Composite Self-Concordant Minimization*

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 (LIONS)*
École Polytechnique Fédérale de Lausanne (EPFL)
CH1015-Lausanne, Switzerland

Editor: Unknown

Abstract

We propose a variable metric framework for minimizing the sum of a self-concordant function and a possibly non-smooth convex function endowed with a computable proximal operator. We theoretically establish the convergence of our framework without relying on the usual Lipschitz gradient assumption on the smooth part. An important highlight of our work is a new set of analytic step-size selection and correction procedures based on the structure of the problem. We describe concrete algorithmic instances of our framework for several interesting large-scale applications and demonstrate them numerically on both synthetic and real data.

Keywords: Proximal-gradient/Newton method, composite minimization, self-concordance, sparse convex optimization, graph learning.

1. Introduction

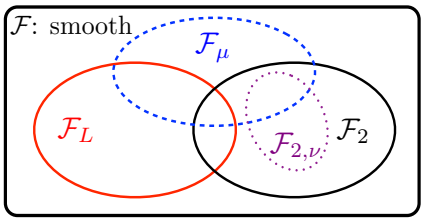
The literature on the formulation, analysis, and applications of *composite convex minimization* is ever expanding due to its broad applications in machine learning, signal processing, and statistics. By composite minimization, we refer to the following optimization problem:

$$F^* := \min_{\mathbf{x}\in\mathbb{R}^n} \left\{ F(\mathbf{x}) \;\; | \;\; F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \right\}, \tag{1}$$

where f and g are both closed and convex, and n is the problem dimension. In the canonical setting of the composite minimization problem (1), the functions f and g are assumed to be smooth and non-smooth, respectively (Nesterov, 2013). Such composite objectives naturally arise, for instance, in maximum a posteriori model estimation, where we regularize a model likelihood function as measured by a data-driven smooth term f with a non-smooth model prior g, which carries some notion of model complexity (e.g., sparsity, low-rankness, etc.).

In theory, many convex problem instances of the form (1) have a well-understood structure, and hence high accuracy solutions can be efficiently obtained with polynomial time methods, such as interior point methods (IPM) after transforming them into conic quadratic programming or semidefinite programming formulations (Ben-Tal and Nemirovski, 2001;

*. This work was supported in part by the European Commission under Grant MIRG-268398, ERC Future Proof and SNF 200021-132548, SNF 200021-146750 and SNF CRSII2-147633.



Class	Property $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f),\ \mathbf{v} \in \mathbb{R}^n, 0 \leq \mu \leq L < +\infty$
$\mathcal{F}_L$	$\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_* \leq L \|\mathbf{x} - \mathbf{y}\|$
$\mathcal{F}_\mu$	$\frac{\mu}{2}\|\mathbf{x} - \mathbf{y}\|^2 + f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) \leq f(\mathbf{y})$
$\mathcal{F}_2$	$\|\varphi'''(t)\| \leq 2\varphi''(t)^{3/2}$: $\varphi(t) = f(\mathbf{x} + t\mathbf{v}),\ t \in \mathbb{R}$
$\mathcal{F}_{2,\nu}$	$\mathcal{F}_2$ and $\sup_{\mathbf{v} \in \mathbb{R}^n} \left\{2\nabla f(\mathbf{x})^T\mathbf{v} - \|\mathbf{v}\|_{\mathbf{x}}^2\right\} \leq \nu$

Figure 1: Common structural assumptions on the smooth function f.

Grant et al., 2006; Nesterov and Nemirovski, 1994). *In practice*, however, the curse-of-dimensionality renders these methods impractical for large-scale problems. Moreover, the presence of a non-smooth term g prevents direct applications of scalable smooth optimization techniques, such as sequential linear or quadratic programming.

Fortunately, we can provably trade-off accuracy with computation for large-scale applications by further exploiting the individual structures of f and g. Existing methods invariably rely on two structural assumptions that particularly stand out among many others. First, we often assume that f has Lipschitz continuous gradient (i.e., $f \in \mathcal{F}_L$: cf., Fig. 1). Second, we assume that the proximal operator of g ($\mathrm{prox}_g^{\mathbf{H}}(\mathbf{y}) := \arg\min_{\mathbf{x} \in \mathbb{R}^n} \left\{g(\mathbf{x}) + 1/2\|\mathbf{x} - \mathbf{y}\|_{\mathbf{H}}^2\right\}$) is somewhat easy to compute for some $\mathbf{H} \succ 0$ (e.g., $\mathbf{H}$ is diagonal). On the basis of these structures, we can design algorithms featuring a full spectrum of (nearly) dimension-independent, global convergence rates with well-understood analytical complexity (see Table 1).

Table 1: Taxonomy of convex optimization methods when $f \in \mathcal{F}_L$ to reach $F(\mathbf{x}^k) - F^* \leq \epsilon$.

Order	Method example	Main oracle	Analytical complexity
1-st	[Accelerated] gradient	$\nabla f, \mathrm{prox}_g^{L\mathbb{I}_n}$	$[\mathcal{O}(\epsilon^{-1})]$ $\mathcal{O}(\epsilon^{-1/2})$
1^+-th	Proximal quasi-Newton	$\mathbf{H}_k, \nabla f, \mathrm{prox}_g^{\mathbf{H}_k}$	$\mathcal{O}(\log \epsilon^{-1})$ or faster
2-nd	Proximal Newton	$\nabla^2 f, \nabla f, \mathrm{prox}_g^{\nabla^2 f}$	$\mathcal{O}(\log\log \epsilon^{-1})$

(Becker and Fadili, 2012; Lee et al., 2012; Nesterov, 2004; Nocedal and Wright, 2006).

