \in \mathbb{R}^K, \tag{1.3}$$

where the function φ is actually a special sparse group Lasso regularizer. The first term of the GGL regularizer promotes sparsity in the K estimated precision matrices $\Theta^{(k)}$'s. The zeros in these precision matrices tend to occur at the same indices due to the second term of the GGL regularizer. In addition, Figure 1 illustrates the structure of the decision variable Θ and the vector belonging to one group $\Theta_{[ij]}$.

Inspired by the impressive numerical performance of the rPPA for solving the FGL model [38], we will design a proximal point dual Newton algorithm (PPDNA) for solving the GGL model. Specifically, a proximal point algorithm (PPA) [24] is applied to the primal formulation of the GGL model, and a superlinearly convergent semismooth Newton method is designed to solve the dual formulations of the PPA subproblems. Thanks to the fact that the GGL regularizer is an extension of the sparse group Lasso regularizer, the generalized Jacobian of the proximal mapping of the GGL regularizer can be characterized based on that of the sparse group Lasso regularizer, where the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

explicit form was given in [39]. As a result, the former naturally inherits the structured sparsity (referred to as the second order sparsity) of the latter. Consequently, multiplying a sparse Hessian matrix by a vector in the semismooth Newton method is reasonably cheap, and one could expect that the superlinearly convergent semismooth Newton method is numerically efficient for solving the PPA subproblems. In addition to achieving low cost in computing the semismooth Newton directions by exploiting the second order sparsity, we also establish the linear convergence guarantee of the PPDNA.

Though the framework of the PPDNA for solving the GGL model is closely related to the rPPA for solving the FGL model [38] and the semismooth Newton based augmented Lagrangian method (SSNAL) for solving the sparse group Lasso problems [39], both the theoretical analysis and numerical implementation should be further investigated owing to the following difficulties of the GGL model. First, unlike the FGL regularizer, the GGL regularizer is a nonpolyhedral function, and consequently the Lipschitz continuity of the Karush–Kuhn–Tucker (KKT) solution mapping associated with the GGL model is not as straightforward to establish as in [38]. We should mention here that the Lipschitz continuity of the KKT solution mapping plays an important role in establishing the convergence rate of the PPDNA, just as in the case of rPPA and SSNAL. Second, the subproblem of the PPDNA for solving the GGL model differs from those of the SSNAL and rPPA, which are strongly convex. Therefore, the stopping criteria previously used in SSNAL and rPPA are no longer applicable. The main contributions of this paper can be summarized as follows.

1. We prove the Lipschitz continuity of the KKT solution mapping associated with the GGL model, by taking advantage of the strict convexity of the function $-\log\det(\cdot)$ in its effective domain, the nonsingularity of its Jacobian, and Clarke's implicit function theorem [5, 6]. Consequently, the linear convergence of the iterative sequence generated by the PPDNA can be established based on the classical results in [24]. Moreover, by choosing the penalty parameter to be sufficiently large, the PPDNA can be made to attain any desired linear convergence rate. More generally, the Lipschitz continuity of the KKT solution mapping of the model still holds even if the GGL regularizer is replaced by any other convex positively homogeneous function.
2. We derive a surrogate generalized Jacobian of the proximal mapping of the GGL regularizer. The second order sparsity in the surrogate generalized Jacobian is analyzed in depth and fully exploited in the PPDNA. Therefore, the superlinearly (or even quadratically) convergent semismooth Newton method can solve the PPA subproblems very efficiently since the semismooth Newton directions can be computed cheaply.
3. We introduce fairly easy-to-check stopping criteria (via the duality theory) for computing inexact solutions of the PPA subproblems without sacrificing the global or linear convergence of the PPDNA. In fact, the standard stopping criteria adopted by Rockafellar [24] would involve the unknown optimal values of the subproblems, which are not easy to check unless the objective function is strongly convex with an explicitly given strong convexity parameter.

The remaining parts of the paper are organized as follows. Section 2 presents some definitions and preliminary results, which include the proximal mapping of the GGL regularizer, its generalized Jacobian, the proximal mapping of the log-determinant function, and its derivative. We analyze in section 3 the Lipschitz continuity of the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

KKT solution mapping associated with the GGL model, which is the key property for deriving the linear convergence rate of our proposed algorithm. In section 4, we propose the PPDNA for solving the GGL model and investigate its convergence properties. We report the numerical performance of the PPDNA on categorical text data and time-varying stock prices data in section 5 and conclude the paper in section 6.

Notation. The following notation will be used in the rest of the paper.

- $\mathbb{S}^p_+$ ($\mathbb{S}^p_{++}$) denotes the cone of positive semidefinite (definite) matrices in the space of $p \times p$ real symmetric matrices $\mathbb{S}^p$. For any $A, B \in \mathbb{S}^p$, we write $A \succeq B$ if $A - B \in \mathbb{S}^p_+$ and $A \succ B$ if $A - B \in \mathbb{S}^p_{++}$. In particular, $A \succeq 0$ ($A \succ 0$) indicates that $A \in \mathbb{S}^p_+$ ($A \in \mathbb{S}^p_{++}$).
- We let $\mathbb{Z}$ ($\mathbb{Z}_+$, $\mathbb{Z}_{++}$) be the Cartesian product of K copies of $\mathbb{S}^p$ ($\mathbb{S}^p_+$, $\mathbb{S}^p_{++}$).
- For any matrix A, A_{ij} denotes the (i,j)th element of A.
- For any $X := (X^{(1)}, \ldots, X^{(K)}) \in \mathbb{Z}$, $X_{[ij]} := [X^{(1)}_{ij}; \ldots; X^{(K)}_{ij}] \in \mathbb{R}^K$ denotes the column vector obtained by taking out the (i,j)th elements across all K matrices $X^{(k)}$, $k = 1, 2, \ldots, K$.
- I_n denotes the $n \times n$ identity matrix, and I denotes an identity matrix or map when the dimension is clear from the context.
- We use $\lambda_{\max}(\mathcal{A})$ to denote the largest eigenvalue of a self-adjoint linear operator $\mathcal{A}$.
- For a given closed convex set Ω and a vector x, we denote the Euclidean projection of x onto Ω by $\Pi_\Omega(x) := \arg\min_{x' \in \Omega}\{\|x - x'\|\}$.
- We denote ceil(x) as the smallest integer greater than or equal to $x \in \mathbb{R}$.

2. Preliminaries. In this section, we first recall the definition and some relevant properties of the Moreau–Yosida regularization of a proper and closed convex function, which will play an important role in the subsequent theoretical analysis and algorithmic design. Let $\mathcal{E}$ be a finite dimensional real Hilbert space and $g : \mathcal{E} \to \mathbb{R} \cup \{+\infty\}$ be a proper and closed convex function. The Moreau–Yosida regularization [21, 35] of g is defined by

$$\Psi_g(u) := \min_{u'} \left\{ g(u') + \frac{1}{2}\|u' - u\|^2 \right\} \ \forall\, u \in \mathcal{E}, \tag{2.1}$$

and the proximal mapping of g, the unique minimizer of (2.1), is given by

$$\mathrm{Prox}_g(u) := \arg\min_{u'} \left\{ g(u') + \frac{1}{2}\|u' - u\|^2 \right\} \ \forall\, u \in \mathcal{E}.$$

One critical property of the Moreau–Yosida regularization is that $\Psi_g(\cdot)$ is a continuously differentiable convex function with the following gradient:

$$\nabla \Psi_g(u) = u - \mathrm{Prox}_g(u) \ \forall\, u \in \mathcal{E}.$$

In addition, the proximal mapping satisfies the following Moreau identity [25, Theorem 31.5]:

$$\mathrm{Prox}_{\sigma g}(u) + \sigma \mathrm{Prox}_{\sigma^{-1} g^*}(u/\sigma) = u \ \forall\, u \in \mathcal{E},\ \sigma > 0, \tag{2.2}$$

where g^* is the conjugate function of g (see, e.g., [25] for its definition).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

2.1. Proximal mapping of the GGL regularizer and its generalized Jacobian. We investigate in this section the proximal mapping of the GGL regularizer $\mathcal{P}$ in (1.2) and its generalized Jacobian. Recall the function in (1.3):

$$\mathcal{P}(\Theta) = \sum_{i\neq j} \varphi(\Theta_{[ij]}) \text{ with } \varphi(x) = \lambda_1\|x\|_1 + \lambda_2\|x\| \ \forall\, x \in \mathbb{R}^K.$$

By definition, the proximal mapping of $\mathcal{P}$ is given as follows: for any $X \in \mathbb{Z}$,

$$\begin{aligned} &\mathrm{Prox}_{\mathcal{P}}(X) = \arg\min_{\Theta\in\mathbb{Z}} \left\{\mathcal{P}(\Theta) + \tfrac{1}{2}\|\Theta - X\|^2\right\} \\ &= \arg\min_{\Theta\in\mathbb{Z}} \left\{\textstyle\sum_{i\neq j}\left\{\varphi(\Theta_{[ij]}) + \frac{1}{2}\|\Theta_{[ij]} - X_{[ij]}\|^2\right\} + \frac{1}{2}\sum_i \|\Theta_{[ii]} - X_{[ii]}\|^2\right\}. \end{aligned} \tag{2.3}$$

It is obvious that problem (2.3) is separable for each vector $\Theta_{[ij]} \in \mathbb{R}^K$. Therefore, for any $i, j \in \{1, 2, \ldots, p\}$, the vector $(\mathrm{Prox}_{\mathcal{P}}(X))_{[ij]}$, consisting of all entries of $\mathrm{Prox}_{\mathcal{P}}(X)$ in the (i, j)th position, is given explicitly by

$$(\mathrm{Prox}_{\mathcal{P}}(X))_{[ij]} = \begin{cases} \mathrm{Prox}_{\varphi}(X_{[ij]}) & \text{if } \ i \neq j, \\ X_{[ii]} & \text{if } \ i = j. \end{cases} \tag{2.4}$$

By this equation, one can compute $\mathrm{Prox}_{\mathcal{P}}$ via performing $p(p-1)/2$ computations of Prox_{φ}, and this task can be done in parallel. Parts of the second order information of the underlying problem are contained in the generalized Jacobian of $\mathrm{Prox}_{\mathcal{P}}$, which can be characterized by the generalized Jacobian of Prox_{φ} through using the relationship (2.4) between $\mathrm{Prox}_{\mathcal{P}}$ and Prox_{φ}. Fortunately, the generalized Jacobian of Prox_{φ} has been carefully investigated in [39] and has an explicit expression.

