A proximal Newton framework for composite minimization: Graph learning without Cholesky decompositions and matrix inversions

Quoc Tran Dinh QUOC.TRANDINH@EPFL.CH
Anastasios Kyrillidis ANASTASIOS.KYRILLIDIS@EPFL.CH
Volkan Cevher VOLKAN.CEVHER@EPFL.CH
Laboratory for Information and Inference Systems, École Polytechnique Fédérale de Lausanne, Switzerland

Abstract

We propose an algorithmic framework for convex minimization problems of a composite function with two terms: a self-concordant function and a possibly nonsmooth regularization term. Our method is a new proximal Newton algorithm that features a local quadratic convergence rate. As a specific instance of our framework, we consider the sparse inverse covariance matrix estimation in graph learning problems. Via a careful dual formulation and a novel analytic step-size selection procedure, our approach for graph learning avoids *Cholesky decompositions and matrix inversions* in its iteration making it attractive for parallel and distributed implementations.

1. Introduction

Sparse inverse covariance matrix estimation is a key step in graph learning problems. To understand the setup, let us consider learning a Gaussian Markov random field (GMRF) of p nodes/variables from a dataset $\mathcal{D} := \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m\}$, where $\mathbf{x}_j \in \mathcal{D}$ is a p-dimensional random vector with Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Let $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ be the inverse covariance (or the precision) matrix for the model. To satisfy the conditional dependencies with respect to the GMRF, $\boldsymbol{\Theta}$ must have zero in $\boldsymbol{\Theta}_{ij}$ corresponding to the absence of an edge between node i and node j (Dempster, 1972). To this end, one can use the empirical covariance matrix $\widehat{\boldsymbol{\Sigma}}$ to learn the underlying graph structure. Unfortunately, this approach is fundamentally ill-posed since the empirical estimates converge to the true covariance at a $(1/\sqrt{m})$-rate (Dempster, 1972). Hence, inferring the true graph structure accurately requires an overwhelming number of samples. Unsurprisingly, we usually have less samples than the ambient dimension, compounding the difficulty of estimation.

While the possible GMRF structures are exponentially large, the most interesting graphs are rather simple with a sparse set of edges. Provable learning of such graphs can be achieved by ℓ_1-norm regularization in the maximum log-likelihood estimation:

$$\boldsymbol{\Theta}^* \in \arg\min_{\boldsymbol{\Theta} \succ 0} \Big\{ \underbrace{-\log\det(\boldsymbol{\Theta}) + \mathrm{tr}(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Theta})}_{=:f(\boldsymbol{\Theta})} + \underbrace{\rho\|\mathbf{vec}(\boldsymbol{\Theta})\|_1}_{=:g(\boldsymbol{\Theta})} \Big\}, \quad (1)$$

where $\rho > 0$ is a parameter to balance the fidelity error and the sparsity of the solution and **vec** is the vectorization operator. Here, $f(\boldsymbol{\Theta})$ corresponds to the empirical log-likelihood and $g(\boldsymbol{\Theta})$ is the sparsity-promoting term. Under this setting, the authors in (Ravikumar et al., 2011) prove that $m = \mathcal{O}(d^2 \log p)$ is sufficient for correctly estimating the GMRF, where d is the graph node-degree. Moreover, the above formulation still makes sense for learning other graph models, such as the Ising model, due to the connection of $f(\boldsymbol{\Theta})$ to the Bregman distance (Banerjee et al., 2008).

Numerical solution methods for solving problem (1) have been widely studied in the literature now. For instance, in (Banerjee et al., 2008; Scheinberg & Rish, 2009; Scheinberg et al., 2010; Hsieh et al., 2011; Rolfs et al., 2012; Olsen et al., 2012) the authors proposed first order primal and dual approaches to (1) and used state-of-the-art structural convex optimization techniques such as coordinate descent methods and Lasso-based procedures. Alternatively, the authors in (Hsieh et al., 2011; Olsen et al., 2012) focused on the second order methods and, practically, achieved fast methods with a high accuracy. In (Scheinberg et al., 2010; Yuan, 2012), the authors studied alternating direction methods to solve (1), while the work in (Li & Toh,

Laboratory for Information and Inference Systems (LIONS), EPFL, Lausanne, Switzerland.

2010) is based on interior point-type methods. Algorithmic approaches where more structure is known a priori can be found in (Lu, 2010).

The complexity of the state-of-the-art approaches mentioned above is dominated by the Cholesky decomposition ($\mathcal{O}(p^3)$ in general), which currently creates an important scalability bottleneck. This decomposition appears mandatory since all these approaches employ a guess-and-check step-size selection procedures to ensure the iterates remain in the positive definite (PD) cone and the inversion of a $p \times p$ matrix, whose theoretical cost normally scales with the cost of $p \times p$ matrix multiplications ($\mathcal{O}(p^3)$ direct, $\mathcal{O}(p^{2.807})$ Strassen, and $\mathcal{O}(p^{2.376})$ Coppersmith-Winograd). The inversion operation is seemingly mandatory in the optimization of (1) since the calculation of the descent direction $\nabla f(\mathbf{\Theta}_i) := -\mathbf{\Theta}_i^{-1} + \widehat{\mathbf{\Sigma}}$ requires it, and quadratic cost approximations to $f(\mathbf{\Theta})$ also need it. Via Cholesky decompositions, one can first check if the current solution satisfies the PD cone constraint and then recycle the decomposition for inversion for the next iteration.

Contributions: We propose a new proximal-Newton framework for solving the general problem of (1). Our algorithm consists of two phases. In Phase 1, we apply a damped proximal Newton scheme with a new, analytic step-size selection procedure, and prove that our objective function always decreases at least a certain fixed amount. Moreover, our step-size selection is optimal in the sense that it cannot be improved without additional assumptions on the problem structure. In Phase 2, we simply apply the full step proximal-Newton iteration as we get into its provable quadratic convergence region which we can compute explicitly.

In the context of graph learning, we discuss a specific instance of our framework, which avoids Cholesky decompositions and matrix inversions *altogether*. Hence, the per iteration complexity of our approach is dominated by the cost of $p \times p$ matrix multiplications. This is because (i) our analytic step-size selection procedure ensures the positive definiteness of the iterates doing away with *global strategies* such as line-search which demands the objective evaluations (via Cholesky), and (ii) we avoid calculating the gradient explicitly and hence matrix inversion by a careful dual formulation. As a result, our approach is attractive for distributed and parallel implementations.

Paper outline: In Section 2, we first recall some fundamental concepts of convex optimization and self-concordant functions. Then, we describe the basic optimization set up and show the unique solvability of the problem. In Section 3 we outline our algorithmic framework and describe its analytical complexity. We also deal with the solution of the subproblems by applying the new dual approach in this section. Section 4 presents an application of our theory to graph selection problems. Experimental results on real graph learning problems can be found in Section 5.

2. Preliminaries

Basic definitions: We reserve lower-case and bold lower-case letters for scalar and vector representation, respectively. Upper-case bold letters denote matrices. Let **vec**: $\mathbb{R}^{p\times p} \to \mathbb{R}^{p^2}$ be the vectorization operator which maps a matrix to a single column, and **mat**: $\mathbb{R}^{p^2} \to \mathbb{R}^{p\times p}$ is the inverse mapping of **vec** which transforms a vector to a matrix. For a closed convex function f, we denote its domain by $\mathrm{dom}(f)$.

Definition 2.1 (Self-concordant functions (Nesterov & Nemirovski, 1994; Boyd & Vandenberghe, 2004))**.** *A convex function $h : \mathbb{R} \to \mathbb{R}$ is (standard) self-concordant if $|h'''(x)| \leq 2h''(x)^{3/2}$, $\forall x \in \mathbb{R}$. Furthermore, a function $h : \mathbb{R}^n \to \mathbb{R}$ is self-concordant if, for any $t \in \mathbb{R}$, the function $\phi(t) := h(\mathbf{x} + t\mathbf{v})$ is self-concordant for all $\mathbf{x}, \mathbf{v} \in \mathbb{R}^n$.*

Let $h \in \mathcal{C}^3(\mathrm{dom}(f))$ be a strictly convex and self-concordant function. For a given vector $\mathbf{v} \in \mathbb{R}^n$, the local norm around $\mathbf{x} \in \mathrm{dom}(f)$ with respect to $h(\cdot)$ is defined as $\|\mathbf{v}\|_{\mathbf{x}} := \left(\mathbf{v}^T\nabla^2 h(\mathbf{x})\mathbf{v}\right)^{1/2}$ while the corresponding dual norm is given as $\|\mathbf{v}\|_{\mathbf{x}}^* := \max_{\|\mathbf{u}\|_{\mathbf{x}}\leq 1} \mathbf{u}^T\mathbf{v} = \left(\mathbf{v}^T\nabla^2 h(\mathbf{x})^{-1}\mathbf{v}\right)^{1/2}$. Let $\omega : \mathbb{R} \to \mathbb{R}_+$ be a function defined as $\omega(t) := t - \ln(1+t)$ and $\omega_* : [0,1] \to \mathbb{R}_+$ be a function defined as $\omega_*(t) := -t - \ln(1-t)$. The functions ω and ω_* are both nonnegative, strictly convex and increasing. Based on (Nesterov, 2004), we recall the following estimates:

$$\omega(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}) + \nabla h(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + h(\mathbf{x}) \leq h(\mathbf{y}), \quad (2)$$
$$h(\mathbf{y}) \leq h(\mathbf{x}) + \nabla h(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + \omega_*(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}), \quad (3)$$

where (2) holds for all $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$, and (3) holds for all $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$ such that $\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} < 1$.