Unfortunately, existing large-scale algorithms have become inseparable with the Lipschitz gradient assumption on f and are still being applied to solve (1) in applications where this assumption does not hold. For instance, when $\mathrm{prox}_g^H(\mathbf{y})$ is not easy to compute, it is still possible to establish convergence—albeit slower—with smoothing, splitting or primal-dual decomposition techniques (Chambolle and Pock, 2011; Eckstein and Bertsekas, 1992; Nesterov, 2005a,b; Tran-Dinh et al., 2013c). However, when $f \notin \mathcal{F}_L$, the composite problems of the form (1) are not within the full theoretical grasp. In particular, there is no known global convergence rate. One kludge to handle $f \notin \mathcal{F}_L$ is to use sequential quadratic approximation of f to reduce the subproblems to the Lipschitz gradient case. For local convergence of these methods, we need *strong regularity* assumptions on f (i.e., $\mu\mathbb{I} \preceq \nabla^2 f(\mathbf{x}) \preceq L\mathbb{I}$) near the optimal solution. Attempts at global convergence require a *globalization strategy* such

as line search procedures (cf., Section 1.2). However, neither the strong regularity nor the line search assumptions can be certified *a priori*.

To this end, we address the following question in this paper: "Is it possible to efficiently solve large-scale instances of (1) for non-global Lipschitz continuous gradient f with rigorous global convergence guarantees?" The answer is positive (at least for a broad class of functions): We can still cover a full spectrum of global convergence rates with well-characterizable computation and accuracy trade-offs (akin to Table 1 for $f \in \mathcal{F}_L$) for self-concordant f (in particular, self-concordant barriers) (Nemirovski and Todd, 2009; Nesterov and Nemirovski, 1994):

Definition 1 (Self-concordant (barrier) functions) *A convex function $f : \mathbb{R}^n \to \mathbb{R}$ is said to be self-concordant (i.e., $f \in \mathcal{F}_M$) with parameter M, if $|\varphi'''(t)| \leq M\varphi''(t)^{3/2}$, where $\varphi(t) := f(\mathbf{x} + t\mathbf{v})$ for all $t \in \mathbb{R}$, $\mathbf{x} \in \mathrm{dom}(f)$ and $\mathbf{v} \in \mathbb{R}^n$ such that $\mathbf{x} + t\mathbf{v} \in \mathrm{dom}(f)$. When $M = 2$, the function f is said to be a standard self-concordant, i.e., $f \in \mathcal{F}_2$.*[1] *A standard self-concordant function $f \in \mathcal{F}_2$ is a ν-self-concordant barrier of a given convex set Ω with parameter $\nu > 0$, i.e., $f \in \mathcal{F}_\nu$, when φ also satisfies $|\varphi'(t)| \leq \sqrt{\nu}\varphi''(t)^{1/2}$ and $f(\mathbf{x}) \to +\infty$ as $\mathbf{x} \to \partial\Omega$, the boundary of Ω.*

While there are other definitions of self-concordant functions and self-concordant barriers (Boyd and Vandenberghe, 2004; Nemirovski and Todd, 2009; Nesterov and Nemirovski, 1994; Nesterov, 2004), we use Definition 1 in the sequel, unless otherwise stated.

1.1 Why is the assumption $f \in \mathcal{F}_2$ interesting for composite minimization?

The assumption $f \in \mathcal{F}_2$ in (1) is quite natural for two reasons. First, several important applications directly feature a self-concordant f, which does not have global Lipschitz continuous gradient. Second, self-concordant composite problems can enable approximate solutions of general constrained convex problems where the constraint set is endowed with a ν-self-concordant barrier function.[2] Both settings clearly benefit from scalable algorithms. Hence, we now highlight three examples below, based on compositions with the log-functions. Keep in mind that this list of examples is not meant to be exhaustive.

Log-determinant: The matrix variable function $f(\boldsymbol{\Theta}) := -\log\det\boldsymbol{\Theta}$ is self-concordant with $\mathrm{dom}(f) := \{\boldsymbol{\Theta} \in \mathbb{S}^p \mid \boldsymbol{\Theta} \succ 0\}$. As a stylized application, consider learning a Gaussian Markov random field (GMRF) of p nodes/variables from a dataset $\mathcal{D} := \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_m\}$, where $\boldsymbol{\phi}_j \in \mathcal{D}$ 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; cf., (Dempster, 1972).

We can learn GMRF's with theoretical guarantees from as few as $\mathcal{O}(d^2 \log p)$ data samples, where d is the graph node degree, via ℓ_1-norm regularization formulation (see (Raviku-

1. We use this constant for convenience in the derivations since if $f \in \mathcal{F}_M$, then $(M^2/4)f \in \mathcal{F}_2$.
2. Let us consider a constrained convex minimization $\mathbf{x}^*_C := \arg\min_{\mathbf{x}\in C} g(\mathbf{x})$, where the feasible convex set C is endowed with a ν-self-concordant barrier $\Psi_C(\mathbf{x})$. If we let $f(\mathbf{x}) := \frac{\epsilon}{\nu}\Psi_C(\mathbf{x})$, then the solution $\mathbf{x}^*$ of the composite minimization problem (1) well-approximates $\mathbf{x}^*_C$ as $g(\mathbf{x}^*) \leq g(\mathbf{x}^*_C) + (\nabla f(\mathbf{x}^*) + \partial g(\mathbf{x}^*))^T(\mathbf{x}^* - \mathbf{x}^*_C) + \epsilon$. The middle term can be controlled by accuracy at which we solve the composite minimization problem (Nesterov, 2011, 2013).

mar et al., 2011)):

$$\boldsymbol{\Theta}^* := \underset{\boldsymbol{\Theta} \succ 0}{\arg\min} \Big\{ \underbrace{-\log\det(\boldsymbol{\Theta}) + \operatorname{tr}(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Theta})}_{=:f(\boldsymbol{\Theta})} + \underbrace{\lambda \|\mathbf{vec}(\boldsymbol{\Theta})\|_1}_{=:g(\boldsymbol{\Theta})} \Big\}, \tag{2}$$

where $\lambda > 0$ parameter balances a Gaussian model likelihood and the sparsity of the solution, $\widehat{\boldsymbol{\Sigma}}$ is the empirical covariance estimate, and **vec** is the vectorization operator. The formulation also applies for learning models beyond GMRF's, such as the Ising model, since $f(\boldsymbol{\Theta})$ acts also as a Bregman distance (Banerjee et al., 2008).