Let the multifunction $\widehat{\partial}\mathrm{Prox}_{\varphi} : \mathbb{R}^K \rightrightarrows \mathbb{S}^K$ be the generalized Jacobian of Prox_{φ}. Directly from the formula (10) in [39], the multifunction $\widehat{\partial}\mathrm{Prox}_{\varphi}$ can be described as follows: for any $u \in \mathbb{R}^K$,

$$\begin{aligned} &\widehat{\partial}\mathrm{Prox}_{\varphi}(u) \\ &= \left\{(I - \Sigma)\Lambda \in \mathbb{S}^K \,\middle|\, v = \mathrm{Prox}_{\lambda_1\|\cdot\|_1}(u),\ \Sigma \in \partial\Pi_{\mathbb{B}_{\lambda_2}}(v),\ \Lambda \in \partial\mathrm{Prox}_{\lambda_1\|\cdot\|_1}(u)\right\}, \end{aligned} \tag{2.5}$$

where $\mathbb{B}_{\lambda_2} := \{v \in \mathbb{R}^K \,|\, \|v\| \leq \lambda_2\}$, and $\partial\Pi_{\mathbb{B}_{\lambda_2}}$ and $\partial\mathrm{Prox}_{\lambda_1\|\cdot\|_1}$ are the Clarke generalized Jacobians (see [5, Definition 2.6.1] for the definition) of $\Pi_{\mathbb{B}_{\lambda_2}}$ and $\mathrm{Prox}_{\lambda_1\|\cdot\|_1}$, respectively. Therefore, the surrogate generalized Jacobian $\widehat{\partial}\mathrm{Prox}_{\mathcal{P}}(X) : \mathbb{Z} \rightrightarrows \mathbb{Z}$ of $\mathrm{Prox}_{\mathcal{P}}$ at any given X can be described as follows:

$$\begin{cases} \mathcal{W} \in \widehat{\partial}\mathrm{Prox}_{\mathcal{P}}(X) \text{ if and only if } \exists\ M^{(ij)} \in \widehat{\partial}\mathrm{Prox}_{\varphi}(X_{[ij]})\ \forall\, i < j, \\ \text{such that } (\mathcal{W}[Y])_{[ij]} = \begin{cases} M^{(ij)}Y_{[ij]} & \text{if } i < j, \\ Y_{[ii]} & \text{if } i = j, \quad i, j = 1, \ldots, p, \ \ \forall\, Y \in \mathbb{Z}. \\ M^{(ji)}Y_{[ij]} & \text{if } j < i, \end{cases} \end{cases} \tag{2.6}$$

The next proposition will explain why $\widehat{\partial}\mathrm{Prox}_{\mathcal{P}}(X)$ in (2.6) can be treated as the surrogate generalized Jacobian of $\mathrm{Prox}_{\mathcal{P}}$ at X. Based on [39, Theorem 3.1], one can easily prove the proposition. We omit the details here.

PROPOSITION 2.1. *Let $\mathcal{P}$ be the GGL regularizer defined by* (1.2) *and $X \in \mathbb{Z}$ be any given element. The surrogate generalized Jacobian $\widehat{\partial}\mathrm{Prox}_{\mathcal{P}}(\cdot)$ defined in* (2.6)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

is nonempty compact valued and upper semicontinuous. Any element in the set $\widehat{\partial}\mathrm{Prox}_{\mathcal{P}}(X)$ *is a self-adjoint and positive semidefinite operator. Moreover, we have that, for any* $Y \to X$,

$$\mathrm{Prox}_{\mathcal{P}}(Y) - \mathrm{Prox}_{\mathcal{P}}(X) - \mathcal{W}[Y - X] = O(||Y - X||^2) \; \forall \, \mathcal{W} \in \widehat{\partial}\mathrm{Prox}_{\mathcal{P}}(Y).$$

2.2. Properties of the log-determinant function. In this subsection, we present some properties on the proximal mapping of the following log-determinant function h and its derivative that are mainly adopted from the papers [31, 33]:

$$h(X) := \begin{cases} -\log\det X & \text{if } X \in \mathbb{S}^p_{++}, \\ +\infty & \text{otherwise.} \end{cases} \tag{2.7}$$

Let $\beta > 0$ be given. Define the following scalar functions:

$$\phi^+_\beta(x) := (\sqrt{x^2 + 4\beta} + x)/2, \;\; \phi^-_\beta(x) := (\sqrt{x^2 + 4\beta} - x)/2 \; \forall \, x \in \mathbb{R}.$$

In addition, for any $A \in \mathbb{S}^p$ with eigenvalue decomposition $A = Q\mathrm{Diag}(d_1, \ldots, d_p)Q^T$, we define

$$\phi^+_\beta(A) := Q\mathrm{Diag}(\phi^+_\beta(d_1), \ldots, \phi^+_\beta(d_p))Q^T, \;\; \phi^-_\beta(A) := Q\mathrm{Diag}(\phi^-_\beta(d_1), \ldots, \phi^-_\beta(d_p))Q^T.$$

One can observe that $\phi^+_\beta(A)$ and $\phi^-_\beta(A)$ are positive definite for any $A \in \mathbb{S}^p$. Using the functions defined above, the following two propositions give the proximal mapping of the log-determinant function h and its derivative.

PROPOSITION 2.2 ([33, Proposition 2.3]). *Let* h *be the log-determinant function defined by* (2.7) *and* β *be a positive scalar. Then, for any* $A \in \mathbb{S}^p$, *it holds that*

$$\phi^+_\beta(A) = \mathrm{Prox}_{\beta h}(A) = \underset{B \in \mathbb{S}^p_{++}}{\arg\min} \big\{ h(B) + \tfrac{1}{2\beta}\|B - A\|^2 \big\},$$
$$\Psi_{\beta h}(A) = \min_{B \in \mathbb{S}^p_{++}} \big\{ \beta h(B) + \tfrac{1}{2}\|B - A\|^2 \big\} = -\beta \log\det(\phi^+_\beta(A)) + \tfrac{1}{2}\|\phi^-_\beta(A)\|^2.$$

PROPOSITION 2.3 ([31, Lemma 2.1 (b)]). *Let* β *be a given positive scalar. The function* $\phi^+_\beta : \mathbb{S}^p \to \mathbb{S}^p$ *is continuously differentiable. For any* $A \in \mathbb{S}^p$ *with eigenvalue decomposition* $A = Q\mathrm{Diag}(d_1, \ldots, d_p)Q^T$, *the derivative* $(\phi^+_\beta)'(A)[B]$ *at any* $B \in \mathbb{S}^p$ *is given by*

$$(\phi^+_\beta)'(A)[B] = Q(\Gamma \odot (Q^T B Q))Q^T,$$

where $\Gamma \in \mathbb{S}^p$ *is defined by*

$$\Gamma_{ij} = \frac{\phi^+_\beta(d_i) + \phi^+_\beta(d_j)}{(d_i^2 + 4\beta)^{1/2} + (d_j^2 + 4\beta)^{1/2}}, \;\; i, j = 1, 2, \ldots, p.$$

3. Lipschitz continuity of the KKT solution mapping. In this section, we will prove that the KKT solution mapping associated with the GGL problem is Lipschitz continuous. More generally, we emphasize that the Lipschitz continuity of the KKT solution mapping still holds even if the GGL regularizer is replaced by any other convex positively homogeneous function, since the key properties we need from the regularizer $\mathcal{P}$ are convexity and positive homogeneity.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

The analysis in this section is based on Clarke's implicit function theorem. For notational convenience, we denote

$$f(\Theta) := \sum_{k=1}^{K} h(\Theta^{(k)}) + \langle S^{(k)}, \Theta^{(k)} \rangle, \ \ \Theta \in \mathbb{Z}. \tag{3.1}$$

Then the GGL problem (1.1) can be equivalently reformulated as follows:

$$\min_{\Omega,\Theta} \ \ \{ f(\Omega) + \mathcal{P}(\Theta) \,|\, \Omega - \Theta = 0 \}. \tag{3.2}$$

The Lagrangian function associated with problem (3.2) is given by

$$\mathcal{L}(\Omega, \Theta, X) := f(\Omega) + \mathcal{P}(\Theta) + \langle \Omega - \Theta, X \rangle, \ \ (\Omega, \Theta, X) \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z},$$

and the dual problem of (3.2) is easily shown to be

$$\max_{X} \ \sum_{k=1}^{K} \big(\log\det(X^{(k)} + S^{(k)}) + p \big) - \mathcal{P}^*(X). \tag{3.3}$$

In addition, the KKT system associated with (3.2) and (3.3) is given by

$$X + \mathrm{Prox}_{f^*}(\Omega - X) = 0, \ -X + \mathrm{Prox}_{\mathcal{P}^*}(\Theta + X) = 0, \ \Omega - \Theta = 0. \tag{3.4}$$

Since the log-determinant function h is strictly convex and the solution set to problem (1.1) is assumed to be nonempty, problem (3.2) has a unique solution. Furthermore, by using [3, Proposition 4.75] one can easily show that the KKT system (3.4) also has a unique solution, denoted by $(\overline{\Omega}, \overline{\Theta}, \overline{X})$.

For any given $(U, V, W) \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$, we consider the following linearly perturbed form of problem (3.2):

$$\begin{array}{ll} \min\limits_{\Omega,\Theta} & f(\Omega) + \mathcal{P}(\Theta) - \langle (U, V), (\Omega, \Theta) \rangle \\ \text{s.t.} & \Omega - \Theta + W = 0. \end{array} \tag{3.5}$$

As in Rockafellar [23], we define the following maximal monotone operator:

$$\begin{aligned} &\mathcal{T}_{\mathcal{L}}(\Omega, \Theta, X) \\ &:= \{ (U, V, W) \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \,|\, (U, V, -W) \in \partial \mathcal{L}(\Omega, \Theta, X) \}, \ \ (\Omega, \Theta, X) \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}. \end{aligned}$$

We also define the KKT solution mapping $\mathcal{S} : \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ as

$$\mathcal{S}(U, V, W) := \mathcal{T}_{\mathcal{L}}^{-1}(U, V, W) = \text{the set of all KKT points for problem (3.5)}. \tag{3.6}$$

Define a mapping $\mathcal{H} : (\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}) \times (\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}) \to \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ as follows: for any $(U, V, W) \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ and $(\Omega, \Theta, X) \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$,

$$\mathcal{H}((U, V, W), (\Omega, \Theta, X)) = \begin{pmatrix} X - U + \mathrm{Prox}_{f^*}(\Omega - X + U) \\ -X - V + \mathrm{Prox}_{\mathcal{P}^*}(\Theta + X + V) \\ \Omega - \Theta + W \end{pmatrix}. \tag{3.7}$$

Then it is easy to see that if $\mathcal{S}(U, V, W)$ is nonempty, then it must be a singleton and it satisfies $\mathcal{H}((U, V, W), \mathcal{S}(U, V, W)) = 0$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

LEMMA 3.1. *Let $Z \in \mathbb{Z}$ and f be defined by (3.1). Then all $\mathcal{G}_f \in \partial \mathrm{Prox}_f(Z)$ and $\mathcal{G}_{f^*} \in \partial \mathrm{Prox}_{f^*}(Z)$ are self-adjoint and positive definite with $\lambda_{\max}(\mathcal{G}_f) < 1$ and $\lambda_{\max}(\mathcal{G}_{f^*}) < 1$.*

Proof. The proof can be derived from [31, Lemma 2.1]. ∎

Since the GGL regularizer $\mathcal{P}$ defined by (1.2) is positively homogeneous, its conjugate function $\mathcal{P}^*$ is an indicator function of a closed convex set [26, Example 11.4(a)]. Therefore, $\mathrm{Prox}_{\mathcal{P}^*}$ is the projection onto a closed convex set. We know further from [30, Theorem 2.3] that for any $Y \in \mathbb{Z}$, any element in $\partial \mathrm{Prox}_{\mathcal{P}^*}(Y)$ is a self-adjoint operator whose eigenvalues are in the interval $[0, 1]$. Thus, by the proof of [30, Theorem 2.5], we can obtain the following lemma, which will be used in Theorem 3.3 to analyze the Lipschitz continuity of the KKT solution mapping $\mathcal{S}$ defined by (3.6).