Problem statement: In this paper, we consider the following structural convex optimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \Big\{ F(\mathbf{x}) \;\; | \;\; F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \Big\}, \quad (4)$$

where $f(\mathbf{x})$ is a *convex, self-concordant* function and $g(\mathbf{x})$ is a proper, lower semicontinuous and nonsmooth convex regularization term. It is easy to certify that problem (1) can be transformed into (4) by using the tranformation $\mathbf{x} := \mathbf{vec}(\mathbf{\Theta})$:

$$f(\mathbf{x}) := \begin{cases} -\log\det(\mathbf{mat}(\mathbf{x})) + \mathrm{tr}(\widehat{\mathbf{\Sigma}}\mathbf{mat}(\mathbf{x})), & \mathbf{mat}(\mathbf{x}) \succ 0, \\ +\infty & \text{otherwise,} \end{cases}$$

$g(\mathbf{x}) := \rho \|\mathbf{x}\|_1$ and $n := p^2$.

Proximity operator: A basic tool to handle nonsmooth convex functions is proximity operator which is defined as follows. Let g be a proper lower semicontinuous, nonsmooth and convex in $\mathbb{R}^n$. We denote by $\partial g(\mathbf{x})$ the subdifferential of g at $\mathbf{x}$. Let f be a self-concordant function and $\bar{\mathbf{x}} \in \mathrm{dom} f$ be fixed. We define $P_g^{\bar{\mathbf{x}}}(\mathbf{u}) := (\nabla^2 f(\bar{\mathbf{x}}) + \partial g)^{-1}(\mathbf{u})$ for $\mathbf{u} \in \mathbb{R}^n$. This operator has the following cocoercive property

$$(P_g^{\bar{\mathbf{x}}}(\mathbf{u}) - P_g^{\bar{\mathbf{x}}}(\mathbf{v}))^T(\mathbf{u}-\mathbf{v}) \geq \left\|P_g^{\bar{\mathbf{x}}}(\mathbf{u}) - P_g^{\bar{\mathbf{x}}}(\mathbf{v})\right\|_{\bar{\mathbf{x}}}^2, \quad (5)$$

for all $\mathbf{u}, \mathbf{v}$; cf., (Lee et al., 2012). Consequently, $P_g^{\bar{\mathbf{x}}}$ is a nonexpansive mapping, i.e.,

$$\left\|P_g^{\bar{\mathbf{x}}}(\mathbf{u}) - P_g^{\bar{\mathbf{x}}}(\mathbf{v})\right\|_{\bar{\mathbf{x}}} \leq \|\mathbf{u}-\mathbf{v}\|_{\bar{\mathbf{x}}}^*. \quad (6)$$

Unique solvability of the problem: We generalize the result in (Hsieh et al., 2011) to show that problem (4) is uniquely solvable.

Lemma 2.2. *Suppose that the function f is self-concordant and g is proper lower semicontinuous and convex. For some $\mathbf{x} \in \mathrm{dom}(F)$, let $\lambda(\mathbf{x}) := \|\nabla f(\mathbf{x}) + \mathbf{v}\|_{\mathbf{x}}^* < 1$ for $\mathbf{v} \in \partial g(\mathbf{x})$. Then the solution $\mathbf{x}^*$ of (4) exists and is unique.*

The proof of this lemma can be done similarly as the main theorem in (Nesterov, 2004). For completeness, we provide it in the appendix.

3. Two-phase proximal Newton method

Our algorithmic framework is simply a proximal-Newton method which generates an iterative sequence $\{\mathbf{x}^k\}_{k\geq 0}$ starting from $\mathbf{x}^0 \in \mathrm{dom}(F)$. The new point $\mathbf{x}^{k+1}$ is computed as $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}^k$, where $\alpha_k \in (0,1]$ is an appropriate step size and $\mathbf{d}^k$ is the proximal-Newton-type direction which is a unique solution of the following subproblem:

$$\min_{\mathbf{d}} Q_{\bar{\mathbf{x}}^k}(\mathbf{d}; \mathbf{x}^k) + g(\mathbf{x}^k + \mathbf{d}). \quad (7)$$

Here, $Q_{\bar{\mathbf{x}}}(\mathbf{d}; \mathbf{x})$ is the following quadratic surrogate of the function f around x:

$$Q_{\bar{\mathbf{x}}}(\mathbf{d}; \mathbf{x}) := f(\mathbf{x}) + \nabla f(\mathbf{x})^T \mathbf{d} + \frac{1}{2}\mathbf{d}^T \nabla^2 f(\bar{\mathbf{x}})\mathbf{d}. \quad (8)$$

The approximate point $\bar{\mathbf{x}}^k$ will be specified later. The optimality condition for (7) is written as follows:

$$\mathbf{0} \in \partial g(\mathbf{x}+\mathbf{d}) + \nabla f(\mathbf{x}) + \nabla^2 f(\bar{\mathbf{x}})\mathbf{d}. \quad (9)$$

If we define $S^{\bar{\mathbf{x}}}(\mathbf{x}) := \nabla^2 f(\bar{\mathbf{x}})\mathbf{x} - \nabla f(\mathbf{x})$ then the unique solution $\mathbf{d}$ of (7) can be computed as

$$\mathbf{d} := \left(P_g^{\bar{\mathbf{x}}} \circ S^{\bar{\mathbf{x}}}\right)(\mathbf{x}) - \mathbf{x} := R_g^{\bar{\mathbf{x}}}(\mathbf{x}) - \mathbf{x} = -(\mathbb{I} - R_g^{\bar{\mathbf{x}}})(\mathbf{x}). \quad (10)$$

Here, $R_g^{\bar{\mathbf{x}}}(\cdot) := \left(P_g^{\bar{\mathbf{x}}} \circ S^{\bar{\mathbf{x}}}\right)(\cdot)$ and $\left(P_g^{\bar{\mathbf{x}}} \circ S^{\bar{\mathbf{x}}}\right)(\cdot) \equiv P_g^{\bar{\mathbf{x}}}(S^{\bar{\mathbf{x}}}(\cdot))$. For the rest of the paper, given a current estimate $\mathbf{x}$, we reserve $\mathbf{x}^+$ to denote the new estimate as $\mathbf{x}^+ := \mathbf{x} + \alpha\mathbf{d}$. Consequently, given the discussion above, the new iteration point $\mathbf{x}^+$ can be computed as

$$\mathbf{x}^+ = (1-\alpha)\mathbf{x} + \alpha R_g^{\bar{\mathbf{x}}}(\mathbf{x}). \quad (11)$$

The following lemma shows that the fixed point of $R_g^{\bar{\mathbf{x}}}$ is the unique solution of (4). The proof of this lemma is straightforward, and is omitted.

Lemma 3.1. *Let $R_g^{\bar{\mathbf{x}}}$ be a mapping defined by (11). Then $\mathbf{x}^*$ is the unique solution of (4) if and only if $\mathbf{x}^*$ is the fixed-point of $R_g^{\bar{\mathbf{x}}}$, i.e., $\mathbf{x}^* = R_g^{\bar{\mathbf{x}}}(\mathbf{x}^*)$.*

As usual, we call the mapping $R_g^{\bar{\mathbf{x}}}$ defined by (11) the *proximal-Newton-type operator*. Lemma 3.1 suggests that we can generate an iterative sequence based on the fixed-point principle. Theoretically, under certain assumptions, one can desire that the mapping $R_g^{\bar{\mathbf{x}}}$ is contractive and the sequence generated by this scheme is convergent. Hence, we characterize this below.

3.1. Full-step proximal-Newton scheme

We first show in this subsection that if we start sufficiently close to the solution $\mathbf{x}^*$ then we can compute our iterations with full-step, i.e., $\alpha = 1$. In this case, the proximal-Newton-type scheme (11) features the following iteration:

$$\mathbf{x}^+ := R_g^{\bar{\mathbf{x}}}(\mathbf{x}). \quad (12)$$

If the exact Hessian $\nabla^2 f(\mathbf{x})$ is used instead of $\nabla^2 f(\bar{\mathbf{x}})$ then we call this scheme the full-step proximal Newton (FPN) scheme. Consequently, we have $\mathbf{d} = x^+ - x$. Let us define $\lambda := \|\mathbf{d}\|_{\mathbf{x}}$ and $\lambda_+ = \|\mathbf{d}^+\|_{\mathbf{x}^+} = \left\|R_g^{\mathbf{x}^+}(\mathbf{x}^+) - \mathbf{x}^+\right\|_{\mathbf{x}^+}$. The following theorem establishes the local quadratic convergence of the FPN scheme (12).

Theorem 3.2. *Let $\bar{\mathbf{x}} := \mathbf{x}$ and $\mathbf{x}^+$ be a point generated by the full-step proximal Newton scheme (12). Then, if $\lambda = \|\mathbf{d}\|_{\mathbf{x}} < 1 - \frac{1}{\sqrt{2}} \approx 0.292893$, it holds that*

$$\lambda_+ \leq (1 - 4\lambda + 2\lambda^2)^{-1}\lambda^2. \quad (13)$$

Consequently, the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ generated by the FPN scheme $\mathbf{x}^{k+1} := R_g^{\mathbf{x}^k}(\mathbf{x}^k)$ with $\lambda_0 := \left\|\mathbf{x}^1 - \mathbf{x}^0\right\|_{\mathbf{x}^0} \leq \sigma \leq \bar{\sigma} := \frac{5-\sqrt{17}}{4} \approx 0.219224$, locally converges to $\mathbf{x}^*$ the unique solution of (4) at a quadratic rate.