Numerical solution methods for solving problem (2) have been extensively studied, e.g. in (Banerjee et al., 2008; Hsieh et al., 2011; Lee et al., 2012; Lu, 2010; Olsen et al., 2012; Rolfs et al., 2012; Scheinberg and Rish, 2009; Scheinberg et al., 2010; Yuan, 2012). However, none so far exploits $f \in \mathcal{F}_{2,\nu}$ and feature global convergence guarantees: cf., Sect. 1.2.

Log-barrier for linear inequalities: The function $f(\mathbf{x}) := -\log(\mathbf{a}^T\mathbf{x} - b)$ is a self-concordant barrier with $\operatorname{dom}(f) := \big\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}^T\mathbf{x} > b\big\}$. As a stylized application, consider the low-light imaging problem in signal processing (Harmany et al., 2012), where the imaging data is collected by counting photons hitting a detector over the time. In this setting, we wish to accurately reconstruct an image in low-light, which leads to noisy measurements due to low photon count levels. We can express our observation model using the Poisson distribution as:

$$\mathbb{P}(\mathbf{y}|\mathcal{A}(\mathbf{x})) = \prod_{i=1}^{m} \frac{(\mathbf{a}_i^T\mathbf{x})^{y_i}}{y_i!} e^{-\mathbf{a}_i^T\mathbf{x}},$$

where $\mathbf{x}$ is the true image, $\mathcal{A}$ is a linear operator that projects the scene onto the set of observations, $\mathbf{a}_i$ is the i-th row of $\mathcal{A}$, and $\mathbf{y} \in \mathbb{Z}_+^m$ is a vector of observed photon counts.

Via the log-likelihood formulation, we stumble upon a composite minimization problem:

$$\mathbf{x}^* := \underset{\mathbf{x} \in \mathbb{R}^n}{\arg\min} \Big\{ \underbrace{\sum_{i=1}^{m} \mathbf{a}_i^T\mathbf{x} - \sum_{i=1}^{m} y_i \log(\mathbf{a}_i^T\mathbf{x})}_{=:f(\mathbf{x})} + g(\mathbf{x}) \Big\}, \tag{3}$$

where $f(\mathbf{x})$ is self-concordant (but not standard). In the above formulation, the typical image priors $g(\mathbf{x})$ include the ℓ_1-norm for sparsity in a known basis, total variation semi-norm of the image, and the positivity of the image pixels. While the formulation (3) seems specific to imaging, it is also common in sparse regression with unknown noise variance (Städler et al., 2012), heteroschedastic LASSO (Dalalyan et al., 2013), and barrier approximations of, e.g., the Dantzig selector (Candes and Tao, 2007) as well.

The current state of the art solver is called SPIRAL-TAP (Harmany et al., 2012), which biases the logarithmic term (i.e., $\log(\mathbf{a}_i^T\mathbf{x} + \varepsilon) \to \log(\mathbf{a}_i^T\mathbf{x})$, where $\varepsilon \ll 1$) and then applies non-monotone composite gradient descent algorithms for $\mathcal{F}_L$ with a Barzilai-Borwein stepsize as well as other line-search strategies.

Logarithm of concave quadratic functions: The function $f(\mathbf{x}) := -\log\left(\sigma^2 - \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2\right)$ is self-concordant with $\operatorname{dom}(f) := \big\{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 < \sigma^2\big\}$. As a stylized application,

we consider the basis pursuit denoising (BPDN) formulation (van den Berg and Friedlander, 2008) as:

$$\mathbf{x}^* := \arg\min_{\mathbf{x}\in\mathbb{R}^n} \left\{ g(\mathbf{x}) \mid \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 \leq \sigma^2 \right\}. \tag{4}$$

The BPDN criteria is commonly used in magnetic resonance imaging (MRI) where $\mathbf{A}$ is a subsampled Fourier operator, $\mathbf{y}$ is the MRI scan data, and σ^2 is a known machine noise level (i.e., obtained during a pre-scan). In (4), g is an image prior, e.g., similar to the Poisson imaging problem. Approximate solutions to (4) can be obtained via a barrier formulation:

$$\mathbf{x}_t^* := \arg\min_{\mathbf{x}\in\mathbb{R}^n} \Big\{ \underbrace{-t\log\left(\sigma^2 - \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2\right)}_{=:f(\mathbf{x})} \; + \; g(\mathbf{x}) \Big\}, \tag{5}$$

where $t > 0$ is a penalty parameter which controls the quality of the approximation. The BPDN formulation is quite generic and has several other applications in statistical regression, geophysics, and signal processing.

Several different approaches solve the BPDN problem (4), some of which require projections onto the constraint set, including Douglas-Rachford splitting, proximal methods, and the SPGL$_1$ method (van den Berg and Friedlander, 2008; Combettes and Wajs, 2005).

1.2 Related work

Our attempt is to briefly describe the work that revolves around (1) with the main assumptions of $f \in \mathcal{F}_L$ and the proximal operator of g being computationally tractable. In fact, Douglas-Rachford splitting methods can obtain numerical solutions to (1) when the self-concordant functions are endowed with tractable proximal maps. However, it is computationally easier to calculate the gradient of $f \in \mathcal{F}_2$ than their proximal maps.

One of the main approaches in this setting is based on operator splitting. By presenting the optimality condition of problem (1) as an inclusion of two monotone operators, one can apply splitting techniques, such as forward-backward or Douglas-Rachford methods, to solve the resulting monotone inclusion (no Arias and Combettes, 2011; Facchinei and Pang, 2003; Goldstein and Osher, 2009). In our context, several variants of this approach have been studied. For example, projected gradient or proximal-gradient methods and fast proximal-gradient methods have been considered, see, e.g., (Beck and Teboulle, 2009a; Mine and Fukushima, 1981; Nesterov, 2013). In all these methods, the main assumption required to prove the convergence is the global Lipschitz continuity of the gradient of the smooth function f. Unfortunately, when $f \notin \mathcal{F}_L$ but $f \in \mathcal{F}_2$, these theoretical results on the global convergence and the global convergence rates are no longer applicable.