LEMMA 3.2. *Let $Y \in \mathbb{Z}$ and $\mathcal{B} : \mathbb{Z} \to \mathbb{Z}$ be any self-adjoint positive definite operator. Then, for any chosen $\mathcal{G}_{\mathcal{P}^*} \in \partial \mathrm{Prox}_{\mathcal{P}^*}(Y)$, the linear operator $I - \mathcal{G}_{\mathcal{P}^*} + \mathcal{G}_{\mathcal{P}^*}\mathcal{B}$ is nonsingular.*

The next theorem will play an essential role in establishing the linear rate of convergence of our proposed proximal point dual Newton algorithm (PPDNA) for solving the GGL problems in section 4.3.

THEOREM 3.3. *Let $\mathcal{S} : \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ be the KKT solution mapping defined by (3.6). Then the following hold:*

(a) *Any element in $\partial_{(\Omega,\Theta,X)}\mathcal{H}((0,0,0),(\overline{\Omega},\overline{\Theta},\overline{X}))$ is nonsingular. Here, we say $\mathcal{G} \in \partial_{(\Omega,\Theta,X)}\mathcal{H}((0,0,0),(\overline{\Omega},\overline{\Theta},\overline{X}))$ if for some linear operator $\mathcal{M}$, it holds that $(\mathcal{M},\mathcal{G}) \in \partial\mathcal{H}((0,0,0),(\overline{\Omega},\overline{\Theta},\overline{X}))$.*

(b) *The mapping $\mathcal{S}$ is Lipschitz continuous near the origin; i.e., there exist a neighborhood $\mathcal{N}$ of the origin and a positive scalar κ such that $\mathcal{S}(U,V,W) \neq \emptyset$ for any $(U,V,W) \in \mathcal{N}$ and*

$$\begin{aligned} &\|\mathcal{S}(U,V,W) - \mathcal{S}(U',V',W')\| \\ &\leq \kappa\|(U,V,W) - (U',V',W')\| \ \forall\, (U,V,W),(U',V',W') \in \mathcal{N}. \end{aligned} \tag{3.8}$$

Proof. Since $\mathrm{Prox}_{\mathcal{P}}$ is directionally differentiable, we know from the Moreau identity (2.2) that $\mathrm{Prox}_{\mathcal{P}^*}$ is also directionally differentiable. Therefore, it follows from the chain rule presented in [29, Lemma 2.1] that for any $\mathcal{G} \in \partial_{(\Omega,\Theta,X)}\mathcal{H}((0,0,0),(\overline{\Omega},\overline{\Theta},\overline{X}))$, there exist $\mathcal{G}_{f^*} \in \partial\mathrm{Prox}_{f^*}(\overline{\Omega} - \overline{X})$ and $\mathcal{G}_{\mathcal{P}^*} \in \partial\mathrm{Prox}_{\mathcal{P}^*}(\overline{\Theta} + \overline{X})$ such that

$$\mathcal{G}(\Delta\Omega,\Delta\Theta,\Delta X) = \begin{pmatrix} \Delta X + \mathcal{G}_{f^*}(\Delta\Omega - \Delta X) \\ -\Delta X + \mathcal{G}_{\mathcal{P}^*}(\Delta\Theta + \Delta X) \\ \Delta\Omega - \Delta\Theta \end{pmatrix} \ \forall\, (\Delta\Omega,\Delta\Theta,\Delta X) \in \mathbb{Z}\times\mathbb{Z}\times\mathbb{Z}.$$

Suppose that there exists $(\Delta\Omega,\Delta\Theta,\Delta X) \in \mathbb{Z}\times\mathbb{Z}\times\mathbb{Z}$ such that $\mathcal{G}(\Delta\Omega,\Delta\Theta,\Delta X) = 0$, i.e.,

$$\begin{cases} \Delta X + \mathcal{G}_{f^*}(\Delta\Omega - \Delta X) = 0, \\ \Delta X - \mathcal{G}_{\mathcal{P}^*}(\Delta\Theta + \Delta X) = 0, \\ \Delta\Omega - \Delta\Theta = 0. \end{cases} \tag{3.9}$$

It follows from Lemma 3.1 that both $\mathcal{G}_{f^*}$ and $\mathcal{B} := \mathcal{G}_{f^*}^{-1} - I$ are self-adjoint and positive definite. This, together with (3.9), implies that

$$\Delta\Omega = -\mathcal{B}\Delta X \ \text{ and } \ \mathcal{C}\Delta X = 0, \tag{3.10}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

where $\mathcal{C} := I - \mathcal{G}_{\mathcal{P}^*} + \mathcal{G}_{\mathcal{P}^*}\mathcal{B}$. We know from Lemma 3.2 that $\mathcal{C}$ is nonsingular. This, together with (3.9) and (3.10), implies that

$$\Delta\Omega = 0,\ \Delta\Theta = 0,\ \text{and}\ \Delta X = 0.$$

Therefore, $\mathcal{G}$ is nonsingular, and consequently the statement (a) holds.

The global Lipschitz continuities of the proximal mappings Prox_{f^*} and $\mathrm{Prox}_{\mathcal{P}^*}$ imply that the mapping $\mathcal{H}$ defined by (3.7) is Lipschitz continuous. Therefore, the proof of (b) can be obtained by (a), the fact that for any $(U, V, W) \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$, the set $\mathcal{S}(U, V, W)$ must be a singleton if it is nonempty, and Clarke's implicit function theorem (see [5, page 256] or [6, Theorem 3.6]). The proof is completed. ∎

4. Proximal point dual Newton algorithm. We aim to develop an implementable proximal point dual Newton algorithm (PPDNA) for solving the GGL problem (3.2). The PPDNA is essentially a proximal point algorithm (PPA) for solving the primal form of the GGL model, and the PPA subproblems are solved via their corresponding dual problems. The dual of each subproblem is to maximize a concave function whose gradient is a semismooth function and thus can be solved by the semismooth Newton method. We begin this section by introducing the PPA [24], i.e., given Ω_0, $\Theta_0 \in \mathbb{Z}_{++}$ and $\sigma_0 > 0$, the updating scheme is given by

$$\begin{cases} (\Omega_{t+1}, \Theta_{t+1}) \approx P_t(\Omega_t, \Theta_t) := \arg\min\limits_{\Omega,\Theta}\ \Phi_{\sigma_t}(\Omega, \Theta), \\ \sigma_{t+1} \uparrow \sigma_\infty \le \infty,\ \ t = 0, 1, \ldots, \end{cases} \tag{4.1}$$

where

$$\Phi_{\sigma_t}(\Omega, \Theta) := f(\Omega) + \mathcal{P}(\Theta) + \tfrac{1}{2\sigma_t}\|(\Omega, \Theta) - (\Omega_t, \Theta_t)\|^2 + \delta_{\mathbb{F}}(\Omega, \Theta) \tag{4.2}$$

with $\delta_{\mathbb{F}}$ being the indicator function of the set $\mathbb{F} := \{(\Omega, \Theta) \in \mathbb{Z} \times \mathbb{Z} \,|\, \Omega - \Theta = 0\}$, i.e., $\delta_{\mathbb{F}}(\Omega, \Theta) = 0$ if $(\Omega, \Theta) \in \mathbb{F}$, and $\delta_{\mathbb{F}}(\Omega, \Theta) = \infty$ if $(\Omega, \Theta) \notin \mathbb{F}$.

We allow for inexactness in the updating scheme (4.1) and apply the standard criteria proposed by Rockafellar [24] for controlling the inexactness: given nonnegative summable sequences $\{\varepsilon_t\}$ and $\{\gamma_t\}$ such that $\gamma_t < 1$ for all $t \ge 0$,

$$\text{(A)} \qquad \|(\Omega_{t+1}, \Theta_{t+1}) - P_t(\Omega_t, \Theta_t)\| \ \le\ \varepsilon_t,$$

$$\text{(B)} \qquad \|(\Omega_{t+1}, \Theta_{t+1}) - P_t(\Omega_t, \Theta_t)\| \ \le\ \gamma_t\|(\Omega_{t+1}, \Theta_{t+1}) - (\Omega_t, \Theta_t)\|.$$

Since the exact minimizer $P_t(\Omega_t, \Theta_t)$ is typically unknown in each iteration, we should introduce practically implementable stopping criteria in place of (A) and (B) in the subsequent analysis.

4.1. Dual based semismooth Newton method for solving PPA subproblems. In this section, we aim to design the aforementioned dual based semismooth Newton method for solving the subproblems of the PPA framework (4.1):

$$\begin{array}{ll} \min\limits_{\Omega,\Theta} & f(\Omega) + \mathcal{P}(\Theta) + \frac{1}{2\sigma_t}\|(\Omega, \Theta) - (\Omega_t, \Theta_t)\|^2 \\ \text{s.t.} & \Omega - \Theta = 0. \end{array} \tag{4.3}$$

The Lagrangian function associated with problem (4.3) is given by

$$\begin{aligned} \mathcal{L}_t(\Omega, \Theta, X) := {} & f(\Omega) + \mathcal{P}(\Theta) + \langle \Omega - \Theta, X\rangle \\ & + \tfrac{1}{2\sigma_t}\|\Omega - \Omega_t\|^2 + \tfrac{1}{2\sigma_t}\|\Theta - \Theta_t\|^2,\ \ (\Omega, \Theta, X) \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}, \end{aligned}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

and the Lagrangian dual problem of (4.3) is

$$\max_X \Big\{ \Upsilon_t(X) := \min_{\Omega,\Theta} \ \mathcal{L}_t(\Omega,\Theta,X) \Big\}. \tag{4.4}$$

Since we aim to solve problem (4.3) via its dual problem (4.4), a natural idea is to find the explicit expression for the dual objective function Υ_t first. The explicit expression for Υ_t can be obtained from the Moreau–Yosida regularization as follows:

$$\begin{aligned}
&\Upsilon_t(X)\\
&= \min_{\Omega} \ \big\{ f(\Omega) + \langle \Omega, X\rangle + \tfrac{1}{2\sigma_t}\|\Omega - \Omega_t\|^2 \big\} + \min_{\Theta} \ \big\{ \mathcal{P}(\Theta) - \langle \Theta, X\rangle + \tfrac{1}{2\sigma_t}\|\Theta - \Theta_t\|^2 \big\}\\
&= \sum_{k=1}^{K} \Big\{ \tfrac{1}{\sigma_t}\Psi_{\sigma_t h}\big(\Omega_t^{(k)} - \sigma_t(S^{(k)} + X^{(k)})\big) - \tfrac{1}{2\sigma_t}\|\Omega_t^{(k)} - \sigma_t(S^{(k)} + X^{(k)})\|^2 \Big\}\\
&\quad + \sum_{k=1}^{K} \tfrac{1}{2\sigma_t}\|\Omega_t^{(k)}\|^2 + \tfrac{1}{\sigma_t}\Psi_{\sigma_t \mathcal{P}}\big(\Theta_t + \sigma_t X\big) - \tfrac{1}{2\sigma_t}\|\Theta_t + \sigma_t X\|^2 + \tfrac{1}{2\sigma_t}\|\Theta_t\|^2.
\end{aligned}$$