The proof of Theorem 3.2 can be found in the appendix.

3.2. Damped proximal Newton scheme

We now establish that, with an appropriate choice of the step-size $\alpha \in (0,1]$, the iterative sequence $\{\mathbf{x}^k\}_{k\geq 0}$ generated by the *damped proximal Newton* scheme (11) is a decreasing sequence, i.e. $F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\sigma)$ whenever $\lambda_k \geq \sigma$, where $\sigma > 0$ is fixed. First, we show the following property for the new iteration $\mathbf{x}^+$.

Lemma 3.3. *Suppose that $\mathbf{x}^+$ is a point generated by $\mathbf{x}^+ := \mathbf{x} + \alpha\mathbf{d} \equiv (1-\alpha)\mathbf{x} + \alpha R_g^{\bar{\mathbf{x}}}(\mathbf{x})$. Then, we have*

$$F(\mathbf{x}^+) \leq F(\mathbf{x}) - \left(\alpha \|\mathbf{d}\|_{\bar{\mathbf{x}}}^2 - \omega^*(\alpha \|\mathbf{d}\|_{\mathbf{x}})\right), \qquad (14)$$

provided that $\|\mathbf{d}\|_x < 1/\alpha$.

Proof. Let $\mathbf{y} = \mathbf{x} + \mathbf{d} = R_g^{\bar{\mathbf{x}}}(\mathbf{x}) = R_g^{\bar{\mathbf{x}}}(\mathbf{x})$. It follows from the optimality condition of (9) that there exists $\mathbf{v}_\mathbf{y} \in \partial g(\mathbf{y})$ such that

$$\mathbf{v}_\mathbf{y} = -\nabla f(\mathbf{x}) - \nabla^2 f(\bar{\mathbf{x}})(\mathbf{y} - \mathbf{x}). \qquad (15)$$

Since f is self-concordant, by (3), for any $\mathbf{x}^+$ such that $\|\mathbf{x}^+ - \mathbf{x}\|_\mathbf{x} < 1$ we have

$$F(\mathbf{x}^+) \leq F(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{x}^+ - \mathbf{x}) + \omega^*(\|\mathbf{x}^+ - \mathbf{x}\|_\mathbf{x}) + g(\mathbf{x}^+) - g(\mathbf{x}). \qquad (16)$$

Since g is convex, $\alpha \in [0,1]$, by using (15) we have

$$\begin{aligned} g(\mathbf{x}^+) - g(\mathbf{x}) &= g((1-\alpha)\mathbf{x} + \alpha\mathbf{y}) - g(\mathbf{x}) \\ &\leq \alpha[g(\mathbf{y}) - g(\mathbf{x})] \leq \alpha\mathbf{v}_\mathbf{y}^T(\mathbf{y} - \mathbf{x}) = \alpha\mathbf{v}_\mathbf{y}^T\mathbf{d} \\ &\overset{(15)}{=} -\alpha\nabla f(\mathbf{x})^T\mathbf{d} - \alpha\|\mathbf{d}\|_{\bar{\mathbf{x}}}^2. \end{aligned} \qquad (17)$$

Now, substituting (17) into (16) and noting that $\mathbf{x}^+ - \mathbf{x} = \alpha\mathbf{d}$ we obtain the following result

$$\begin{aligned} F(\mathbf{x}^+) &\leq F(\mathbf{x}) + \omega^*(\alpha\|\mathbf{d}\|_\mathbf{x}) \\ &\quad - \alpha\nabla f(\mathbf{x})^T\mathbf{d} - \alpha\mathbf{d}^T\nabla^2 f(\mathbf{x})\mathbf{d} + \alpha\nabla f(\mathbf{x})^T\mathbf{d} \\ &= F(\mathbf{x}) - [\alpha\|\mathbf{d}\|_{\bar{\mathbf{x}}}^2 - \omega^*(\alpha\|\mathbf{d}\|_\mathbf{x})], \end{aligned} \qquad (18)$$

which is indeed (14), provided that $\|\mathbf{d}\|_\mathbf{x} < 1/\alpha$. □

Let us choose $\bar{\mathbf{x}} := \mathbf{x}$, i.e., we use the exact Hessian of f at the current iteration. The following theorem provides an explicit formula for the step size α.

Theorem 3.4. *Let $\bar{\mathbf{x}} := \mathbf{x}$ and $\lambda := \|\mathbf{d}\|_\mathbf{x}$, where $\mathbf{d}$ is the solution of (7). If we choose $\alpha := (1+\lambda)^{-1} \in (0,1]$ then the scheme (12) generates a new point $\mathbf{x}^+$ such that*

$$F(\mathbf{x}^+) \leq F(\mathbf{x}) - \omega(\lambda). \qquad (19)$$

Moreover, the step $\alpha = (1+\lambda)^{-1}$ is optimal.

Proof. By the choice of α, we have $\alpha\|\mathbf{d}\|_\mathbf{x} = (1+\lambda)^{-1}\lambda < 1$. By using the estimate (14) we have

$$F(\mathbf{x}^+) \leq F(\mathbf{x}) - (1+\lambda)^{-1}\lambda^2 + \omega^*\left((1+\lambda)^{-1}\lambda\right). \qquad (20)$$

Since $\frac{t^2}{1+t} - \omega^*(\frac{t}{1+t}) = \omega(t)$ for any $t > 0$, (20) implies (19). Finally, we note that the function $\varphi(\alpha) := \alpha\lambda(1+\lambda) + \ln(1-\alpha\lambda)$ is maximized at $\alpha = (1+\lambda)^{-1}$, which shows that the given α is optimal. □

Lemma 3.4 shows that the damped proximal Newton scheme generates a new point $\mathbf{x}^+$ such that it decreases the objective function F of (4) at least $\omega(\sigma)$ at each iteration whenever $\lambda \geq \sigma$.

Quadratic convergence: Similar to the full-step proximal-Newton scheme (12), we can also show the quadratic convergence of the damped proximal-Newton scheme (11). This statement is summarized in the following theorem.

Theorem 3.5. *Let $\bar{\mathbf{x}} := \mathbf{x}$ and $\mathbf{x}^+$ be a point generated by the damed proximal Newton scheme (11) with $\alpha := (1+\lambda)^{-1}$. Then, if $\lambda = \|\mathbf{d}\|_\mathbf{x} < 1 - \frac{1}{\sqrt{2}}$, it holds that*

$$\lambda_+ \leq (1 - 4\lambda + 2\lambda^2)^{-1}(1-\lambda^2)\lambda^2. \qquad (21)$$

Hence, the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ generated by the DPN *scheme $\mathbf{x}^{k+1} := (1-\alpha_k)\mathbf{x}^k + \alpha_k R_g^{\mathbf{x}^k}(\mathbf{x}^k)$ with $\lambda_0 \leq \sigma \leq \bar{\sigma} := 0.221876$ and $\alpha_k = (1+\lambda_k)^{-1}$, locally converges to $\mathbf{x}^*$, the unique solution of (4) at a quadratic rate.*

The proof of Theorem 3.5 can be found in the appendix. Note that the value $\bar{\sigma}$ in Theorem 3.5 is larger than in Theorem 3.2. However, both values are not tight.

3.3. The algorithm pseudocode

As proved by Theorems 3.4 and 3.5, we can only use the damped proximal-Newton scheme to build the algorithm. In this subsection, we present a two-phase proximal-Newton algorithm. We first select a constant $\sigma \in (0, \bar{\sigma}]$. At each iteration, we compute the new point $\mathbf{x}^+$ by using the damped proximal Newton scheme (11) until we get $\lambda = \|\mathbf{d}\|_\mathbf{x} \leq \sigma$. Then, we switch to the full-step Newton scheme and perform it until the convergence is achieved. These steps are described in Algorithm 1.

Note that the radius σ of the quadratic convergence region in Algorithm 1 can be fixed at its upper bound $\bar{\sigma}$. The maximum number of iterations $j_{\max}$ and $k_{\max}$ can also be specified, if necessary.

Algorithm 1 (*Proximal Newton algorithm*)

Initialization:
Require a starting point $\mathbf{x}^0 \in \text{dom}(F)$ and a constant $\sigma \in (0, \bar{\sigma}]$, where $\bar{\sigma} := \frac{(5-\sqrt{17})}{4} \approx 0.219224$.