Other mainstream approaches for (1) include augmented Lagrangian and alternating techniques: cf., (Boyd et al., 2011; Goldfarb and Ma, 2012). These methods have empirically proven to be quite powerful in specific applications. The main disadvantage of these methods is the manual tuning of the penalty parameter in the augmented Lagrangian function, which is not yet well-understood for general problems. Consequently, the analysis of global convergence as well as the convergence rate is an issue since the performance of the algorithms strongly depends on the choice of this penalty parameter in practice. Moreover, as indicated in a recent work (Goldstein et al., 2012), alternating direction methods of multipliers as well as alternating linearization methods can be viewed as splitting methods in

the convex optimization context. Hence, it is unclear if this line of work is likely to lead to any rigorous guarantees when $f \in \mathcal{F}_2$.

An emerging direction for solving composite minimization problems (1) is based on the proximal-Newton method. The origins of this method can be traced back to the work of (Bonnans, 1994), which relies on the concept of *strong regularity* introduced by (Robinson, 1980) for generalized equations. In the convex case, this method has been studied by several authors such as (Becker and Fadili, 2012; Lee et al., 2012; Schmidt et al., 2011). So far, methods along this line are applied to solve a generic problem of the form (1) even when $f \in \mathcal{F}_2$. The convergence analysis of these methods is encouraged by standard Newton methods and requires the strong regularity of the Hessian of f near the optimal solution (i.e., $\mu\mathbb{I} \preceq \nabla^2 f(\mathbf{x}) \preceq L\mathbb{I}$). Moreover, the global convergence can only be proved by applying a certain globalization strategy such as line-search or trust-region. Unfortunately, none of these assumptions can be verified before the algorithm execution for the intended applications.

1.3 Our contributions

Interior point methods are always an option while solving the self-concordant composite problems (1) numerically by means of disciplined convex programming (Grant et al., 2006; Löefberg, 2004). More concretely, in the IPM setting, we set up an equivalent problem to (1) that typically avoids the non-smooth term $g(x)$ in the objective by lifting the problem dimensions with slack variables and introducing additional constraints. The new constraints may then be embedded into the objective through a barrier function. We then solve a sequence of smooth problems (e.g., with Newton methods) and "path-follow"[3] to obtain an accurate solution (Nemirovski and Todd, 2009; Nesterov, 2004). In this loop, many of the underlying structures within the original problem, such as sparsity, can be lost due to pre-conditioning or Newton direction scaling (e.g., Nesterov-Todd scaling, Nesterov and Todd (1997)). The efficiency and the memory bottlenecks of the overall scheme then heavily depends on the workhorse algorithm that solves the smooth problems.

In stark contrast, we introduce an algorithmic framework that directly handles the composite minimization problem (1) without increasing the original problem dimensions. For large-scale problems, this is the main argument in favor of our approach. Instead of solving a sequence of smooth problems, we solve a sequence of non-smooth proximal problems with a variable metric (i.e., our workhorse). Fortunately, these proximal problems feature the composite form (1) with a Lipschitz gradient (and oft-times strongly convex) smooth term. Hence, we leverage the tremendous amount of research on large-scale algorithms (cf., Table 1) done over the last decades. Surprisingly, we can even retain the original problem structures that lead to computational ease in many cases (e.g., see Section 4.1).

Our specific contributions can be summarized as follows:

1. We propose a new *variable metric* framework for minimizing the sum $f + g$ of a self-concordant function f and a convex, possibly nonsmooth function g. Our approach relies on the solution of a convex subproblem obtained by linearizing and regularizing the first term f. To achieve monotonic descent, we develop a new set of *analytic* step-size selection and correction procedures based on the structure of the problem.

3. It is also referred to as a homotopy method.

2. We establish both the global and the local convergence of different variable metric strategies. We first derive an expected result: when the variable metric is the Hessian $\nabla^2 f(\mathbf{x}^k)$ of f at iteration k, the resulting algorithm locally exhibits quadratic convergence rate within an explicit region. We then show that variable metrics satisfying the Dennis-Moré-type condition (Dennis and Moré, 1974) exhibit superlinear convergence.

3. We pay particular attention to diagonal variable metrics as many of the proximal subproblems can be solved exactly (i.e., in closed form). We derive conditions on when these variants achieve locally linear convergence.

4. We apply our algorithms to the aforementioned large-scale real-world and synthetic problems to highlight the strengths and the weaknesses of our scheme. For instance, in the graph learning problem (2), our framework can avoid matrix inversions as well as Cholesky decompositions in learning graphs. In Poisson intensity reconstruction (3), up to around $80\times$ acceleration is possible over the state-of-the-art solver.

We highlight three key practical contributions to numerical optimization. First, in the proximal-Newton method, our analytical step-size procedures allow us to do away with any globalization strategy (e.g., line-search). This has a significant practical impact when the evaluation of the functions is expensive. We show how to combine the analytical step-size selection with the standard backtracking or forward line-search procedures to enhance the global convergence of our method. Our analytical quadratic convergence characterization helps us adaptively switch from *damped* step-size to a *full* step-size. Second, in the proximal-gradient method setting, we establish a step-size selection and correction mechanism. The step-size selection procedure can be considered as a predictor, where existing step-size rules that leverage local information can be used. The step-size corrector then adapts the local information of the function to achieve the best theoretical decrease in the objective function. While our procedure does not require any function evaluations, we can further enhance convergence whenever we are allowed function evaluations. Finally, our framework, as we demonstrate in (Tran-Dinh et al., 2013b), accommodates a path-following strategy, which enable us to approximately solve constrained non-smooth convex minimization problems with rigorous guarantees.

Paper outline. In Section 2, we first recall some fundamental concepts of convex optimization and self-concordant functions used in this paper. Section 3 presents our algorithmic framework using three different instances with convergence results, complexity estimates and modifications. Section 4 deals with three concrete instances of our algorithmic framework. Section 5 provides numerical experiments to illustrate the impact of the proposed methods. Section 6 concludes the paper.