The last equality is achieved when $\Omega^{(k)} = \phi^+_{\sigma_t}\big(\Omega_t^{(k)} - \sigma_t(S^{(k)} + X^{(k)})\big)$ for $k = 1, 2, \ldots, K$, and $\Theta = \mathrm{Prox}_{\sigma_t \mathcal{P}}(\Theta_t + \sigma_t X)$. One can find that Υ_t is continuously differentiable and strongly concave, and the unique solution to problem (4.4) can be obtained by solving the following nonsmooth system:

$$\nabla \Upsilon_t(X) = 0, \tag{4.5}$$

where

$$\nabla \Upsilon_t(X) = \big(\phi^+_{\sigma_t}(W_t^{(1)}(X)), \ldots, \phi^+_{\sigma_t}(W_t^{(K)}(X))\big) - \mathrm{Prox}_{\sigma_t \mathcal{P}}\big(V_t(X)\big)$$

with

$$W_t^{(k)}(X) := \Omega_t^{(k)} - \sigma_t(S^{(k)} + X^{(k)}),\ k = 1, \ldots, K, \text{ and } V_t(X) := \Theta_t + \sigma_t X.$$

We can see that if $X = \arg\min_X \Upsilon_t(X)$, then one has that $\Omega = \Theta$ with $\Omega^{(k)} = \phi^+_{\sigma_t}\big(W_t^{(k)}(X)\big)$ for $k = 1, 2, \ldots, K$, $\Theta = \mathrm{Prox}_{\sigma_t \mathcal{P}}(V_t(X))$. Recall that $\phi^+_{\sigma_t}(\cdot)$ is differentiable and its derivative is given by Proposition 2.3. Thus, the surrogate generalized Jacobian $\widehat{\partial}(\nabla \Upsilon_t)(X)$: $\mathbb{Z} \rightrightarrows \mathbb{Z}$ of $\nabla \Upsilon_t$ at any X can be defined as follows:

$$\begin{cases}
\mathcal{V} \in \widehat{\partial}(\nabla \Upsilon_t)(X) \text{ if and only if there exists } \mathcal{W} \in \widehat{\partial}\mathrm{Prox}_{\mathcal{P}}(V_t(X)/\sigma_t) \text{ such that}\\
\text{for any } D \in \mathbb{Z},\\
\mathcal{V}[D] = -\sigma_t\Big((\phi^+_{\sigma_t})'(W_t^{(1)}(X))[D^{(1)}], \ldots, (\phi^+_{\sigma_t})'(W_t^{(K)}(X))[D^{(K)}] \Big) - \sigma_t \mathcal{W}[D].
\end{cases}$$

Based on the surrogate generalized Hessian $\widehat{\partial}(\nabla \Upsilon_t)(\cdot)$ of Υ_t, one can apply the following dual based semismooth Newton method (Algorithm 4.1) for solving problem (4.4) via solving the nonsmooth equation (4.5). To know more about the local and global methods for nonsmooth equations, we refer the readers to [9, sections 7 and 8] and references therein. The main computational cost of Algorithm 4.1 lies in Step 1 for finding the Newton direction. Therefore, we carefully analyze the second order sparsity structure in the surrogate generalized Jacobian and fully exploit the structure

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Algorithm 4.1 (DN($\Omega_t, \Theta_t, \sigma_t$)) Dual based Semismooth Newton Method.

Given $\bar{\eta} \in (0,1)$, $\tau \in (0,1]$, and $\mu \in (0,1/2)$, $\rho \in (0,1)$. Input $X_{t,0} \in \mathbb{Z}_{++}$. Iterate the following steps for $j = 0, 1, \ldots$.

repeat

Step 1. (Newton direction) Choose a specific map $\mathcal{V}_j \in \widehat{\partial}(\nabla\Upsilon_t)(X_{t,j})$. Use the conjugate gradient (CG) method to find an approximate solution D_j to

$$\mathcal{V}_j[D] = -\nabla\Upsilon_t(X_{t,j})$$

such that $\|\mathcal{V}_j[D_j] + \nabla\Upsilon_t(X_{t,j})\| \le \min(\bar{\eta}, \|\nabla\Upsilon_t(X_{t,j})\|^{1+\tau})$.

Step 2. (Line search) Set $\alpha_j = \rho^{m_j}$, where m_j is the smallest nonnegative integer m for which

$$\Upsilon_t(X_{t,j} + \rho^m D_j) \ge \Upsilon_t(X_{t,j}) + \mu\rho^m\langle\nabla\Upsilon_t(X_{t,j}), D_j\rangle,$$

and set $X_{t,j+1} = X_{t,j} + \alpha_j D_j$.

Step 3. (Primal-dual iterates) Compute the primal-dual iterates $(\widetilde{\Omega}, \widetilde{\Theta}, \widetilde{X})$:

$$\widetilde{\Omega}^{(k)} = \phi^+_{\sigma_t}\big(\Omega_t^{(k)} - \sigma_t(S^{(k)} + \widetilde{X}^{(k)})\big),\ k = 1, \ldots, K,$$
$$\widetilde{\Theta} = \widetilde{\Omega},\ \ \widetilde{X} = X_{t,j+1}.$$

until $(\widetilde{\Omega}, \widetilde{\Theta}, \widetilde{X})$ satisfies some given stopping conditions.

return $(\Omega_{t+1}, \Theta_{t+1}, X_{t+1}) = (\widetilde{\Omega}, \widetilde{\Theta}, \widetilde{X})$.

to reduce the cost. Due to the computation of $\phi^+_{\sigma_t}(\cdot)$ in Υ_t and $\nabla\Upsilon_t$, the jth iteration of Algorithm 4.1 requires Km_j computations of eigenvalue decompositions.

The following proposition states that Algorithm 4.1 for solving the dual of the PPA subproblem (4.4) is globally convergent and locally superlinearly or even quadratically convergent if the parameter τ is chosen to be 1.

PROPOSITION 4.1. *Let $\{X_{t,j}\}_{j\ge0}$ be the infinite sequence generated by Algorithm* 4.1. *Then $\{X_{t,j}\}_{j\ge0}$ converges to the unique optimal solution $\overline{X}_t$ of* (4.4)*, and the convergence rate is at least superlinear:*

$$\|X_{t,j+1} - \overline{X}_t\| = \mathcal{O}(\|X_{t,j} - \overline{X}_t\|^{1+\tau}),\ \tau \in (0,1].$$

Proof. Since $\mathrm{Prox}_{\mathcal{P}}$ is directionally differentiable, it follows from Proposition 2.1 that $\mathrm{Prox}_{\mathcal{P}}$ is strongly semismooth with respect to the multifunction $\widehat{\partial}\mathrm{Prox}_{\mathcal{P}}$ in (2.6) (for its definition, see, e.g., [17, Definition 1]). Therefore, the conclusion follows from the strong concavity of $\Upsilon_t(\cdot)$, Proposition 2.3, and [17, Theorem 3]. ∎

4.2. Implementable stopping criteria for PPA subproblems. Due to the lack of explicit forms of the exact solution $P_t(\Omega_t, \Theta_t)$, the stopping conditions (A) and (B) need to be replaced by some implementable conditions. Since Φ_{σ_t} defined by (4.2) is strongly convex with modulus $1/2\sigma_t$, one has the estimate

$$\Phi_{\sigma_t}(\Omega_{t+1}, \Theta_{t+1}) - \inf\Phi_{\sigma_t}(\Omega, \Theta) \ge \tfrac{1}{2\sigma_t}\|(\Omega_{t+1}, \Theta_{t+1}) - P_t(\Omega_t, \Theta_t)\|^2,$$

which implies that

$$\|(\Omega_{t+1}, \Theta_{t+1}) - P_t(\Omega_t, \Theta_t)\| \le \sqrt{2\sigma_t(\Phi_{\sigma_t}(\Omega_{t+1}, \Theta_{t+1}) - \inf\Phi_{\sigma_t}(\Omega, \Theta))}.$$

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

The unknown value $\inf \Phi_{\sigma_t}(\Omega, \Theta)$ can be replaced by any of its lower bounds converging to it. One choice is to consider the objective value of the dual problem (4.4). In particular, one has that

$$\inf \Phi_{\sigma_t}(\Omega, \Theta) = \max \Upsilon_t(X) \geq \Upsilon_t(X_{t,j}) \; \forall \, j = 0, 1, \ldots,$$

and hence

(4.6)

$$\|(\Omega_{t+1}, \Theta_{t+1}) - P_t(\Omega_t, \Theta_t)\| \leq \sqrt{2\sigma_t(\Phi_{\sigma_t}(\Omega_{t+1}, \Theta_{t+1}) - \Upsilon_t(X_{t,j}))}, \; j = 0, 1, \ldots.$$

Therefore, we can terminate Algorithm 4.1 if $(\Omega_{t+1}, \Theta_{t+1}, X_{t+1})$ satisfies the following conditions: given nonnegative summable sequences $\{\varepsilon_t\}$ and $\{\gamma_t\}$ such that $\gamma_t < 1$ for all $t \geq 0$,

$$\Phi_{\sigma_t}(\Omega_{t+1}, \Theta_{t+1}) - \Upsilon_t(X_{t+1}) \; \leq \; \varepsilon_t^2/2\sigma_t, \tag{A$'$}$$

$$\Phi_{\sigma_t}(\Omega_{t+1}, \Theta_{t+1}) - \Upsilon_t(X_{t+1}) \; \leq \; (\gamma_t^2/2\sigma_t)\|(\Omega_{t+1}, \Theta_{t+1}) - (\Omega_t, \Theta_t)\|^2. \tag{B$'$}$$

4.3. Linear rate convergence of PPDNA. Now, we are ready to formally present the promised PPDNA for solving problem (3.2).

Algorithm 4.2 (PPDNA) Proximal Point Dual based Newton Algorithm.

Input Ω_0, $\Theta_0 \in \mathbb{Z}_{++}$ and $\sigma_0 > 0$. Iterate the following steps for $t = 0, 1, \ldots$:

Step 1. Apply Algorithm 4.1 to obtain

$$(\Omega_{t+1}, \Theta_{t+1}, X_{t+1}) = \mathrm{DN}(\Omega_t, \Theta_t, \sigma_t)$$

under stopping criterion (A$'$) or (B$'$).

Step 2. Update $\sigma_{t+1} \uparrow \sigma_\infty \leq \infty$.