Phase 1: (*Damped proximal Newton iterations*).
for $j = 0$ **to** $j_{\max}$ **do**
1. Compute the proximal-Newton search direction $\mathbf{d}^j := R_g^{\mathbf{x}^j}(\mathbf{x}^j) - \mathbf{x}^j$ as (10).
2. Compute $\lambda_j := \left\|\mathbf{d}^j\right\|_{\mathbf{x}^j}$.
3. **if** $\lambda_j \leq \sigma$ **then** terminate Phase 1.
4. Otherwise, update the next iteration $\mathbf{x}^{j+1} := \mathbf{x}^j + \alpha_j \mathbf{d}^j$, where $\alpha_j := (1+\lambda_j)^{-1}$.

end for
Phase 2: (*Full-step proximal Newton iterations*).
Set $\mathbf{x}^0 := \mathbf{x}^j$ from Phase 1 and choose a desired accuracy $\varepsilon > 0$.
for $k = 0$ **to** $k_{\max}$ **do**
1. Compute the proximal-Newton direction $d^k := R_g^{\mathbf{x}^k}(\mathbf{x}^k) - \mathbf{x}^k$ as (10).
2. Compute $\lambda_k := \left\|\mathbf{d}^k\right\|_{\mathbf{x}^k}$.
3. **if** $\lambda_k \leq \varepsilon$ **then** terminate Phase 2.
4. Otherwise, update $\mathbf{x}^{k+1} := \mathbf{x}^k + \mathbf{d}^k$.

end for

3.4. Iteration-complexity analysis

We analyze the complexity of Algorithm 1 by separating Phase 1 and Phase 2. This analysis is summarized in the following theorem.

Theorem 3.6. *The maximum number of iterations required in Phase 1 does not exceed* $j_{\max} := \left\lfloor \frac{F(\mathbf{x}^0)-F(\mathbf{x}^*)}{\omega(\sigma)} \right\rfloor + 1$, *where* $\mathbf{x}^*$ *is the unique solution of* (4). *The maximum number of iterations required in Phase 2 to obtain* $\lambda_k \leq \varepsilon$ *does not exceed* $k_{\max} := O\left(\ln\ln\left(\frac{c}{\varepsilon}\right)\right)$, *where* $c := (1 - 4\sigma + 2\sigma^2)^{-1} > 0$.

Proof. Since $\lambda_j \geq \sigma$ for all $j \geq 0$ in Phase 1, it follows from Theorem 3.4 that $F(\mathbf{x}^{j+1}) \leq F(\mathbf{x}^j) - \omega(\sigma)$. By induction we have $F(\mathbf{x}^*) \leq F(\mathbf{x}^{j_{\max}}) \leq F(\mathbf{x}^0) - j_{\max}\omega(\sigma)$. This implies that $j_{\max} \leq [F(\mathbf{x}^0) - F(\mathbf{x}^*)]/\omega(\sigma)$. Hence, we can fix $j_{\max} := \left\lfloor \frac{F(\mathbf{x}^0)-F(\mathbf{x}^*)}{\omega(\sigma)} \right\rfloor + 1$. Next, let $c := (1 - 4\sigma + 2\sigma^2)^{-1} > 0$. By induction, it follows from Theorem 3.2 we have $\lambda_k \leq (c)^{2^k-1}\lambda_0^{2^k} \leq (c)^{2^k-1}\sigma^{2^k}$. In order to ensure $\lambda_k \leq \varepsilon$ we require $(c)^{2^k-1}\sigma^{2^k}$, which leads to $k \leq O(\ln\ln(c/\varepsilon))$. Hence, we can show that $k_{\max} := O(\ln\ln(c/\varepsilon))$. □

We note that we do not use $j_{\max}$ as a stopping criterion of Phase 1 of Algorithm 1. In practice, we only need an upper bound of this quantity to provide a safeguard for the number of iterations. If we fix σ at $\bar{\sigma}$ then $c \approx 4.561553$ and the complexity of Phase 2 becomes $O\left(\ln\ln\left(\frac{4.5}{\varepsilon}\right)\right)$.

3.5. Dual solution approach of the subproblem

In this subsection we consider a specific instance of g: $g(\mathbf{x}) := \rho\|\mathbf{x}\|_1$. First, we derive a dual formulation of the convex subproblem (7). For notational convenience, we let $\mathbf{q} := \nabla f(\mathbf{x})$, $\mathbf{H} := \nabla^2 f(\mathbf{x})$. Then, the convex subproblem (7) can be written equivalently as:

$$\min_{\mathbf{y}\in\mathbb{R}^n}\left\{\frac{1}{2}\mathbf{y}^T\mathbf{H}\mathbf{y} + (\mathbf{q}-\mathbf{H}\mathbf{x})^T\mathbf{y} + \rho\|\mathbf{y}\|_1\right\}. \tag{22}$$

By using the min-max principle, we can write (22) as

$$\max_{\|\mathbf{u}\|_\infty\leq 1}\min_{\mathbf{y}\in\mathbb{R}^n}\left\{\frac{1}{2}\mathbf{y}^T\mathbf{H}\mathbf{y} + (\mathbf{q}-\mathbf{Q}\mathbf{x})^T\mathbf{y} + \rho\mathbf{u}^T\mathbf{y}\right\}. \tag{23}$$

Solving the inner minimization in (23) we obtain:

$$\min_{\|\mathbf{u}\|_\infty\leq 1}\left\{\frac{1}{2}\mathbf{u}^T\mathbf{H}^{-1}\mathbf{u} + \tilde{\mathbf{q}}^T\mathbf{u}\right\}, \tag{24}$$

where $\tilde{\mathbf{q}} := \frac{1}{\rho}(\mathbf{H}^{-1}\mathbf{q} - \mathbf{x})$. Note that the objective function $\varphi(\mathbf{u}) := \frac{1}{2}\mathbf{u}^T\mathbf{H}^{-1}\mathbf{u} + \tilde{\mathbf{q}}^T\mathbf{u}$ of (24) is strongly convex. One can apply the fast projected gradient methods with linear convergence rate in (Nesterov, 2007; Beck & Teboulle, 2009) for solving this problem.

In order to recover the solution of the primal subproblem (7), we note that the solution of the parametric minimization problem in (23) is given by $\mathbf{y}^*(\mathbf{u}) := \mathbf{x} - \mathbf{H}^{-1}(\mathbf{q} + \rho\mathbf{u})$. Let $\mathbf{u}_{\mathbf{x}}^*$ be the optimal solution of (24). We can recover the primal proximal-Newton search direction $\mathbf{d}$ of the subproblem (7) as

$$\mathbf{d} = -\nabla^2 f(\mathbf{x})^{-1}[\nabla f(\mathbf{x}) + \rho\mathbf{u}_{\mathbf{x}}^*]. \tag{25}$$

To compute the quantity $\lambda := \|\mathbf{d}\|_{\mathbf{x}}$ in Algorithm 1, we use (25) such that:

$$\lambda := \|\mathbf{d}\|_{\mathbf{x}} = \|\nabla f(\mathbf{x}) + \rho\mathbf{u}_{\mathbf{x}}^*\|_{\mathbf{x}}^*. \tag{26}$$

Note that computing λ by (26) requires the inverse of the Hessian matrix $\nabla^2 f(\mathbf{x})$.

4. Application to graph selection

In this section, we customize the theory framework of Algorithm 1 by using only Phase 1 to solve the graph selection problem (1).

Quantification: For clarity, we retain the matrix variable form as presented in (1). We note that $f(\mathbf{\Theta})$

is a self-concordant convex function, while $g(\boldsymbol{\Theta})$ is a proper, lower semicontinuous and nonsmooth convex function. Thus, our theory presented above can be applied to (1). Given the current estimate $\boldsymbol{\Theta}_i \succ 0$, we have $\nabla f(\boldsymbol{\Theta}_i) = \mathbf{vec}(\widehat{\boldsymbol{\Sigma}} - \boldsymbol{\Theta}_i^{-1})$ and $\nabla^2 f(\boldsymbol{\Theta}_i) = \boldsymbol{\Theta}_i^{-1} \otimes \boldsymbol{\Theta}_i^{-1}$. Under this setting, the dual subproblem (24) becomes:

$$\mathbf{U}^* = \underset{\|\mathbf{vec}(\mathbf{U})\|_\infty \leq 1}{\arg\min} \left\{ \frac{1}{2}\mathrm{tr}((\boldsymbol{\Theta}_i\mathbf{U})^2) + \mathrm{tr}(\widetilde{\mathbf{Q}}\mathbf{U}) \right\}, \quad (27)$$

where $\widetilde{\mathbf{Q}} := \rho^{-1}[\boldsymbol{\Theta}_i\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Theta}_i - 2\boldsymbol{\Theta}_i]$. Given the dual solution $\mathbf{U}^*$ of (27), the primal proximal-Newton search direction (i.e. the solution of (7)) is computed as

$$\boldsymbol{\Delta}_i := -\left((\boldsymbol{\Theta}_i\widehat{\boldsymbol{\Sigma}} - \mathbb{I})\boldsymbol{\Theta}_i + \rho\boldsymbol{\Theta}_i\mathbf{U}^*\boldsymbol{\Theta}_i\right). \quad (28)$$

The quantity λ_i defined in (26) can be computed by

$$\lambda_i := \left(p - 2\cdot\mathrm{tr}\left(\mathbf{W}_i\right) + \mathrm{tr}\left(\mathbf{W}_i^2\right)\right)^{1/2}. \quad (29)$$

where $\mathbf{W}_i := \boldsymbol{\Theta}_i(\widehat{\boldsymbol{\Sigma}} + \rho\mathbf{U}^*)$.

The graph learning algorithm: Algorithm 2 summarizes the proposed scheme for graph selection.

Algorithm 2 (*Dual PN for graph selection (DPNGS)*)

Input: Matrix $\Sigma \succ 0$ and a given tolerance $\varepsilon > 0$.