2. Preliminaries

Notation: We reserve lower-case and bold lower-case letters for scalar and vector representation, respectively. Upper-case bold letters denote matrices. We denote $\mathbb{S}^p_{++}$ for the set of symmetric positive definite matrices of size $p \times p$. For a proper, lower semicon-

tinuous convex function f from $\mathbb{R}^n$ to $\mathbb{R} \cup \{+\infty\}$, we denote its domain by $\mathrm{dom}(f)$, i.e., $\mathrm{dom}(f) := \{\mathbf{x} \in \mathbb{R}^n \mid f(\mathbf{x}) < +\infty\}$ (see, e.g., Rockafellar (1970)).

Weighted norm and local norm: Given a matrix $\mathbf{H} \in \mathbb{S}^n_{++}$, we define the weighted norm $\|\mathbf{x}\|_{\mathbf{H}} := \sqrt{\mathbf{x}^T\mathbf{H}\mathbf{x}}$, $\forall \mathbf{x} \in \mathbb{R}^n$; its dual norm is defined as $\|\mathbf{x}\|^*_{\mathbf{H}} := \max_{\|\mathbf{y}\|_{\mathbf{H}} \le 1} \mathbf{y}^T\mathbf{x} = \sqrt{\mathbf{x}^T\mathbf{H}^{-1}\mathbf{x}}$. Let $f \in \mathcal{F}_2$ and $\mathbf{x} \in \mathrm{dom}(f)$ so that $\nabla^2 f(\mathbf{x})$ is positive definite. For a given vector $\mathbf{v} \in \mathbb{R}^n$, the local norm around $\mathbf{x} \in \mathrm{dom}(f)$ with respect to f is defined as $\|\mathbf{v}\|_{\mathbf{x}} := \left(\mathbf{v}^T\nabla^2 f(\mathbf{x})\mathbf{v}\right)^{1/2}$, while the corresponding dual norm is given by $\|\mathbf{v}\|^*_{\mathbf{x}} = \left(\mathbf{v}^T\nabla^2 f(\mathbf{x})^{-1}\mathbf{v}\right)^{1/2}$.

Subdifferential and subgradient: Given a proper, lower semicontinuous convex function, we define the subdifferential of g at $\mathbf{x} \in \mathrm{dom}(g)$ as

$$\partial g(\mathbf{x}) := \left\{\mathbf{v} \in \mathbb{R}^n \mid g(\mathbf{y}) - g(\mathbf{x}) \ge \mathbf{v}^T(\mathbf{y} - \mathbf{x}),\ \forall \mathbf{y} \in \mathrm{dom}(g)\right\}.$$

If $\partial g(\mathbf{x}) \neq \emptyset$ then each element in $\partial g(\mathbf{x})$ is called a subgradient of g at $\mathbf{x}$. In particular, if g is differentiable, we use $\nabla g(\mathbf{x})$ to denote its derivative at $\mathbf{x} \in \mathrm{dom}(g)$, and $\partial g(\mathbf{x}) \equiv \{\nabla f(\mathbf{x})\}$.

Proximity operator: A basic tool to handle the nonsmoothness of a convex function g is its proximity operator (or proximal operator) $\mathrm{prox}_g^{\mathbf{H}}$, whose definition is given in Section 1. For notational convenience in our derivations, we alter this definition in the sequel as follows: Let g be a proper lower semicontinuous and convex in $\mathbb{R}^n$ and $\mathbf{H} \in \mathbb{S}^n_{++}$. We define

$$P^g_{\mathbf{H}}(\mathbf{u}) := (\mathbf{H} + \partial g)^{-1}(\mathbf{u}) = \arg\min_{\mathbf{x} \in \mathbb{R}^n} \left\{ g(\mathbf{x}) + \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} - \mathbf{u}^T\mathbf{x} \right\}, \quad \forall \mathbf{u} \in \mathbb{R}^n, \tag{6}$$

as the proximity operator for the nonsmooth g, which has the following properties.

Lemma 2 *The operator $P^g_{\mathbf{H}}$ in* (6) *is single-valued and satisfies the following property:*

$$(P^g_{\mathbf{H}}(\mathbf{u}) - P^g_{\mathbf{H}}(\mathbf{v}))^T(\mathbf{u} - \mathbf{v}) \ge \left\|P^g_{\mathbf{H}}(\mathbf{u}) - P^g_{\mathbf{H}}(\mathbf{v})\right\|^2_{\mathbf{H}}, \tag{7}$$

for all $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$. Consequently, $P^g_{\mathbf{H}}$ is a nonexpansive mapping, i.e.,

$$\left\|P^g_{\mathbf{H}}(\mathbf{u}) - P^g_{\mathbf{H}}(\mathbf{v})\right\|_{\mathbf{H}} \le \|\mathbf{u} - \mathbf{v}\|^*_{\mathbf{H}}. \tag{8}$$

Proof The single-valuedness of $P^g_{\mathbf{H}}$ is obvious due to the strong convexity of the objective function in (6). Let $\boldsymbol{\xi}_{\mathbf{u}} := P^g_{\mathbf{H}}(\mathbf{u})$ and $\boldsymbol{\xi}_{\mathbf{v}} := P^g_{\mathbf{H}}(\mathbf{v})$. By the definition of $P^g_{\mathbf{H}}$, we have $\mathbf{u} - \mathbf{H}\boldsymbol{\xi}_{\mathbf{u}} \in \partial g(\boldsymbol{\xi}_{\mathbf{u}})$ and $\mathbf{v} - \mathbf{H}\boldsymbol{\xi}_{\mathbf{u}} \in \partial g(\boldsymbol{\xi}_{\mathbf{v}})$. Since g is convex, we have $(\mathbf{u} - \mathbf{H}\boldsymbol{\xi}_{\mathbf{u}} - (\mathbf{v} - \mathbf{H}\boldsymbol{\xi}_{\mathbf{v}}))^T(\boldsymbol{\xi}_{\mathbf{u}} - \boldsymbol{\xi}_{\mathbf{v}}) \ge 0$. This inequality leads to $(\mathbf{u} - \mathbf{v})^T(\boldsymbol{\xi}_{\mathbf{u}} - \boldsymbol{\xi}_{\mathbf{v}}) \ge (\boldsymbol{\xi}_{\mathbf{u}} - \boldsymbol{\xi}_{\mathbf{v}})^T\mathbf{H}(\boldsymbol{\xi}_{\mathbf{u}} - \boldsymbol{\xi}_{\mathbf{v}}) = \|\boldsymbol{\xi}_{\mathbf{u}} - \boldsymbol{\xi}_{\mathbf{v}}\|^2_{\mathbf{H}}$ which is indeed (7). Via the generalized Cauchy-Schwarz inequality, (7) leads to (8). ■