Along the line of Rockafellar's works [23, 24], the local linear convergence rate of the primal and dual iterative sequences generated by the PPA can be guaranteed by the Lipschitz continuity of the KKT solution mapping near the origin under proper stopping criteria of the PPA subproblems. However, the Lipschitz property of the KKT solution mapping requires the uniqueness of the KKT point, and this property is not straightforward to establish when the regularizer $\mathcal{P}$ is not a piecewise linear-quadratic function. As the property to ensure the linear convergence rate, especially the uniqueness assumption, is too restrictive to hold, Luque [20] extended the results and proved the local linear convergence of the PPA under the local upper Lipschitz continuity (see, e.g., [22, p. 208] for the definition) of the KKT solution mapping at the origin [7, p. 387]. The local upper Lipschitz continuity condition does not make the assumption on the uniqueness of the solution. However, the local upper Lipschitz continuity property may not hold when the KKT solution mapping is not piecewise polyhedral. Fortunately, for our GGL model, the strict convexity of the log-determinant function guarantees the uniqueness of the solution, and we prove in Theorem 3.3 that the KKT solution mapping $\mathcal{S}$ of the GGL model (defined by (3.6)) is Lipschitz continuous near the origin by taking advantage of the nice properties of the log-determinant function and Clarke's implicit function theorem. Therefore, the local linear convergence rate of the PPDNA can be obtained via the classical results by Rockafellar. The convergence results of Algorithm 4.2 for solving problem (3.2) are presented below.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

THEOREM 4.2. *Let $\{(\Omega_t, \Theta_t, X_t)\}_{t\geq 0}$ be an infinite sequence generated by Algorithm* 4.2 *under stopping criterion* (A′). *Then the sequence $\{(\Omega_t, \Theta_t)\}_{t\geq 0}$ converges to the unique solution $(\overline{\Omega}, \overline{\Theta})$ of* (3.2), *and the sequence $\{X_t\}_{t\geq 0}$ converges to the unique solution $\overline{X}$ of* (3.3). *Furthermore, if both criteria* (A′) *and* (B′) *are executed in Algorithm* 4.2, *then there exists $\bar{t} \geq 0$ such that for all $t \geq \bar{t}$, one has that*

$$\|(\Omega_{t+1}, \Theta_{t+1}, X_{t+1}) - (\overline{\Omega}, \overline{\Theta}, \overline{X})\| \leq \varrho_t \|(\Omega_t, \Theta_t, X_t) - (\overline{\Omega}, \overline{\Theta}, \overline{X})\|,$$

where the convergence rate is given by

$$1 > \varrho_t := [\kappa(\kappa^2 + \sigma_t^2)^{-1/2} + \gamma_t]/(1-\gamma_t) \to \varrho_\infty = \kappa(\kappa^2 + \sigma_\infty^2)^{-1/2} \ (\varrho_\infty = 0 \text{ if } \sigma_\infty = \infty)$$

and the parameter κ is from (3.8).

Proof. The global convergence of Algorithm 4.2 can be obtained from (4.6), [24, Theorem 1], and the uniqueness of the KKT point. The linear rate of convergence can be derived from (4.6), Theorem 3.3(b), and [24, Theorem 2]. The proof is completed. □

4.4. Extensions of PPDNA. Although the theoretical analysis and the algorithmic design presented in section 3 and section 4 focus on the GGL regularizer, these results can also be applied to the joint graphical model (1.1) with a different regularizer satisfying the following conditions:

(a) the regularizer is convex and positively homogenous (e.g., a norm function);
(b) the proximal mapping associated with the regularizer can be efficiently computed, and its surrogate generalized Jacobian can be explicitly characterized.

For example, we can show that both the pairwise fused graphical Lasso regularizer [8, equation (5)] and the sequential fused graphical Lasso regularizer [34, formula (2.2)] satisfy the conditions (a) and (b). More specifically, let λ_1 and λ_2 be positive parameters. The pairwise fused graphical Lasso regularizer and sequential fused graphical Lasso regularizer are given as follows:

$$\mathcal{P}_1(\Theta) = \lambda_1 \sum_{k=1}^{K} \sum_{i\neq j} |\Theta_{ij}^{(k)}| + \lambda_2 \sum_{k<k'} \sum_{i\neq j} |\Theta_{ij}^{(k)} - \Theta_{ij}^{(k')}| = \sum_{i\neq j} \varphi_1(\Theta_{[ij]}), \tag{4.7}$$

where $\varphi_1(x) = \lambda_1 \|x\|_1 + \lambda_2 \sum_{k<k'} |x_k - x_{k'}|,\ x \in \mathbb{R}^K$, and

$$\mathcal{P}_2(\Theta) = \lambda_1 \sum_{k=1}^{K} \sum_{i\neq j} |\Theta_{ij}^{(k)}| + \lambda_2 \sum_{k=2}^{K} \sum_{i\neq j} |\Theta_{ij}^{(k)} - \Theta_{ij}^{(k-1)}| = \sum_{i\neq j} \varphi_2(\Theta_{[ij]}), \tag{4.8}$$

where $\varphi_2(x) = \lambda_1 \|x\|_1 + \lambda_2 \sum_{k=2}^{K} |x_k - x_{k-1}|,\ x \in \mathbb{R}^K$.

By applying the same procedure as in section 2.1 for the GGL regularizer, we can obtain the proximal mappings associated with $\mathcal{P}_i$, $i = 1, 2$, and their surrogate generalized Jacobians from that of the clustered Lasso regularizer [18] and the fused lasso regularizer [17], respectively. Therefore, we can apply our PPDNA framework for solving the joint graphical model with a different regularizer given by either (4.7) or (4.8).

In addition to the direct extensions to the two regularizers above, the PPDNA framework is also applicable to joint graphical models with other regularizers discussed in [13]. More specifically, the regularizers in [13] have the following form:

$$\mathcal{P}(\Theta) = \mathcal{Q}_1(\Theta) + \mathcal{Q}_2(\Theta),$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

where $\mathcal{Q}_1(\Theta) := \lambda_1 \sum_{k=1}^K \sum_{i\neq j} |\Theta_{ij}^{(k)}|$, $\mathcal{Q}_2(\Theta) := \lambda_2 \sum_{k=2}^K \psi(\Theta^{(k)} - \Theta^{(k-1)})$. All the choices of the penalty function ψ in [13, section 2.1] can ensure condition (a) except for the Laplcacian penalty $\psi(\cdot) = \|\cdot\|^2$. Therefore, except for the case of the Laplacian penalty, the PPDNA framework can be directly applied once condition (b) holds. For the exceptional case, we may slightly modify our framework. Specifically, each iteration should be modified as follows: given Ω_0, Θ_0, $\Lambda_0 \in \mathbb{Z}_{++}$ and $\sigma_0 > 0$, the updating scheme is given by

$$\text{(4.9)} \qquad \begin{cases} (\Omega_{t+1}, \Theta_{t+1}, \Lambda_{t+1}) \approx P_t(\Omega_t, \Theta_t, \Lambda_{t+1}) := \arg\min\limits_{\Omega,\Theta,\Lambda} \ \Phi_{\sigma_t}(\Omega, \Theta, \Lambda), \\ \sigma_{t+1} \uparrow \sigma_\infty \le \infty, \ \ t = 0, 1, \ldots, \end{cases}$$

where

$$\Phi_{\sigma_t}(\Omega, \Theta, \Lambda) := f(\Omega) + \mathcal{Q}_1(\Theta) + \mathcal{Q}_2(\Lambda) + \tfrac{1}{2\sigma_t}\|(\Omega, \Theta, \Lambda) - (\Omega_t, \Theta_t, \Lambda_t)\|^2 + \delta_{\mathbb{F}}(\Omega, \Theta, \Lambda),$$

with $\delta_{\mathbb{F}}$ being the indicator function of the set

$$\mathbb{F} := \{(\Omega, \Theta, \Lambda) \in \mathbb{Z} \times \mathbb{Z} \,|\, \Omega - \Theta = 0, \, \Lambda - \Theta = 0\}.$$

Then the resulting modified PPDNA can be obtained by using arguments similar to those in section 4. But we should mention that further investigation will be necessary to overcome the underlying difficulty that the dual of the subproblems of (4.9) may not necessarily be strongly convex. We leave this part as our future research topic.

5. Numerical results. In this section, we evaluate the performance of the PPDNA in comparison with the ADMM and the proximal Newton-type method implemented in the work [34], which is referred to as MGL.[1] All the experiments are performed in MATLAB (version 9.7) on a Windows workstation (24-core, Intel Xeon E5-2680 @ 2.50GHz, 128 GB of RAM).

5.1. Implementation of ADMM. In this section, we briefly describe the ADMM for solving the dual problem (3.3), which can be equivalently written as follows:

$$\text{(5.1)} \qquad \min_{X,\, Z} \ \left\{ \sum_{k=1}^K h(Z^{(k)}) + \mathcal{P}^*(X) \,\Big|\, Z - X = S \right\}.$$

Given a parameter $\sigma > 0$, the augmented Lagrangian function associated with (5.1) is defined by

$$\widehat{\mathcal{L}}_\sigma(X, Z, \Theta) = \sum_{k=1}^K h(Z^{(k)}) + \mathcal{P}^*(X) + \langle Z - X - S, \, \Theta\rangle + \frac{\sigma}{2}\|Z - X - S\|^2,$$

and the KKT optimality conditions are

$$\Theta - \mathrm{Prox}_{\mathcal{P}}(\Theta + X) = 0, \ \ Z - X - S = 0, \ \ Z^{(k)} - \mathrm{Prox}_h(Z^{(k)} - \Theta^{(k)}) = 0, \ \ k = 1, \ldots, K.$$

The iterative scheme of the ADMM for problem (5.1) can be described as follows: given $\tau \in (0, (1+\sqrt{5})/2)$ and an initial point $(X_0, Z_0, \Theta_0) \in \mathbb{Z}_{++} \times \mathbb{Z}_{++} \times \mathbb{Z}_{++}$, the

[1] The solver is available at http://senyang.info/.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

tth iteration is given by

$$\text{(5.2)} \qquad \begin{cases} X_{t+1} = \arg\min_X \ \widehat{\mathcal{L}}_\sigma(X, Z_t, \Theta_t), \\ Z_{t+1} = \arg\min_Z \ \widehat{\mathcal{L}}_\sigma(X_{t+1}, Z, \Theta_t), \\ \Theta_{t+1} = \Theta_t + \tau\sigma(Z_{t+1} - X_{t+1} - S), \end{cases}$$

where X_{t+1} can be updated by $X_{t+1} = (Z_t + \sigma^{-1}\Theta_t - S) - \mathrm{Prox}_{\mathcal{P}}(Z_t + \sigma^{-1}\Theta_t - S)$, and $Z_{t+1} = (Z_{t+1}^{(1)}, \ldots, Z_{t+1}^{(K)})$ can be updated by $Z_{t+1}^{(k)} = \phi^{+}_{\sigma^{-1}}\big(X_t^{(k)} - \frac{1}{\sigma}\Theta_t^{(k)} + S^{(k)}\big)$, $k = 1, \ldots, K$.

In the practical implementation, we tuned the parameter σ according to the progress of primal and dual feasibilities (see, e.g., [15, section 4.4]) and used a larger step-length τ of 1.618. These two techniques can empirically accelerate the convergence speed. It is worth noting that the ADMM implemented by Yang et al. [34] used a fixed penalty parameter σ and the step-length $\tau = 1$.

5.2. Settings of experiments. The experimental settings are the same as those in [38, section 4]. We adopt the stopping criteria of PPDNA, ADMM, and MGL as below. Let $\epsilon > 0$ be a given tolerance. It is set as 10^{-6} in the following experiments.