Output: An approximate solution $\boldsymbol{\Theta}_i$ of (1).
Initialization: Find a starting point $\boldsymbol{\Theta}_0 \succ 0$.
for $i = 0$ **to** $i_{\max}$ **do**

1. Set $\widetilde{\mathbf{Q}} := \rho^{-1}\left(\boldsymbol{\Theta}_i\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Theta}_i - 2\boldsymbol{\Theta}_i\right)$.
2. Compute $\mathbf{U}^*$ in (27).
3. Compute λ_i by (29), where $\mathbf{W}_i := \boldsymbol{\Theta}_i(\widehat{\boldsymbol{\Sigma}} + \rho\mathbf{U}^*)$.
4. If $\lambda_i \leq \varepsilon$ terminate.
5. Compute $\boldsymbol{\Delta}_i := -\left((\boldsymbol{\Theta}_i\widehat{\boldsymbol{\Sigma}} - \mathbb{I})\boldsymbol{\Theta}_i + \rho\boldsymbol{\Theta}_i\mathbf{U}^*\boldsymbol{\Theta}_i\right)$.
6. Set $\alpha_i := (1 + \lambda_i)^{-1}$.
7. Update $\boldsymbol{\Theta}_{i+1} := \boldsymbol{\Theta}_i + \alpha_i\boldsymbol{\Delta}_i$.

end for

Overall, Algorithm 2 does not require any matrix inversion operation. It only needs matrix-vector and matrix-matrix calculations, making the parallelization of the code easier. We note that due to the predefined step-size selection α in Algorithm 1 we do not need to do any backtracking line-search step. This advantage can avoid some overhead computation regarding the evaluation of the objective function which is usually expensive in this application.

Arithmetical complexity analysis: Since the analytical complexity is provided in Theorem 3.6, we only analyze the arithmetical complexity of Algorithm 2 here. As we work through the dual problem, the primal solution is dense even if majority of the entries are rather small (e.g., smaller than 10^{-6}).[1] Hence, the arithmetical complexity of Algorithm 2 is dominated by the complexity of $p \times p$ matrix multiplications.

For instance, the computation of $\widetilde{\mathbf{Q}}$ and $\boldsymbol{\Delta}_i$ require basic matrix multiplications. For the computation of λ_i, we require two trace operations: $\mathrm{tr}(\mathbf{W}_i)$ in $\mathcal{O}(p)$ time-complexity and $\mathrm{tr}(\mathbf{W}_i^2)$ in $\mathcal{O}(p^2)$ time-complexity. We note here that, while $\mathbf{W}_i$ is a *dense* matrix, the trace operation requires only the computation of the diagonal elements of $\mathbf{W}_i^2$. Given $\boldsymbol{\Theta}_i$, α_i and $\boldsymbol{\Delta}_i$, $\boldsymbol{\Theta}_{i+1}$ requires $O(p^2)$ time-complexity.

To compute (27), we can use the fast projected gradient method (FPGM) (Nesterov, 2007; Beck & Teboulle, 2009) with step size $1/L$ where L is the Lipschitz constant of the gradient of the objective function in (27). It is easy to observe that $L := \gamma_{\max}^2(\boldsymbol{\Theta}_i)$ where $\gamma_{\max}(\boldsymbol{\Theta}_i)$ is the largest eigenvalue of $\boldsymbol{\Theta}_i$. For sparse $\boldsymbol{\Theta}_i$, we can approximately compute $\gamma_{\max}(\boldsymbol{\Theta}_i)$ is $\mathcal{O}(p^2)$ by using *iterative power methods* (typically, 10 iterations suffice). The projection onto $\|\mathbf{vec}(\mathbf{U})\|_\infty \leq 1$ clips the elements by unity in $\mathcal{O}(p^2)$ time. Thus, the time overhead due to acceleration is within $\mathcal{O}(p^2)$.

Given the above, FPGM requires a constant number of iterations $k_{\max}$, which is independent of the dimension p, to achieve an ε_{in} solution accuracy. Overall, the time-complexity for the solution in (27) is $\mathcal{O}(k_{\max}M)$, where M is the cost of matrix multiplication.

Remark 4.1 (Parallel and distributed implementation ability)**.** *In Algorithm 2, the outer loop does not require any Cholesky decomposition or matrix inversion. Suppose that the fast projected gradient method is applied to solve the dual subproblem* (27)*. The main operation needed in the whole algorithm is matrix-matrix multiplication of the form* $\boldsymbol{\Theta}_i\mathbf{U}\boldsymbol{\Theta}_\mathbf{i}$*, where* $\boldsymbol{\Theta}_i$ *and* $\mathbf{U}$ *are symmetric positive definite. This operation can naturally be computed in a parallel or distributed manner. For more details of such computations we refer the reader to (Bertsekas & Tsitsiklis, 1989).*

5. Numerical experiments

In this section we test DPNGS (Algorithm 2 in Section 4) and compare it with the state-of-the-art graph selection algorithm QUadratic Inverse Covariance (QUIC) algorithm (Hsieh et al., 2011) on a real world data set. The QUIC algorithm is also a Newton-based method,

[1] In our MATLAB code, we made no attempts for sparsification of the primal solution. The overall complexity of the algorithm can improve via thresholding tricks.

which in addition exploits the sparsity in solving its primal subproblems. We note that QUIC was implemented in C language while our codes are implemented in MATLAB.

Implementation details: We test DPNGS on MATLAB 2011b running on a PC Intel Xeon X5690 at 3.47GHz per core with 94Gb RAM. To solve (27), we use FPGM scheme as detailed in the appendix. We terminate FPGM if either $\|\mathbf{U}_{k+1} - \mathbf{U}_k\|_F \leq \varepsilon_{\text{in}} \max\{\|\mathbf{U}_k\|_F, 1\}$ or the number of iterations reaches $k_{\max}$ where $\varepsilon_{\text{in}} > 0$ and $k_{\max}$ will be specified later. The stopping criterion of the outer loop is $\lambda_i \leq 10^{-6}$ and the maximum of outer iterations is chosen as $i_{\max} := 200$. We test the following three variants of DPNGS:

1. DPNGS: $\varepsilon_{\text{in}} = 10^{-6}$ and $k_{\max} = 1000$.
2. DPNGS(5): $\varepsilon_{\text{in}} = 10^{-4}$ and $k_{\max} = 5$.
3. DPNGS(10): $\varepsilon_{\text{in}} = 10^{-5}$ and $k_{\max} = 10$.

The DPNGS(5) and DPNGS(10) variants can be considered as inexact variants of DPNGS.

Real-world data: In our experiments, we use the real biology data preprocessed by (Li & Toh, 2010) to compare the performance of the DPNGS variants above and QUIC (Hsieh et al., 2011) for 5 problems: `Lymph` ($p = 587$), `Estrogen` ($p = 692$), `Arabidopsis` ($p = 834$), `Leukemia` ($p = 1225$) and `Hereditary` ($p = 1869$). This dataset can be found at `http://ima.umn.edu/~maxxa007/send_SICS/`.

Convergence behaviour analysis: First, we verify the convergence behaviour of Algorithm 2 by analyzing the quadratic convergence of the quantity λ_i, where λ_i is defined by (29). Our analysis is based on the `Lymph` problem with $p = 587$ variables. We note that λ_i reveals the weighted norm of the proximal-gradient mapping of the problem. The convergence behaviour is plotted in Figure 1 for three different values of ρ, namely $\rho = 0.25$, $\rho = 0.1$, $\rho = 0.05$ and $\rho = 0.01$. Figure 1 shows that whenever the values of λ_i gets into the quadratic region, it converges with only a few iterations. As ρ becomes smaller, we need more iterations to get into the quadratic convergence region.

Next, we illustate the step-size α_i of DPNGS. Figure 2 shows the increasing behaviour of the step size on the same dataset. Since $\alpha_i = (1 + \lambda_i)^{-1}$, it converges quickly at the last iterations. We also compare our objective value and the objective value reported by QUIC in Figure 3: we reach the objective value -4.141662 after 69 iterations while QUIC needs 159 iterations.

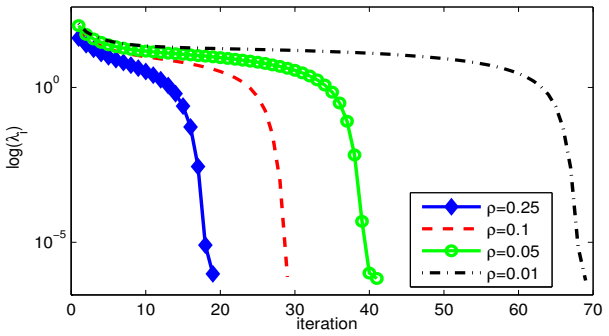

Figure 1. Quadratic convergence of DPNGS

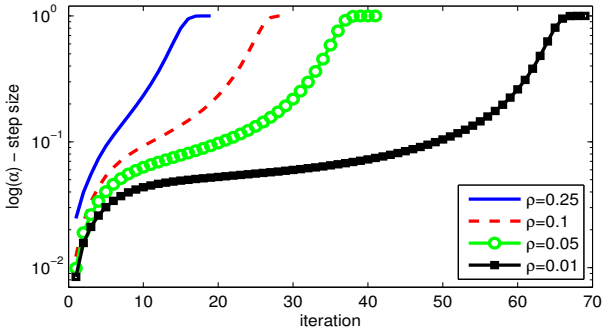

Figure 2. The step size of DPNGS

The last figure is the histogram of the solution in log scale reported by DPNGS and QUIC. Due to the dual solution approach, DPNGS reports an approximate solution with similar sparsity pattern as the one of QUIC. However, our solution has many small numbers instead of zero as in QUIC as revealed in Figure 4. *This seems to be the main weakness of the dual approach*: it obviates matrix inversions by avoiding the primal problem, which can return solutions with exact zeros thanks to its soft-thresholding prox-operator.