Key self-concordant bounds: Based on (Nesterov, 2004, Theorems 4.1.7 and 4.1.8), for a given standard self-concordant function f, we recall the following inequalities

$$\omega(\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + f(\mathbf{x}) \le f(\mathbf{y}), \tag{9}$$

$$f(\mathbf{y}) \le f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y} - \mathbf{x}) + \omega_*(\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}}), \tag{10}$$

where $\omega : \mathbb{R} \to \mathbb{R}_+$ is defined as $\omega(t) := t - \ln(1+t)$ and $\omega_* : [0,1] \to \mathbb{R}_+$ is defined as $\omega_*(t) := -t - \ln(1-t)$. These functions are both nonnegative, strictly convex and increasing. Hence, (9) holds for all $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$, and (10) holds for all $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$ such that $\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} < 1$. In contrast to the "global" inequalities for the function classes $\mathcal{F}_L$ and $\mathcal{F}_\mu$ (cf., Fig. 1), the self-concordant inequalities are based on "local" quantities. Moreover, these bounds are no longer quadratic which prevents naive applications of the methods from $\mathcal{F}_{\mu,L}$.

3. Composite self-concordant optimization

In this section, we propose a *variable metric* optimization framework that rigorously trades off computation and accuracy of solutions without transforming (1) into a higher dimension smooth convex optimization problem. We assume theoretically that the proximal subproblems can be solved exactly. However, in practice, we solve these problems up to a sufficiently high accuracy (typically, it is at least higher than the desired accuracy of (1) at the few last iterations). In our theoretical characterizations, we only rely on the following assumption:

Assumption A.1 *The function f is convex and standard self-concordant (see Definition 1). The function g from $\mathbb{R}^n$ to $\mathbb{R} \cup \{+\infty\}$ is proper, lower semicontinuous, convex and possibly nonsmooth with a tractable proximity operator.*

Unique solvability of (1) and its optimality condition: First, we show that problem (1) is uniquely solvable. The proof of this lemma can be done similarly as (Nesterov, 2004, Theorem 4.1.11) and is provided in the appendix.

Lemma 3 *Suppose that the functions f and g of problem (1) satisfy Assumption 1. Let $\lambda(\mathbf{x}) := \|\nabla f(\mathbf{x}) + \mathbf{v}\|^*_{\mathbf{x}^*} < 1$, for some $\mathbf{x} \in \mathrm{dom}(F)$ and $\mathbf{v} \in \partial g(\mathbf{x})$. Then the solution $\mathbf{x}^*$ of (1) exists and is unique.*

Since this problem is convex, the following optimality condition is necessary and sufficient:

$$\mathbf{0} \in \nabla f(\mathbf{x}^*) + \partial g(\mathbf{x}^*). \tag{11}$$

The solution $\mathbf{x}^*$ is called *strongly regular* if $\nabla^2 f(\mathbf{x}^*) \succ 0$. In this case, $\infty > \sigma^*_{\max} \geq \sigma^*_{\min} > 0$, where $\sigma^*_{\min}$ and $\sigma^*_{\max}$ are the smallest and the largest eigenvalue of $\nabla^2 f(\mathbf{x}^*)$.

Fixed-point characterization: Let $\mathbf{H} \in \mathbb{S}^n_{++}$. We define $S_{\mathbf{H}}(\mathbf{x}) := \mathbf{H}\mathbf{x} - \nabla f(\mathbf{x})$. Then, from (11), we have

$$S_{\mathbf{H}}(\mathbf{x}^*) \equiv \mathbf{H}\mathbf{x}^* - \nabla f(\mathbf{x}^*) \in \mathbf{H}\mathbf{x}^* + \partial g(\mathbf{x}^*).$$

By using the definition of $P^g_{\mathbf{H}}(\cdot)$ in (6), one can easily derive the fixed-point expression

$$\mathbf{x}^* = P^g_{\mathbf{H}}\left(S_{\mathbf{H}}(\mathbf{x}^*)\right), \tag{12}$$

that is, $\mathbf{x}^*$ is the fixed-point of the mapping $R^g_{\mathbf{H}}(\cdot)$, where $R^g_{\mathbf{H}}(\cdot) := P^g_{\mathbf{H}}(S_{\mathbf{H}}(\cdot))$. The formula in (12) suggests that we can generate an iterative sequence based on the fixed-point principle, i.e., $\mathbf{x}^{k+1} := R^g_{\mathbf{H}}(\mathbf{x}^k)$ starting from $\mathbf{x}^0 \in \mathrm{dom}(F)$ for $k \geq 0$. Theoretically, under certain assumptions, one can ensure that the mapping $R^g_{\mathbf{H}}$ is contractive and the sequence generated by this scheme is convergent.

Our variable metric framework: Given a point $\mathbf{x}^k \in \mathrm{dom}(F)$ and a symmetric positive definite matrix $\mathbf{H}_k$, we consider the function

$$Q(\mathbf{x}; \mathbf{x}^k, \mathbf{H}_k) := f(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x} - \mathbf{x}^k) + \frac{1}{2}(\mathbf{x} - \mathbf{x}^k)^T\mathbf{H}_k(\mathbf{x} - \mathbf{x}^k), \tag{13}$$

for $\mathbf{x} \in \mathrm{dom}(F)$. The function $Q(\cdot; \mathbf{x}^k, \mathbf{H}_k)$ is—seemingly—a quadratic approximation of f around $\mathbf{x}^k$. Now, we study the following scheme to generate a sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$:

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k, \tag{14}$$

where $\alpha_k \in (0, 1]$ is a step size and $\mathbf{d}^k$ is a search direction.