- The PPDNA is terminated if $\eta_P \le \epsilon$, where

$$\eta_P := \max\left\{ \frac{\|\Theta - \mathrm{Prox}_{\mathcal{P}}(\Theta + X)\|}{1 + \|\Theta\|}, \frac{\|\Theta - \Omega\|}{1 + \|\Theta\|}, \frac{\|\Omega - \mathrm{Prox}_h^K(\Omega - S - X)\|}{1 + \|\Omega\|} \right\}$$

  with

$$\mathrm{Prox}_h^K(\Theta) := (\mathrm{Prox}_h(\Theta^{(1)}), \ldots, \mathrm{Prox}_h(\Theta^{(K)})) \in \mathbb{Z}, \ \Theta \in \mathbb{Z}.$$

- The ADMM is terminated when $\eta_A \le \epsilon$ or 20000 iterations are taken, where

$$\eta_A := \max\left\{ \frac{\|\Theta - \mathrm{Prox}_{\mathcal{P}}(\Theta + X)\|}{1 + \|\Theta\|}, \frac{\|Z - X - S\|}{1 + \|S\|}, \frac{\|Z - \mathrm{Prox}_h^K(Z - \Theta)\|}{1 + \|Z\|} \right\}.$$

- The MGL is terminated when the relative difference of its objective value with respect to the primal objective value obtained by the PPDNA is smaller than the given tolerance ϵ or the relative duality gap achieved by the PPDNA, i.e.,

$$\Delta_M := \frac{\mathrm{pobj}_M - \mathrm{pobj}_P}{1 + |\mathrm{pobj}_M| + |\mathrm{pobj}_P|} < \max\{\epsilon, \mathrm{relgap}_P\},$$

  where pobj_P, pobj_M, and relgap_P are the objective values obtained by the PPDNA, the MGL, and the relative duality gap attained by the PPDNA.

It is worth mentioning that we adopt a warm-starting technique in the initial stage of the PPDNA, instead of starting it from scratch. The warm-starting procedure consists of first running the ADMM (with identity matrices as the starting point) for a fixed number of iterations (3000 steps in our experiments) or up to a given tolerance (100ϵ in our experiments), and then using the resulting approximate solution as an initial point to warm-start the PPDNA. This idea is greatly motivated by two facts: (1) the ADMM can generate a solution of low to medium accuracy efficiently and might become slow when higher accuracy is required; (2) our algorithm PPDNA has been proven to be locally linearly convergent. Therefore, the warm-starting technique can integrate the advantages of both ADMM and PPDNA.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

We set the initial parameter in the stopping criterion (A′) to be $\varepsilon_0 = 0.5$ and decrease it by a ratio $\varsigma > 1$, i.e., $\varepsilon_{k+1} = \varepsilon_k/\varsigma$. Likewise, the parameter γ_k in the stopping criterion (B′) is updated in the same fashion as that for ε_k. For the parameters in Algorithm 4.1, we simply set $\bar{\eta} = 0.1$ and $\tau \in [0.1, 0.2]$ according to [40]. For the step on line search, we set $\mu = 10^{-4}$ and $\rho = 0.5$.

5.3. Descriptions of data sets. In this part, we describe the data sets which will be used later. Since these data sets have been discussed in [38], we briefly review them for the ease of reading:

- *University webpages data set.*[2] The original data was collected from computer science departments of various universities in 1997, manually classified into seven different classes: Student, Faculty, Course, Project, Staff, Department, and Other. The data we use, consisting of the first four classes, is preprocessed by stemming techniques [4]. Two thirds of the pages were randomly chosen for training (*Webtrain*) and the remaining third for testing (*Webtest*).
- 20 *newsgroups data set.*[3] This data set has 20 topics of newsgroup documents, and some of the topics are closely related to each other, while others are highly unrelated. Four subgroups are named as *NGcomp*, *NGrec*, *NGsci*, and *NGtalk* accordingly and will be used in our experiments.
- *SPX*500 *component stocks.*[4] This data set contains the daily returns of Standard & Poor's 500 (*SPX*500) constituents from 2004 to 2014. We also test on extracted data from 2004 to 2006.

5.4. Performance of PPDNA. In this part, we first give an elementary report of the effectiveness of the GGL model on synthetic nearest-neighbor networks generated by the mechanism in [16]. Second, we illustrate numerically the local linear convergence of the PPDNA for solving two representative instances, in correspondence with Theorem 4.2, which shows theoretically the local linear convergence of the PPDNA.

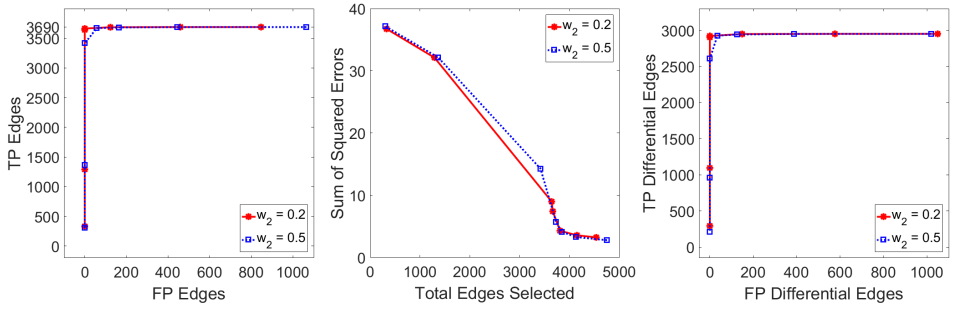
5.4.1. Synthetic data: Nearest-neighbor networks. In this example, we choose $p = 500$ and $K = 3$. The synthetic precision matrices, denoted as $\Sigma^{(k)}$, $k = 1, \ldots, K$, are generated as follows. We first generate p points on a unit square randomly, calculate their pairwise distances, and identify 5 nearest neighbors of each point. The nearest-neighbor network is then obtained by linking any two points that are 5 nearest neighbors of each other, and we denote the number of its edges as N. Subsequently, we obtain each $\Sigma^{(k)}$, $k = 1, 2, 3$, by adding extra edges to the common nearest-neighbor network. For each k, a pair of symmetric zero elements is randomly selected from the nearest-neighbor network and replaced with a value uniformly drawn from the interval $[-1, -0.5] \cup [0.5, 1]$. $\Sigma^{(k)}$ is obtained after this procedure is repeated ceil$(N/4)$ times. We find in our simulation that the true number of edges in the three networks is 3690. Given the precision matrices, we draw 10000 samples from each Gaussian distribution $\mathcal{N}_p(0, (\Sigma^{(k)})^{-1})$ to compute the sample covariance matrices. Next we specify the tuning parameters λ_1 and λ_2. Following [8], we reparameterize λ_1 and λ_2 in order to separate the regularization for "sparsity" and for "similarity" since both parameters contribute to sparsity: λ_1 drives individual network edges to zero, whereas λ_2 drives network edges to zero across all K network estimates at the same

[2] http://ana.cachopo.org/datasets-for-single-label-text-categorization
[3] http://qwone.com/~jason/20Newsgroups/
[4] http://www.yahoo.com

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

time. We reparameterize them in terms of $w_1 = \lambda_1 + \frac{1}{\sqrt{2}}\lambda_2$, $w_2 = \frac{1}{\sqrt{2}}\lambda_2/(\lambda_1 + \frac{1}{\sqrt{2}}\lambda_2)$, which were found in [8] to reflect the levels of sparsity and similarity regularization and were called the sparsity and similarity control parameters, respectively. In order to show the diversity of sparsity in our experiments, we change w_1 with w_2 fixed. Figure 2 characterizes the relative abilities of the GGL model to recover the network structures and to detect change-points.



(a) (b) (c)

FIG. 2. *Performances of the GGL model on nearest-neighbor networks ($p = 500$, $K = 3$).* (a) *The number of true positive edges versus the number of false positive edges.* (b) *The sum of squared errors in edge values versus the total number of edges selected.* (c) *The number of true positive differential edges versus the number of false positive differential edges.*

Figure 2a displays the number of true positive (TP) edges selected against the number of false positive (FP) edges. We say that an edge (i, j) is selected in the estimate $\overline{\Theta}^{(k)}$ if $\overline{\Theta}_{ij}^{(k)} \neq 0$, and the edge is true if $\Sigma_{ij}^{(k)} \neq 0$ and false if $\Sigma_{ij}^{(k)} = 0$. We can see that the model with $w_2 = 0.2$ can recover almost all of the TP edges without FP edges. This suggests that the GGL model is effective for recovering the edges in the nearest-neighbor networks. Figure 2b illustrates the sum of squared errors between estimated edge values and true edge values, i.e., $\sum_{k=1}^{K}\sum_{i<j}\left(\overline{\Theta}_{ij}^{(k)} - \Sigma_{ij}^{(k)}\right)^2$. When the number of the total edges selected increases (i.e., the sparsity control parameter w_1 decreases), the error decreases and finally reaches a fairly low value. Figure 2c plots the number of TP differential edges against FP differential. An edge that differs between networks is called a differential edge and thus corresponds to a change-point. Numerically, we say that the (i, j) edge is estimated to be differential between the kth and the $(k+1)$th networks if $|\overline{\Theta}_{ij}^{(k)} - \overline{\Theta}_{ij}^{(k+1)}| > 10^{-6}$, and we say that it is truly differential if $|\Sigma_{ij}^{(k)} - \Sigma_{ij}^{(k+1)}| > 10^{-6}$. The number of differential edges is computed for all successive pairs of networks. One can observe in Figure 2c that the results obtained with $w_2 = 0.2$ have approximately 3000 TP differential edges and almost no false ones. This suggests that the GGL model can be a suitable model to use in change-point detection of nearest-neighbor networks.

5.4.2. Linear rate convergence. The purpose of this section is to demonstrate numerically the local linear convergence of the PPDNA. Specifically, we conduct experiments on two representative instances: (a) categorical data: *Webtrain* with $(p, K) = (300, 4)$, $(\lambda_1, \lambda_2) = (5\text{e-}3, 5\text{e-}4)$; (b) time-varying data: *SPX500* with $(p, K) = (200, 11)$, $(\lambda_1, \lambda_2) = (5\text{e-}4, 5\text{e-}5)$. Due to the lack of exact optimal solutions of these instances, we run the PPDNA until the accuracy of 10^{-10} is achieved and

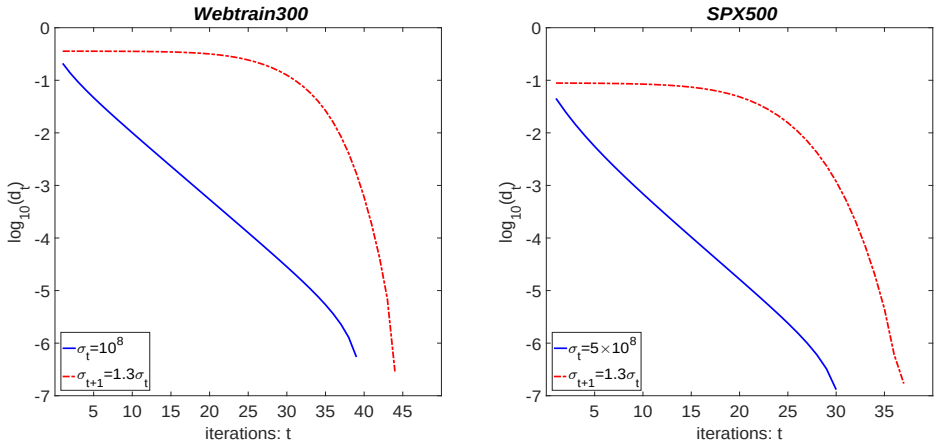
Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

regard the resulting approximate solution as the true solution $(\overline{\Omega}, \overline{\Theta}, \overline{X})$. We denote

$$d_t := \frac{\|\overline{\Omega}_t - \overline{\Omega}\| + \|\overline{\Theta}_t - \overline{\Theta}\| + \|\overline{X}_t - \overline{X}\|}{\|\overline{\Omega}\| + \|\overline{\Theta}\| + \|\overline{X}\|}, \ t = 0, 1, \ldots.$$

In Figure 3, we plot $\log_{10} d_t$ against the iteration count t under two different choices of the penalty parameter σ_t: σ_t is fixed or increased by a ratio. When σ_t is fixed, the solid blue line in the figure indicates that the convergence rate is almost constant. When $\sigma_{t+1} = 1.3\sigma_t$, i.e., the penalty parameter is gradually increasing, the dash-dotted red line shows that the convergence rate is increasingly fast. The observation is consistent with Theorem 4.2, which demonstrates numerically the local linear convergence rate of the PPDNA. We should emphasize that the impressive linear convergence rate depicted in the solid blue curve in Figure 3(a) is attained with σ_t fixed at a large value of 10^8, whereas the slower initial convergence shown in the dash-dotted red curve is due to slowly increasing the parameter σ_t from a small initial value of 2×10^4. The same remark is also applicable to Figure 3(b). (Color available online.)