As a result, DPNGS carries around extremely small coefficients (almost of them smaller than 5×10^{-5}) often preventing it from achieving the same level accuracy as the numerical experiments on the full data set shows. At the same time, since the approach does not rely on coordinate descent on active sets, it appears much less sensitive to the choice of ρ. This could be an advantage of DPNGS in applications requiring smaller ρ values.

Numerical experiments on the full dataset: Now, we report the numerical experiments on the biology dataset and compare the methods. We test both algorithms with four different values of ρ, namely $\rho = 0.25$, $\rho = 0.1$, $\rho = 0.05$ and $\rho = 0.01$. The numerical results are reported in Table 1. Since QUIC exceeds the maximum number of iterations $i_{\max} = 200$

Table 1. Summary of comparison results on real world datasets.

Algorithm	$\rho = 0.25$			$\rho = 0.1$			$\rho = 0.05$			$\rho = 0.01$		
	#iter	time[s]	$F(\mathbf{\Theta}_i)$	#iter	time[s]	$F(\mathbf{\Theta}_i)$	#iter	time[s]	$F(\mathbf{\Theta}_i)$	#iter	time[s]	$F(\mathbf{\Theta}_i)$
	Lymph Problem ($p = 587$)											
DPNGS	19	49.028	613.26	29	61.548	341.89	40	66.635	133.60	69	104.259	-414.17
DPNGS(10)	39	7.470	613.42	34	8.257	342.12	43	8.678	133.87	78	35.543	-413.82
DPNGS(5)	61	7.067	615.87	30	4.323	344.72	41	5.862	136.37	69	123.552	-414.17
QUIC (C code)	22	8.392	613.25	44	33.202	341.88	82	176.135	133.60	201	2103.788	-414.17
	Estrogen Problem ($p = 692$)											
DPNGS	24	141.027	627.87	39	171.721	251.20	52	167.460	-11.59	83	205.262	-643.21
DPNGS(10)	56	15.500	628.10	49	14.092	251.52	59	19.262	-11.25	90	28.930	-642.85
DPNGS(5)	39	9.310	631.53	46	8.388	254.61	51	9.332	-7.69	81	42.955	-639.54
QUIC (C code)	19	7.060	627.85	43	49.235	251.19	81	244.242	-11.60	-	-	-
	Arabidopsis Problem ($p = 834$)											
DPNGS	26	174.947	728.57	43	220.365	228.16	61	253.180	-146.10	100	430.505	-1086.57
DPNGS(10)	48	22.268	728.96	45	22.404	228.57	60	26.007	-145.72	200	101.428	-1038.60
DPNGS(5)	38	9.826	733.67	44	11.113	233.04	57	18.378	-141.84	95	73.948	-1083.53
QUIC (C code)	21	19.684	728.52	49	116.016	228.14	95	562.532	-146.13	-	-	-
	Leukemia Problem ($p = 1255$)											
DPNGS	28	669.548	1143.79	48	624.145	386.37	71	726.688	-279.93	130	1398.133	-2071.33
DPNGS(10)	65	82.497	1144.66	48	60.108	387.26	68	84.017	-279.12	126	166.567	-2070.02
DPNGS(5)	49	38.317	1154.13	48	37.273	395.08	70	50.886	-271.01	124	258.090	-2060.25
QUIC (C code)	18	69.826	1143.76	41	344.199	386.33	76	1385.577	-280.07	-	-	-
	Hereditary Problem ($p = 1869$)											
DPNGS	41	2645.875	1258.31	82	3805.608	-348.49	113	5445.974	-1609.59	183	9020.237	-4569.85
DPNGS(10)	63	242.528	1261.15	80	297.131	-345.47	126	435.159	-1606.67	190	732.802	-4566.66
DPNGS(5)	58	129.821	1290.34	79	169.817	-313.87	126	439.386	-1606.67	179	1140.932	-4537.95
QUIC (C code)	21	437.252	1258.00	45	1197.895	-348.80	84	3182.211	-1609.92	-	-	-

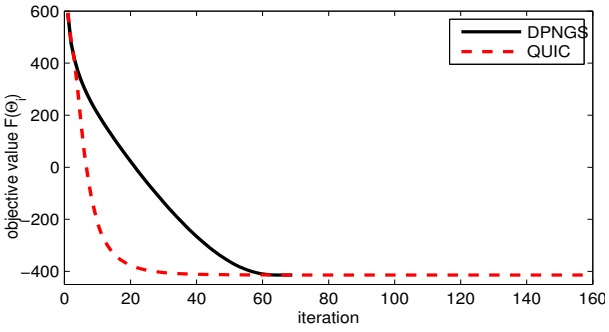

Figure 3. The objective values of DPNGS and QUIC

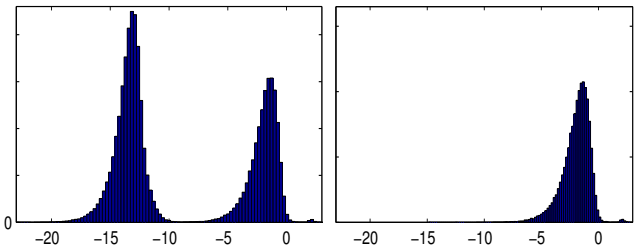

Figure 4. The histogram of the coefficient absolute values of the solution in log-scale of DPNGS and QUIC (right).

and takes a long time, we do not report the results corresponding to $\rho = 0.01$. We note again that our current implementation is done in MATLAB, while QUIC code is (carefully!) implemented in C. Hence, our timings may improve. We highlight several interesting results from Table 1. Firstly, QUIC obtains the highest accuracy results in all cases, which we attribute to the "lack of soft thresholding" in our algorithm. As the DPNGS algorithm carries around a score of extremely small numbers (effectively making the solution dense in the numerical sense), it can only come close to this accuracy within numerical precision. Moreover, QUIC is extremely efficient when ρ value is large, since it exploits the sparsity of the putative solutions via coordinate descent. Unsurprisingly, QUIC slows down significantly as ρ is decreased.

DPGNS(5) and DPNGS(10) algorithms can obtain near optimal solutions quite rapidly. In particular, DPNGS(10) seems to be the most competitive across the board, often times taking fraction of QUIC's time to provide a very close solution. Hence, one can expect these algorithms to be used for initializing other algorithms. For instance, QUIC can be a good candidate. We observed in all cases that in the first few iterations of the algorithm, QUIC performs several Cholesky decompositions to stay within the PD cone. As the complexity of this is large, our step-size selection procedure within QUIC or a DPNGS(10) initialization can be helpful.

6. Conclusions

In this paper, we demonstrate that graph learning is possible without any Cholesky decompositions via analytic step-size selection as well as without matrix inversions via a careful dual formulation. For the step-size selection, we exploit the self-concordant and smooth part of the composite graph learning objective, and then provide an optimal step-size for this class of composite minimization with proximal Newton methods. We show that within the dual formulation of the Newton subproblem, we do not need to explicitly calculate the gradient as it appears in a multiplication form with the Hessian. Thanks to the special structure of this multiplication, we can avoid matrix inversions.

While our theoretical results remove the $\mathcal{O}(p^3)$ complexity bottleneck by using only matrix operations, our MATLAB implementation needs to be improved to be competitive with the state-of-the-art QUIC algorithm. Hence, we plan to explore primal subproblem solutions with our step-size for computational trade-offs.

Acknowledgements

This work was supported in part by the European Commission under Grant MIRG-268398, ERC Future Proof, and SNF 200021-132548. Volkan Cevher also would like to acknowledge Rice University for his Faculty Fellowship.

A. The proofs of technical statements

A.1. The proof of Theorem 3.2

We provide a full-proof of Theorem 3.2.