Let $\mathbf{s}^k$ be a solution of the following problem:

$$\mathbf{s}^k := \operatorname*{arg\,min}_{\mathbf{x}\in\mathrm{dom}(F)} \Big\{Q(\mathbf{x}; \mathbf{x}^k, \mathbf{H}_k) + g(\mathbf{x})\Big\} = P^g_{\mathbf{H}_k}\left(\mathbf{H}_k\mathbf{x}^k - \nabla f(\mathbf{x}^k)\right). \tag{15}$$

Since $\mathbf{H}_k$ is positive definite, $\mathbf{s}^k$ exists and is unique. The direction $\mathbf{d}^k$ is computed as

$$\mathbf{d}^k := \mathbf{s}^k - \mathbf{x}^k. \tag{16}$$

If we define $\mathbf{G}_k := \mathbf{H}_k\mathbf{d}^k$, then $\mathbf{G}_k$ is called the *gradient mapping* (Nesterov, 2004) which behaves similarly as gradient vectors in non-composite minimization. Since problem (15) is uniquely solvable, we can write its optimality condition as

$$\mathbf{0} \in \nabla f(\mathbf{x}^k) + \mathbf{H}_k(\mathbf{s}^k - \mathbf{x}^k) + \partial g(\mathbf{s}^k). \tag{17}$$

In the variable metric framework, depending on the choice of $\mathbf{H}_k$, the iteration scheme (14) leads to different methods for solving (1). For instance,

1. If $\mathbf{H}_k := \nabla^2 f(\mathbf{x}^k)$, then the method (14) is a *proximal-Newton* method.
2. If $\mathbf{H}_k$ is a symmetric positive definite matrix approximation of $\nabla^2 f(\mathbf{x}^k)$, then the method (14) is a *proximal-quasi Newton* method.
3. If $\mathbf{H}_k := L_k\mathbb{I}$, where L_k is, say, an approximation for the local Lipschitz constant of f and $\mathbb{I}$ is the identity matrix, then the method (14) is a *proximal-gradient* method.

Many of these above methods have been studied for (1) when $f \in \mathcal{F}_L$: cf., (Becker and Fadili, 2012; Beck and Teboulle, 2009a; Chouzenoux et al., 2013; Lee et al., 2012). Note however that, since the self-concordant part f of F is not (necessarily) globally Lipschitz continuously differentiable, these approaches are generally not applicable in theory.

Given the search direction $\mathbf{d}^k$ defined by (16), we define the following proximal-Newton decrement[4] λ_k and the weighted norm β_k:

$$\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k} = \left((\mathbf{d}^k)^T\nabla^2 f(\mathbf{x}^k)\mathbf{d}^k\right)^{1/2} \text{ and } \beta_k := \|\mathbf{d}^k\|_{\mathbf{H}_k}. \tag{18}$$

In the sequel, we study three different instances of the variable metric strategy in detail.

4. This notion is borrowed from standard the Newton decrement defined in (Nesterov, 2004, Chapter 4).

3.1 A proximal-Newton method

If we choose $\mathbf{H}_k := \nabla^2 f(\mathbf{x}^k)$, then the method described in (14) is called the *proximal Newton* algorithm. For notational ease, we redefine $\mathbf{s}_n^k := \mathbf{s}^k$ and $\mathbf{d}_n^k := \mathbf{d}^k$, where the subscript n is used to distinguish proximal Newton related quantities from the other variable metric strategies. Moreover, we use the shorthand notation $P_{\bar{\mathbf{x}}}^g := P_{\nabla^2 f(\bar{\mathbf{x}})}^g$, whenever $\bar{\mathbf{x}} \in \mathrm{dom}(f)$. Using (15) and (16), $\mathbf{s}_n^k$ and $\mathbf{d}_n^k$ are given by

$$\mathbf{s}_n^k := P_{\mathbf{x}^k}^g \left(\nabla^2 f(\mathbf{x}^k)\mathbf{x}^k - \nabla f(\mathbf{x}^k) \right), \quad \mathbf{d}_n^k := \mathbf{s}_n^k - \mathbf{x}^k. \tag{19}$$

Then, the proximal-Newton method generates a sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ starting from $\mathbf{x}^0 \in \mathrm{dom}(F)$ according to

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}_n^k, \tag{20}$$

where $\alpha_k \in (0, 1]$ is a step size. If $\alpha_k < 1$, then the iteration (20) is called the *damped proximal-Newton* iteration. If $\alpha_k = 1$, then it is called the *full-step proximal-Newton* iteration.

Global convergence: We first show that with an appropriate choice of the step-size $\alpha_k \in (0, 1]$, the iterative sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ generated by the damped-step proximal Newton scheme (20) 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. The following theorem provides an explicit formula for the step size α_k whose proof can be found in the appendix.

Theorem 4 *If $\alpha_k := \frac{1}{1+\lambda_k} \in (0, 1]$, then the scheme in (20) generates $\mathbf{x}^{k+1}$ satisfies:*

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\lambda_k). \tag{21}$$

Moreover, the step α_k is optimal. The number of iterations to reach the point $\mathbf{x}^k$ such that $\lambda_k < \sigma$ for some $\sigma \in (0, 1)$ is $k_{\max} := \left\lfloor \frac{F(\mathbf{x}^0) - F(\mathbf{x}^*)}{\omega(\sigma)} \right\rfloor + 1$.

Local quadratic convergence rate: We now establish the local quadratic convergence of the scheme (20). A complete proof of this theorem can be found in the appendix.