The dependence of the linear rate of convergence on σ_t also sheds light on the choice of σ_t in our implementation. Basically we adaptively update σ_t to strike a good balance in the trade-off between the convergence rate of the PPDNA and the difficulty in computing the Newton directions (via the CG method) in the semismooth Newton method (Step 1 of Algorithm 4.1). As the condition number of the Newton linear system in Step 1 of Algorithm 4.1 is proportional to σ_t, the CG method will converge more slowly for a larger σ_t. Thus in our experiments, we start from a small σ_0, e.g., $\sigma_0 = 1$, and gradually increase σ_t by some factor $\zeta > 1$, i.e., $\sigma_{t+1} = \zeta\sigma_t$.



(a) (b)

FIG. 3. *The relative distances of the iterates generated by the PPDNA to the optimal solution.* (a) *Webtrain with* $(p, K) = (300, 4)$, $(\lambda_1, \lambda_2) = (5e\text{-}3, 5e\text{-}4)$. (b) *SPX*500 *with* $(p, K) = (200, 11)$, $(\lambda_1, \lambda_2) = (5e\text{-}4, 5e\text{-}5)$.

5.5. Comparison with ADMM and MGL. In this section, we compare our algorithm PPDNA for solving the GGL model with the ADMM described in (5.2) and the MGL implemented in [34]. For the tuning parameters λ_1 and λ_2, we select three pairs for each instance that produce reasonable sparsity. In the following tables, "P" stands for PPDNA; "A" stands for ADMM; "M" stands for MGL. In the column

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

under "Iteration," we report the number of iterations taken by various algorithms. In particular, for the PPDNA, we report the number of the PPA iterations taken and the number (within the parentheses) of semismooth Newton linear systems solved. Let "nnz" denote the number of nonzero entries in the solution Θ obtained by the PPDNA using the following estimation: $\text{nnz} := \min\{k \mid \sum_{i=1}^{k} |\hat{x}_i| \geq 0.999\|\hat{x}\|_1\}$, where $\hat{x} \in \mathbb{R}^{p^2K}$ is the vector obtained by sorting all elements of Θ by magnitude in a descending order. In the tables, "density" denotes the quantity $\text{nnz}/(p^2K)$. The time is displayed in the format of "hours:minutes:seconds," and the fastest method is highlighted in bold. The error reported for the PPDNA in the tables is the relative KKT residual η_P. That of the ADMM is η_A; while the error for the MGL is Δ_M.

Table 1 shows the comparison of three methods PPDNA, ADMM, and MGL on the university webpages data sets. The PPDNA successfully solved all instances in Table 1 within about one minute. For a large majority of tested instances, the PPDNA is faster than the ADMM and the MGL. It suggests that the PPDNA is robust and efficient for solving the GGL model applied to the university webpages data.

Table 2 presents the comparison of PPDNA, ADMM, and MGL on the 20 newsgroups data sets. One can see clearly that the PPDNA outperforms the ADMM and the MGL for most instances in Table 2. It demonstrates that the PPDNA can be efficient for solving the GGL model. For some difficult instances, e.g., *NGcomp* train $(\lambda_1, \lambda_2) = (5\text{e-}4, 5\text{e-}5)$, our PPDNA took less than one minute, while the MGL took more than one hour. Again, the results show that our PPDNA is robust for solving the GGL model. The superior performance of our PPDNA can primarily be attributed to our ability to extract and exploit the sparsity structure (in $\widehat{\partial}\text{Prox}_{\mathcal{P}}$) within the semismooth Newton method to solve the PPA subproblems very efficiently.

Table 3 gives the results on the Standard & Poor's 500 component stock price data set *SPX500*. The table shows that the PPDNA is faster than both the ADMM and the MGL for all instances. In addition, we find that both the PPDNA and the ADMM succeeded in solving all instances, while the MGL failed to solve one of them within three hours. This might imply that the MGL is not robust for solving the GGL model when applied to the stock price data sets. The numerical results show convincingly that our algorithm PPDNA can solve the GGL problem highly efficiently and robustly.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TABLE 1
Performances of PPDNA, ADMM, and MGL on university webpages' data.

Problem	(λ_1, λ_2)	Density	Iteration			Time			Error		
(p, K)			P	A	M	P	A	M	P	A	M
	(1e-02,1e-03)	0.016	16(24)	501	4	**02**	04	10	3.2e-07	9.9e-07	2.9e-07
Webtest	(5e-03,5e-04)	0.048	16(25)	501	6	**02**	04	13	3.2e-07	9.9e-07	2.6e-07
(100,4)	(1e-03,1e-04)	0.225	14(22)	529	32	**02**	04	56	2.4e-07	9.9e-07	1.0e-06
	(1e-02,1e-03)	0.008	14(24)	850	5	**08**	25	37	7.9e-07	1.0e-06	6.7e-08
Webtest	(5e-03,5e-04)	0.026	14(27)	679	7	**10**	19	50	7.9e-07	9.8e-07	4.1e-07
(200,4)	(1e-03,1e-04)	0.163	13(23)	503	77	**07**	11	05:57	2.7e-07	9.9e-07	1.6e-06
	(5e-03,5e-04)	0.016	14(29)	744	8	**28**	33	02:39	5.6e-07	9.9e-07	5.3e-08
Webtest	(1e-03,1e-04)	0.125	16(32)	487	205	45	**22**	21:51	5.9e-07	9.9e-07	1.8e-06
(300,4)	(5e-04,5e-05)	0.256	14(35)	668	1128	55	**30**	01:37:51	3.9e-07	9.9e-07	2.1e-06
	(1e-02,1e-03)	0.012	20(34)	1601	3	**03**	11	08	1.2e-07	1.0e-06	6.0e-06
Webtrain	(5e-03,5e-04)	0.033	20(34)	1601	5	**03**	12	04	1.2e-07	1.0e-06	7.4e-07
(100,4)	(1e-03,1e-04)	0.165	20(34)	1601	22	**04**	13	43	1.2e-07	1.0e-06	7.8e-06
	(5e-03,5e-04)	0.016	20(39)	1325	7	**13**	27	01:18	1.3e-07	1.0e-06	1.3e-06
Webtrain	(1e-03,1e-04)	0.108	20(37)	1397	31	**11**	31	02:49	9.9e-08	1.0e-06	5.0e-06
(200,4)	(5e-04,5e-05)	0.219	20(39)	1397	88	**16**	32	05:00	1.1e-07	1.0e-06	5.3e-06
	(5e-03,5e-04)	0.011	20(62)	1826	10	**01:04**	01:37	08:35	2.4e-07	9.7e-07	2.7e-06
Webtrain	(1e-03,1e-04)	0.080	19(33)	1196	45	**21**	52	09:55	3.4e-07	1.0e-06	6.0e-06
(300,4)	(5e-04,5e-05)	0.177	19(36)	1196	134	**36**	52	13:00	3.7e-07	1.0e-06	6.0e-06

TABLE 2
Performances of PPDNA, ADMM, and MGL on 20 *newsgroups' data.*

Problem	(λ_1, λ_2)	Density	Iteration			Time			Error		
(p, K)			P	A	M	P	A	M	P	A	M
NGcomp	(5e-03,5e-04)	0.021	15(22)	509	31	**16**	26	37:08	6.5e-08	9.9e-07	1.1e-06
test	(1e-03,1e-04)	0.099	16(26)	625	510	**32**	34	01:20:37	7.9e-07	1.0e-06	2.0e-06
(300,5)	(5e-04,5e-05)	0.210	14(24)	494	1481	40	**27**	03:00:00	7.2e-07	1.0e-06	6.1e-06
NGrec	(5e-03,5e-04)	0.004	21(38)	1331	5	**15**	49	04:04	8.0e-08	1.0e-06	4.9e-07
test	(1e-03,1e-04)	0.063	21(39)	1331	13	**20**	58	04:28	8.2e-08	1.0e-06	1.9e-06
(300,4)	(5e-04,5e-05)	0.143	20(37)	1331	36	**20**	58	07:49	3.7e-07	1.0e-06	3.7e-06
NGsci	(5e-03,5e-04)	0.006	17(26)	542	6	**14**	21	05:25	3.8e-07	1.0e-06	2.1e-06
test	(1e-03,1e-04)	0.075	17(27)	553	21	**19**	24	11:17	3.9e-07	9.8e-07	2.1e-06
(300,4)	(5e-04,5e-05)	0.167	17(31)	550	87	25	**24**	17:13	5.0e-07	9.9e-07	2.6e-06
NGtalk	(5e-03,5e-04)	0.026	15(25)	482	16	**24**	26	26:08	9.2e-08	9.6e-07	4.1e-07
test	(1e-03,1e-04)	0.115	12(23)	278	81	20	**13**	13:39	1.2e-07	9.9e-07	1.1e-06
(300,3)	(5e-04,5e-05)	0.240	11(22)	286	337	25	**13**	40:28	9.4e-08	9.7e-07	2.2e-06
NGcomp	(5e-03,5e-04)	0.016	16(31)	1150	13	**33**	57	14:58	1.2e-07	1.0e-06	5.6e-08
train	(1e-03,1e-04)	0.080	15(31)	1153	172	**35**	01:04	40:11	4.6e-07	1.0e-06	1.9e-06
(300,5)	(5e-04,5e-05)	0.153	15(30)	1216	574	**33**	01:07	01:12:51	4.4e-07	1.0e-06	1.8e-06
NGrec	(5e-03,5e-04)	0.005	19(35)	1519	5	**22**	52	02:36	1.4e-07	1.0e-06	3.9e-07
train	(1e-03,1e-04)	0.068	18(37)	1500	16	**31**	01:06	09:45	2.6e-07	1.0e-06	4.8e-06
(300,4)	(5e-04,5e-05)	0.124	18(35)	1542	48	**28**	01:07	09:02	2.9e-07	1.0e-06	5.4e-06
NGsci	(5e-03,5e-04)	0.011	17(30)	1387	10	**21**	54	10:00	1.7e-07	1.0e-06	3.5e-08
train	(1e-03,1e-04)	0.086	16(32)	1389	40	**32**	01:01	08:57	5.1e-07	1.0e-06	2.6e-06
(300,4)	(5e-04,5e-05)	0.152	16(32)	965	206	**37**	42	18:10	4.1e-07	9.9e-07	2.9e-06
NGtalk	(5e-03,5e-04)	0.026	18(32)	2445	13	**26**	01:41	13:24	2.6e-07	1.0e-06	1.9e-06
train	(1e-03,1e-04)	0.103	17(32)	2448	52	**19**	01:46	13:26	1.4e-07	1.0e-06	4.4e-06
(300,3)	(5e-04,5e-05)	0.204	17(38)	2385	213	**28**	01:45	19:43	7.2e-08	1.0e-06	4.9e-06

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 3
Performances of PPDNA, ADMM, and MGL on stock price data.