Proof. Let us fix some $\bar{\mathbf{x}}$ in $\mathrm{dom} f$ and define $P_g^{\bar{\mathbf{x}}} := (\nabla^2 f(\bar{\mathbf{x}}) + \partial g)^{-1}$ and $\bar{S}_g^{\bar{\mathbf{x}}}(\mathbf{z}) := \nabla^2\mathbf{f}(\bar{\mathbf{x}})\mathbf{z} - \nabla\mathbf{f}(\mathbf{z})$. We also denote by $\mathbf{e} \equiv \mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{x}) := [\nabla^2 f(\mathbf{x}) - \nabla^2 f(\bar{\mathbf{x}})](\mathbf{x}^+ - \mathbf{x})$. If the exact Hessian of f is used in (7) then it follows from the optimality condition (9) that

$$\mathbf{0} \in \partial g(\mathbf{x}^+) + \nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{x})(\mathbf{x}^+ - \mathbf{x}).$$

This condition can be written equivalently to

$$\mathbf{0} \in \nabla^2 f(\bar{\mathbf{x}})\mathbf{x}^+ + \partial g(\mathbf{x}^+) - \nabla^2 f(\bar{\mathbf{x}})\mathbf{x} + \nabla\mathbf{f}(\mathbf{x}) + \mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{x}).$$

Therefore, the last relation leads to $\mathbf{x}^+ = P_g^{\bar{\mathbf{x}}}(S_g^{\bar{\mathbf{x}}}(\mathbf{x}) + \mathbf{e})$. We define $\mathbf{d} \equiv \mathbf{d}(\mathbf{x}) := \mathbf{x}^+ - \mathbf{x} = \mathbf{P}_{\mathbf{g}}^{\bar{\mathbf{x}}}(\mathbf{S}_{\mathbf{g}}^{\bar{\mathbf{x}}}(\mathbf{x}) + \mathbf{e}) - \mathbf{x}$. Then

$$\begin{aligned}\mathbf{d}^+ \equiv \mathbf{d}(\mathbf{x}^+) &:= P_g^{\bar{\mathbf{x}}}(S_g^{\bar{\mathbf{x}}}(\mathbf{x}^+) + \mathbf{e}^+) - \mathbf{x}^+ \\ &= P_g^{\bar{\mathbf{x}}}(S_g^{\bar{\mathbf{x}}}(\mathbf{x}^+) + \mathbf{e}^+) - P_g^{\bar{\mathbf{x}}}(S_g^{\bar{\mathbf{x}}}(\mathbf{x}) + \mathbf{e}).\end{aligned}$$

We consider the norm $\left\|\mathbf{d}^+\right\|_{\mathbf{x}}$. By using the nonexpansive property of $P_g^{\bar{\mathbf{x}}}$ we have

$$\begin{aligned}\left\|\mathbf{d}^+\right\|_{\bar{\mathbf{x}}} &= \left\|P_g^{\bar{\mathbf{x}}}(S_g^{\bar{\mathbf{x}}}(\mathbf{x}^+) + \mathbf{e}^+) - P_g^{\bar{\mathbf{x}}}(S_g^{\bar{\mathbf{x}}}(\mathbf{x}) + \mathbf{e})\right\|_{\bar{\mathbf{x}}} \\ &\overset{(6)}{\leq} \left\|S_g^{\bar{\mathbf{x}}}(\mathbf{x}^+) + \mathbf{e}^+ - S_g^{\bar{\mathbf{x}}}(\mathbf{x}) - \mathbf{e}\right\|_{\bar{\mathbf{x}}}^* \\ &\leq \left\|\nabla f(\mathbf{x}^+) - \nabla f(\mathbf{x}) - \nabla^2 f(\bar{\mathbf{x}})(\mathbf{x}^+ - \mathbf{x})\right\|_{\mathbf{x}}^* \\ &+ \left\|\mathbf{e}^+ - \mathbf{e}\right\|_{\bar{\mathbf{x}}}^* \\ &= \left[\left\|\int_0^1 [\nabla^2 f(\mathbf{x}_\tau) - \nabla^2\mathbf{f}(\bar{\mathbf{x}})](\mathbf{x}^+ - \mathbf{x})\mathbf{d}\tau\right\|_{\bar{\mathbf{x}}}^*\right]_{[1]} \\ &+ \left[\left\|\mathbf{e}^+ - \mathbf{e}\right\|_{\bar{\mathbf{x}}}^*\right]_{[2]}, \end{aligned} \tag{30}$$

where $\mathbf{x}_\tau := \mathbf{x} + \tau(\mathbf{x}^+ - \mathbf{x})$. First, we estimate the first term in the last line of (30) at $\bar{\mathbf{x}} = \mathbf{x}$, which we denote by $[\cdot]_{[1]}$. Now, we define $\Sigma := \int_0^1 [\nabla^2 f(\mathbf{x} + \tau(\mathbf{x}^+ - \mathbf{x})) - \nabla^2\mathbf{f}(\mathbf{x})]\mathbf{d}\tau$ and $\mathbf{H} := \nabla^2 f(\mathbf{x})^{-1/2}\Sigma\nabla^2 f(\mathbf{x})^{-1/2}$. Similar to the proof of Theorem 4.1.14 in (Nesterov, 2004), we can show that $\|\mathbf{H}\| \leq (1 - \|\mathbf{d}\|_{\mathbf{x}})^{-1}\|\mathbf{d}\|_{\mathbf{x}}$. Combining this inequality and (30) we deduce

$$[\cdot]_{[1]} = \|\Sigma\mathbf{d}\|_{\mathbf{x}}^* \leq \|\mathbf{H}\|\,\|\mathbf{d}\|_{\mathbf{x}} = (1 - \|\mathbf{d}\|_{\mathbf{x}})^{-1}\|\mathbf{d}\|_{\mathbf{x}}^2. \tag{31}$$

Next, we estimate the second term of (30) which is denoted by $[\cdot]_{[2]}$. We note that $\mathbf{e}_{\mathbf{x}}(\mathbf{x}) = 0$ and $\mathbf{e}_{\mathbf{x}}(\mathbf{x}^+) = [\nabla^2 f(\mathbf{x}^+) - \nabla^2 f(\mathbf{x})]\mathbf{d}^+$. Let $\mathbf{Q} := \nabla^2 f(\mathbf{x})^{-1/2}[\nabla^2 f(\mathbf{x}^+) - \nabla^2 f(\mathbf{x})]\nabla^2 f(\mathbf{x})^{-1/2}$. By applying Theorem 4.1.6 in (Nesterov, 2004) we can estimate $\|\mathbf{Q}\|$ as

$$\begin{aligned}\|\mathbf{Q}\| &\leq \max\left\{1 - (1 - \|\mathbf{d}\|_{\mathbf{x}})^2, \frac{1}{(1 - \|\mathbf{d}\|_{\mathbf{x}})^2} - 1\right\} \\ &= \frac{2\|\mathbf{d}\|_{\mathbf{x}} - \|\mathbf{d}\|_{\mathbf{x}}^2}{(1 - \|\mathbf{d}\|_{\mathbf{x}})^2}. \end{aligned} \tag{32}$$

Therefore, from the definition of $[\cdot]_{[2]}$ we have

$$\begin{aligned}[\cdot]_{[2]}^2 &= [\left\|\mathbf{e}^+ - \mathbf{e}\right\|_{\mathbf{x}}^*]^2 = (\mathbf{e}^+ - \mathbf{e})^T\nabla^2 f(\mathbf{x})^{-1}(e^+ - e) \\ &= (\mathbf{d}^+)^T\nabla^2 f(\mathbf{x})^{1/2}\mathbf{Q}^2\nabla^2 f(\mathbf{x})^{1/2}\mathbf{d}^+ \\ &\leq \|\mathbf{Q}\|^2\left\|\mathbf{d}^+\right\|_{\mathbf{x}}^2. \end{aligned} \tag{33}$$

By substituting (32) into (33) we obtain

$$[\cdot]_{[2]} \leq \frac{2\|\mathbf{d}\|_{\mathbf{x}} - \|\mathbf{d}\|_{\mathbf{x}}^2}{(1 - \|\mathbf{d}\|_{\mathbf{x}})^2}\left\|\mathbf{d}^+\right\|_{\mathbf{x}}. \tag{34}$$

Substituting (31) and (34) into (30) with $\bar{\mathbf{x}} = \mathbf{x}$ we obtain

$$\left\|\mathbf{d}^+\right\|_{\mathbf{x}} \leq \frac{\|\mathbf{d}\|_{\mathbf{x}}^2}{1 - \|\mathbf{d}\|_{\mathbf{x}}} + \frac{2\|\mathbf{d}\|_{\mathbf{x}} - \|\mathbf{d}\|_{\mathbf{x}}^2}{(1 - \|\mathbf{d}\|_{\mathbf{x}})^2}\left\|\mathbf{d}^+\right\|_{\mathbf{x}}.$$

By rearrange this inequality we obtain

$$\left\|\mathbf{d}^+\right\|_{\mathbf{x}} \leq \left[\frac{1-\|\mathbf{d}\|_{\mathbf{x}}}{1-4\|\mathbf{d}\|_{\mathbf{x}}+2\|\mathbf{d}\|_{\mathbf{x}}^2}\right]\|\mathbf{d}\|_{\mathbf{x}}^2. \tag{35}$$

On the other hand, we have

$$\left\|\mathbf{d}^+\right\|_{\mathbf{x}^+} \leq \frac{\left\|\mathbf{d}^+\right\|_{\mathbf{x}}}{1-\|\mathbf{d}\|_{\mathbf{x}}}. \tag{36}$$

Combining (35) and (36) we obtain

$$\left\|\mathbf{d}^+\right\|_{\mathbf{x}^+} \leq \frac{\|\mathbf{d}\|_{\mathbf{x}}^2}{1-4\|\mathbf{d}\|_{\mathbf{x}}+2\|\mathbf{d}\|_{\mathbf{x}}^2},$$

which is (13). Finally, we consider the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ generated by $\mathbf{x}^{k+1} := R_g^{\bar{\mathbf{x}}}(\mathbf{x}^k)$. From (13), we have $\left\|\mathbf{d}^1\right\|_{\mathbf{x}^1} \leq (1-4\left\|\mathbf{d}^0\right\|_{\mathbf{x}^0}+2\left\|\mathbf{d}^0\right\|_{\mathbf{x}^0}^2)^{-1}\left\|\mathbf{d}^0\right\|_{\mathbf{x}^0}^2 \leq (1-4\sigma+2\sigma^2)^{-1}\sigma^2 \leq \sigma$ provided that $0<\sigma\leq \frac{5-\sqrt{17}}{4}\approx 0.219224$. By induction, we can conclude that $\left\|\mathbf{d}^k\right\|_{\mathbf{x}^k} \leq \beta$ for all $k\geq 0$.It follows from (13) that $\left\|\mathbf{d}^{k+1}\right\|_{\mathbf{x}^{k+1}} \leq (1-4\sigma+2\sigma^2)^{-1}\left\|\mathbf{d}^k\right\|_{\mathbf{x}^k}^2$ for all k, which shows that $\left\{\left\|\mathbf{x}^k-\mathbf{x}^*\right\|_{\mathbf{x}^k}\right\}$ converges to zero at a quadratic rate. □

A.2. The proof of Theorem 3.5

Proof. First, we note that $\mathbf{x}^+ = \mathbf{x}+\alpha\mathbf{d} = \mathbf{x}+(1+\lambda)^{-1}x$. Hence, we can estimate $\mathbf{d}^+$ as:

$$\left\|\mathbf{d}^+\right\|_{\mathbf{x}^+} \leq \frac{\left\|\mathbf{d}^+\right\|_{\mathbf{x}}}{1-\alpha\lambda} = (1+\lambda)\left\|\mathbf{d}^+\right\|_{\mathbf{x}}.$$

Combining this inequality and (35) we obtain (21).

In order to prove the quadratic convergence, we first show that if $\lambda_k\leq\sigma$ then $\lambda_{k+1}\leq\sigma$ for all $k\geq 0$. Indeed, we note that the function $\varphi(t) := (1-t^2)(1-4t+2t^2)$ is increasing in $[0,1-1/\sqrt{2}]$. Let $\lambda_0\leq\sigma$. From (21) we have $\lambda_1\leq(1-\sigma^2)\sigma^2(1-4\sigma+2\sigma^2)$. Therefore, if $(1-\sigma^2)\sigma^2(1-4\sigma+2\sigma^2)\leq\sigma$ then $\lambda_1\leq\sigma$. The last requirement leads to $0<\sigma\leq\bar{\sigma} := 0.22187616$. From this argument, we conclude that if $\sigma\in(0,\bar{\sigma}]$ then if $\lambda_0\leq\sigma$ then $\lambda_1\leq\sigma$. By induction, we have $\lambda_k\leq\sigma$ for $k\geq 0$. If we define $c := (1-\sigma^2)(1-4\sigma+2\sigma^2)$ then $c>0$ and (21) implies $\lambda_{k+1}\leq c\lambda^2$ which shows that $\{\lambda_k\}_{k\geq 0}$ locally converges to 0 at a quadratic rate. □

A.3. The proof of Lemma 2.2.

Proof. From the self-concordance of f we have $\omega(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}})+f(\mathbf{x})+\nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x})\leq\mathbf{f}(\mathbf{y})$. On the other hand, since g is convex we have $g(\mathbf{y})\geq\mathbf{g}(\mathbf{x})+\mathbf{v}^{\mathbf{T}}(\mathbf{y}-\mathbf{x})$ for any $\mathbf{v}\in\partial\mathbf{g}(\mathbf{x})$. Hence, $F(\mathbf{y})\geq\mathbf{F}(\mathbf{x})+[\nabla\mathbf{f}(\mathbf{x})+\mathbf{v}]^{\mathbf{T}}(\mathbf{y}-\mathbf{x})+\omega(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}})\geq\mathbf{F}(\mathbf{x})-\lambda(\mathbf{x})\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}+\omega(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}})$, where $\lambda(\mathbf{x}) := \|\nabla f(\mathbf{x})+\mathbf{v}\|_{\mathbf{x}}^*$. Let $\mathcal{L}_F(F(\mathbf{x})) := \{\mathbf{y}\in\mathbb{R}^{\mathbf{n}}\mid\mathbf{F}(\mathbf{y})\leq\mathbf{F}(\mathbf{x})\}$ be a sublevel set of F. For any $y\in\mathcal{L}_F(F(\mathbf{x}))$ we have $F(\mathbf{y})\leq\mathbf{F}(\mathbf{x})$ which leads to $\lambda(\mathbf{x})\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}\geq\omega(\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}})$ due to the previous inequality. Note that ω is a convex and strictly increasing, the equation $\lambda(\mathbf{x})t=\omega(t)$ has unique solution $\bar{t}>0$ if $\lambda(\mathbf{x})<1$. Therefore, for any $0\leq t\leq\bar{t}$ we have $\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}\leq\bar{t}$. This implies that $\mathcal{L}_F(F(\mathbf{x}))$ is bounded. Hence, $\mathbf{x}^*$ exists. The uniqueness of $\mathbf{x}^*$ follows from the increase of ω. □

B. A fast projected gradient algorithm

For completeness, we provide here a variant of the fast-projected gradient method for solving the dual subproblem (27). Let us recall that $\texttt{clip}_r(X) := \text{sign}(X)\min\{|X|,r\}$ (a point-wise operator). The algorithm is presented as follows.

Algorithm 3 (*Fast-projected-gradient algorithm*)

Input: The current iteration $\mathbf{\Theta}_i$ and a given tolerance $\varepsilon_{\text{in}}>0$.
Output: An approximate solution $\mathbf{U}_k$ of (27).
Initialization: Compute a Lipschitz constant L and find a starting point $\mathbf{U}_0\succ 0$.
Set $\mathbf{V}_0 := \mathbf{U}_0$, $t_0 := 1$.
for $k=0$ **to** $k_{\max}$ **do**

1. $\mathbf{V}_{k+1} := \texttt{clip}_1\left(\mathbf{U}_k-\frac{1}{L}\left[\mathbf{\Theta}_i(\mathbf{U}_k+\frac{1}{\rho}\hat{\Sigma})\mathbf{\Theta}_i-\frac{2}{\rho}\mathbf{\Theta}_i\right]\right)$.
2. If $\|\mathbf{V}_{k+1}-\mathbf{V}_k\|_{\text{Fro}}\leq\varepsilon_{\text{in}}\max\{1,\|\mathbf{V}_k\|_{\text{Fro}}\}$ then terminate.
3. $t_{k+1} := 0.5(1+\sqrt{1+4t_k^2})$ and $\beta_k := \frac{t_k-1}{t_{k+1}}$.
4. $\mathbf{U}_{k+1} := \mathbf{V}_{k+1}+\beta_k(\mathbf{V}_{k+1}-\mathbf{V}_k)$.

end for

The main operator in Algorithm 3 is $\mathbf{\Theta}_i\mathbf{U}_k\mathbf{\Theta}_i$ at Step 2, where $\mathbf{\Theta}_i$ and $\mathbf{U}_k$ are symmetric and $\mathbf{\Theta}_i$ may be sparse. This operator requires twice matrix-matrix multiplications. The worst-case complexity of Algorithm 3 is typically $O\left(\sqrt{\frac{L}{\varepsilon_{\text{in}}}}\right)$ which is sublinear. If $\mu=\lambda_{\min}(\mathbf{\Theta}_i)$, the smallest eigenvalue of $\mathbf{\Theta}_i$, is available, we can set $\beta_k := \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}$ and we get a linear convergence rate.

References

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

Beck, A. and Teboulle, M. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

Bertsekas, D.P. and Tsitsiklis, J. N. *Parallel and distributed computation: Numerical methods*. Prentice Hall, 1989.

Boyd, S. and Vandenberghe, L. *Convex Optimization*. University Press, Cambridge, 2004.

Dempster, A. P. Covariance selection. *Biometrics*, 28: 157–175, 1972.

Hsieh, C. J., Sustik, M.A., Dhillon, I.S., and Ravikumar, P. Sparse inverse covariance matrix estimation using quadratic approximation. *Advances in Neutral Information Processing Systems (NIPS)*, 24:1–18, 2011.

Lee, J.D., Sun, Y., and Saunders, M.A. Proximal newton-type methods for convex optimization. *Tech. Report.*, pp. 1–25, 2012.

Li, L. and Toh, K.C. An inexact interior point method for l 1-regularized sparse covariance selection. *Mathematical Programming Computation*, 2(3):291–315, 2010.

Lu, Z. Adaptive first-order methods for general sparse inverse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2000–2016, 2010.

Nesterov, Y. *Introductory lectures on convex optimization: a basic course*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.

Nesterov, Y. Gradient methods for minimizing composite objective function. *CORE Discussion paper*, 76, 2007.

Nesterov, Y. and Nemirovski, A. *Interior-point Polynomial Algorithms in Convex Programming*. Society for Industrial Mathematics, 1994.

Olsen, P.A., Oztoprak, F., Nocedal, J., and Rennie, S.J. Newton-like methods for sparse inverse covariance estimation. *Optimization Online*, 2012.

Ravikumar, P., Wainwright, M. J., Raskutti, G., and Yu, B. High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. *Electron. J. Statist.*, 5:935–988, 2011.

Rolfs, B., Rajaratnam, B., Guillot, D., Wong, I., and Maleki, A. Iterative thresholding algorithm for sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems 25*, pp. 1583–1591, 2012.

Scheinberg, K. and Rish, I. Sinco-a greedy coordinate ascent method for sparse inverse covariance selection problem. *preprint*, 2009.

Scheinberg, K., Ma, S., and Goldfarb, D. Sparse inverse covariance selection via alternating linearization methods. *arXiv preprint arXiv:1011.0097*, 2010.

Yuan, X. Alternating direction method for covariance selection models. *Journal of Scientific Computing*, 51(2):261–273, 2012.