Problem	(λ_1, λ_2)	Density	Iteration			Time			Error		
(p, K)			P	A	M	P	A	M	P	A	M
	(1e-04,1e-05)	0.039	22(33)	3644	6	**04**	22	28	1.1e-07	1.0e-06	9.8e-07
SPX500	(5e-05,5e-06)	0.138	22(35)	3646	8	**05**	23	25	1.5e-07	1.0e-06	8.4e-06
(100,3)	(2e-05,2e-06)	0.238	23(43)	2056	18	**08**	13	08	4.4e-07	1.0e-06	2.5e-05
	(1e-04,1e-05)	0.025	24(31)	1409	8	**12**	23	01:20	8.8e-08	1.0e-06	9.4e-06
SPX500	(5e-05,5e-06)	0.084	21(28)	1239	17	**14**	20	04:23	9.4e-08	1.0e-06	9.0e-06
(200,3)	(2e-05,2e-06)	0.150	20(38)	1363	32	**18**	23	03:28	1.4e-07	9.9e-07	2.0e-05
	(5e-04,5e-05)	0.030	22(29)	3701	11	**12**	01:01	02:20	4.3e-08	1.0e-06	5.4e-06
SPX500	(1e-04,1e-05)	0.127	22(30)	3722	105	**18**	01:21	02:34	9.3e-08	1.0e-06	7.6e-06
(100,11)	(5e-05,5e-06)	0.206	22(30)	2925	393	**21**	01:06	09:12	7.8e-07	1.0e-06	7.8e-06
	(5e-04,5e-05)	0.018	19(24)	1096	28	**31**	53	36:07	8.4e-07	1.0e-06	4.1e-06
SPX500	(1e-04,1e-05)	0.082	19(24)	1125	481	**49**	01:08	01:27:17	7.9e-07	1.0e-06	5.1e-06
(200,11)	(5e-05,5e-06)	0.140	19(27)	1101	1258	**01:05**	01:08	03:00:00	6.1e-07	1.0e-06	1.7e-05

6. Concluding remarks. In this paper, we have taken advantage of the ideas proposed in [17, 39] and implemented a proximal point dual Newton algorithm (PPDNA) to the primal formulation of the group graphical Lasso problems. From a theoretical standpoint, we have shown that the PPDNA is globally convergent and the sequence of primal and dual iterates is Q-linearly convergent, although the group graphical Lasso regularizer is nonpolyhedral. The robustness and superior numerical efficiency of the PPDNA are convincingly demonstrated in various numerical experiments. Therefore, we can firmly conclude that the PPDNA is not only a fast method with nice theoretical guarantees but also a numerically efficient method for solving the group graphical Lasso problems with multiple precision matrices.

REFERENCES

[1] O. BANERJEE, L. E. GHAOUI, AND A. D'ASPREMONT, *Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data*, J. Mach. Learn. Res., 9 (2008), pp. 485–516.

[2] M. BOLLHÖFER, A. EFTEKHARI, S. SCHEIDEGGER, AND O. SCHENK, *Large-scale sparse inverse covariance matrix estimation*, SIAM J. Sci. Comput., 41 (2019), pp. A380–A401, https://doi.org/10.1137/17M1147615.

[3] J. F. BONNANS AND A. SHAPIRO, *Perturbation Analysis of Optimization Problems*, Springer, New York, 2000.

[4] A. CARDOSO-CACHOPO, *Improving Methods for Single-label Text Categorization*, Ph.D. Thesis, Instituto Superior Tecnico, Universidade Tecnica de Lisboa, Lisbon, Portugal, 2007.

[5] F. H. CLARKE, *Optimization and Nonsmooth Analysis*, Classics Appl. Math. 5, SIAM, Philadelphia, 1990, https://doi.org/10.1137/1.9781611971309.

[6] F. H. CLARKE, Y. S. LEDYAEV, R. J. STERN, AND P. R. WOLENSKI, *Nonsmooth Analysis and Control Theory*, Grad. Texts in Math. 178, Springer, New York, 1998.

[7] Y. CUI, D. F. SUN, AND K.-C. TOH, *On the R-superlinear convergence of the KKT residuals generated by the augmented Lagrangian method for convex composite conic programming*, Math. Program., 178 (2019), pp. 381–415.

[8] P. DANAHER, P. WANG, AND D. M. WITTEN, *The joint graphical lasso for inverse covariance estimation across multiple classes*, J. R. Stat. Soc. Ser. B Stat. Methodol., 76 (2014), pp. 373–397.

[9] F. FACCHINEI AND J.-S. PANG, *Finite-dimensional Variational Inequalities and Complementarity Problems*, Springer, New York, 2007.

[10] J. FRIEDMAN, T. HASTIE, AND R. TIBSHIRANI, *Sparse inverse covariance estimation with the graphical lasso*, Biostatistics, 9 (2008), pp. 432–441.

[11] J. FRIEDMAN, T. HASTIE, AND R. TIBSHIRANI, *A Note on the Group Lasso and a Sparse Group Lasso*, preprint, https://arxiv.org/abs/1001.0736, 2010.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[12] A. J. Gibberd and J. D. Nelson, *Regularized estimation of piecewise constant Gaussian graphical models: The group-fused graphical lasso*, J. Comput. Graph. Statist., 26 (2017), pp. 623–634.
[13] D. Hallac, Y. Park, S. Boyd, and J. Leskovec, *Network inference via the time-varying graphical lasso*, in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, 2017, pp. 205–213.
[14] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar, *QUIC: Quadratic approximation for sparse inverse covariance estimation*, J. Mach. Learn. Res., 15 (2014), pp. 2911–2947.
[15] X. Y. Lam, J. Marron, D. F. Sun, and K.-C. Toh, *Fast algorithms for large-scale generalized distance weighted discrimination*, J. Comput. Graph. Statist., 27 (2018), pp. 368–379.
[16] H. Li and J. Gui, *Gradient directed regularization for sparse Gaussian concentration graphs, with applications to inference of genetic networks*, Biostatistics, 7 (2006), pp. 302–317.
[17] X. Li, D. F. Sun, and K.-C. Toh, *On efficiently solving the subproblems of a level-set method for fused lasso problems*, SIAM J. Optim., 28 (2018), pp. 1842–1866, https://doi.org/10.1137/17M1136390.
[18] M. Lin, Y.-J. Liu, D. F. Sun, and K.-C. Toh, *Efficient sparse semismooth Newton methods for the clustered Lasso problem*, SIAM J. Optim., 29 (2019), pp. 2026–2052, https://doi.org/10.1137/18M1207752.
[19] Z. Lu and Y. Zhang, *An augmented Lagrangian approach for sparse principal component analysis*, Math. Program., 135 (2012), pp. 149–193.
[20] F. J. Luque, *Asymptotic convergence analysis of the proximal point algorithm*, SIAM J. Control Optim., 22 (1984), pp. 277–293, https://doi.org/10.1137/0322019.
[21] J.-J. Moreau, *Proximité et dualité dans un espace Hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299.
[22] S. M. Robinson, *Some continuity properties of polyhedral multifunctions*, in Mathematical Programming at Oberwolfach, Springer, New York, 1981, pp. 206–214.
[23] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.
[24] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim., 14 (1976), pp. 877–898, https://doi.org/10.1137/0314056.
[25] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1996.
[26] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, Grundlehren Math. Wiss. 317, Springer, Berlin, 1998.
[27] A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu, *Sparse permutation invariant covariance estimation*, Electron. J. Statist., 2 (2008), pp. 494–515.
[28] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, *A sparse-group lasso*, J. Comput. Graph. Statist., 22 (2013), pp. 231–245.
[29] D. F. Sun, *The strong second-order sufficient condition and constraint nondegeneracy in nonlinear semidefinite programming and their implications*, Math. Oper. Res., 31 (2006), pp. 761–776.
[30] D. F. Sun and L. Qi, *Solving variational inequality problems via smoothing-nonsmooth reformulations*, J. Comput. Appl. Math., 129 (2001), pp. 37–62.
[31] C. J. Wang, D. F. Sun, and K.-C. Toh, *Solving log-determinant optimization problems by a Newton-CG primal proximal point algorithm*, SIAM J. Optim., 20 (2010), pp. 2994–3013, https://doi.org/10.1137/090772514.
[32] S. J. Wright, R. D. Nowak, and M. A. Figueiredo, *Sparse reconstruction by separable approximation*, IEEE Trans. Signal Process., 57 (2009), pp. 2479–2493.
[33] J. Yang, D. F. Sun, and K.-C. Toh, *A proximal point algorithm for log-determinant optimization with group Lasso regularization*, SIAM J. Optim., 23 (2013), pp. 857–893, https://doi.org/10.1137/120864192.
[34] S. Yang, Z. Lu, X. Shen, P. Wonka, and J. Ye, *Fused multiple graphical lasso*, SIAM J. Optim., 25 (2015), pp. 916–943, https://doi.org/10.1137/130936397.
[35] K. Yosida, *Functional Analysis*, Springer, Berlin, 1964.
[36] M. Yuan and Y. Lin, *Model selection and estimation in regression with grouped variables*, J. R. Stat. Soc. Ser. B Stat. Methodol., 68 (2006), pp. 49–67.
[37] M.-C. Yue, Z. Zhou, and A. M.-C. So, *A family of inexact SQA methods for non-smooth convex minimization with provable convergence guarantees based on the Luo–Tseng error bound property*, Math. Program., 174 (2019), pp. 327–358.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[38] N. Zhang, Y. Zhang, D. F. Sun, and K.-C. Toh, *An Efficient Linearly Convergent Regularized Proximal Point Algorithm for Fused Multiple Graphical Lasso Problems*, preprint, https://arxiv.org/abs/1902.06952, 2019.

[39] Y. Zhang, N. Zhang, D. F. Sun, and K.-C. Toh, *An efficient Hessian based algorithm for solving large-scale sparse group Lasso problems*, Math. Program., 179 (2020), pp. 223–263.

[40] X.-Y. Zhao, D. F. Sun, and K.-C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM J. Optim., 20 (2010), pp. 1737–1765, https://doi.org/10.1137/080718206.

Downloaded 08/14/20 to 158.132.175.45. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.