Theorem 5 *Let $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ be a sequence generated by the proximal Newton scheme (20) with $\alpha_k \in (0, 1]$. Then:*

a) *If $\alpha_k \lambda_k < 1 - \frac{1}{\sqrt{2}}$, then it holds that*

$$\lambda_{k+1} \leq \left(\frac{1 - \alpha_k + (2\alpha_k^2 - \alpha_k)\lambda_k}{1 - 4\alpha_k\lambda_k + 2\alpha_k^2\lambda_k^2} \right) \lambda_k. \tag{22}$$

b) *If the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ is generated by the damped proximal-Newton scheme (20), starting from $\mathbf{x}^0$ such that $\lambda_0 \leq \bar{\sigma} := \sqrt{5} - 2 \approx 0.236068$ and $\alpha_k := (1 + \lambda_k)^{-1}$, then it locally converges to $\mathbf{x}^*$ at a quadratic rate.*

c) *Alternatively, if the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ is generated by the full-step proximal-Newton scheme (20) starting from $\mathbf{x}^0$ such that $\lambda_0 \leq \bar{\sigma} := 0.25(5 - \sqrt{17}) \approx 0.219224$ and $\alpha_k = 1$, then it locally converges to $\mathbf{x}^*$ at a quadratic rate.*

A two-phase algorithm for solving (1)**:** Now, by the virtue of the above analysis, we can propose a two-phase proximal-Newton algorithm for solving (1). Initially, we perform the dub-step proximal-Newton iterations until we reach the quadratic convergence region (Phase 1). Then, we perform full-step proximal-Newton iterations, until we reach the desired accuracy (Phase 2). The pseudocode of the algorithm is presented in Algorithm 1.

Algorithm 1 (*Proximal-Newton algorithm*)

Inputs: $\mathbf{x}^0 \in \mathrm{dom}(F)$, tolerance $\varepsilon > 0$.

Initialization: Select a constant $\sigma \in (0, \frac{(5-\sqrt{17})}{4}]$, e.g., $\sigma := 0.2$.

for $k = 0$ **to** $K_{\max}$ **do**

1. Compute the proximal-Newton search direction $\mathbf{d}_n^k$ as in (19).
2. Compute $\lambda_k := \left\|\mathbf{d}_n^k\right\|_{\mathbf{x}^k}$.
3. **if** $\lambda_k > \sigma$ **then** $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}_n^k$, where $\alpha_k := (1 + \lambda_k)^{-1}$.
4. **elseif** $\lambda_k > \varepsilon$ **then** $\mathbf{x}^{k+1} := \mathbf{x}^k + \mathbf{d}_n^k$.
5. **else** terminate.

end for

The radius σ of the quadratic convergence region in Algorithm 1 can be fixed at any value in $(0, \bar{\sigma}]$, e.g., at its upper bound $\bar{\sigma}$. An upper bound $K_{\max}$ of the iterations can also be specified, if necessary. The computational bottleneck in Algorithm 1 is typically incurred Step 1 in Phase 1 and Phase 2, where we need to solve the subproblem (15) to obtain a search direction $\mathbf{d}_n^k$. Since problem (15) is strongly convex, one can apply first order methods to efficiently solve this problem with a linear convergence rate (see, e.g., Beck and Teboulle (2009a); Nesterov (2004, 2013)) and make use of a *warm-start* strategy by employing the information of the previous iterations.

Iteration-complexity analysis. The choice of σ in Algorithm 1 can trade-off the number of iterations between the damped-step and full-step iterations. If we fix $\sigma = 0.2$, then the complexity of the full-step Newton phase becomes $\mathcal{O}\left(\ln\ln\left(\frac{0.28}{\varepsilon}\right)\right)$. The following theorem summarizes the complexity of the proposed algorithm.

Theorem 6 *The maximum number of iterations required in Algorithm 1 does not exceed* $K_{\max} := \left\lfloor \frac{F(\mathbf{x}^0)-F(\mathbf{x}^*)}{0.017} \right\rfloor + \left\lfloor 1.5\left(\ln\ln\left(\frac{0.28}{\varepsilon}\right)\right)\right\rfloor + 2$ *provided that* $\sigma = 0.2$ *to obtain* $F(\mathbf{x}^k) - F(\mathbf{x}^*) \leq \varepsilon$*, where* $\mathbf{x}^*$ *is the unique solution of* (1)*.*

Proof Let $\sigma = 0.2$. From the estimate (22) of Theorem 5 we have $\lambda_k \leq (1-4\lambda_k+2\lambda_k^2)^{-1}\lambda_k^2$. Since $\lambda_0 \leq \sigma$, by induction, we can easily show that $\lambda_k \leq (1 - 4\sigma + 2\sigma^2)^{-1}\lambda_k^2 \leq c\lambda_{k-1}^2$, where $c := 3.57$. This implies $\lambda_k \leq c^{2^k-1}\lambda_0^{2^k} \leq c^{2^k-1}\sigma^{2^k}$. The stopping criterion $\lambda_k \leq \varepsilon$ in Algorithm 1 is ensured if $(c\sigma)^{2^k} \leq c\varepsilon$. Since $c\sigma \approx 0.71 < 1$, the last condition leads to $k \geq (\ln 2)^{-1} \ln\left(\frac{-\ln(c\sigma)}{-\ln(c\varepsilon)}\right)$. By using $c = 3.57$, $\sigma = 0.2$ and the fact that $\ln(2)^{-1} < 1.5$, we can show that the last requirement is fulfilled if $k \geq \left\lfloor 1.5\left(\ln\ln\left(\frac{0.28}{\varepsilon}\right)\right)\right\rfloor + 1$. Now, combining the last conclusion and Theorem 4 with noting that $\omega(\sigma) > 0.017$ we obtain the conclusion of Theorem 6. ■

A modification of the proximal-Newton method: In Algorithm 1, if we remove Step 4 and replace analytic step-size selection calculation in Step 3 with a backtracking line-search, then we reach the proximal Newton method of (Lee et al., 2012). Hence, this approach *in practice* might lead to reduced overall computation since our step-size α_k is selected optimally with respect to the worst case problem structures as opposed to the particular instance of the problem. Since the backtracking approach always starts with the full-step, we also do not need to know whether we are within the quadratic convergence region. Moreover, the cost of evaluating the objective at the full-step in certain applications may not be significantly worse than the cost of calculating α_k or may be dominated by the cost of calculating the Newton direction.

In stark contrast to backtracking, our new theory behooves us to propose a new forward line-search procedure as illustrated by Figure 2. The idea is quite simple: we start with the

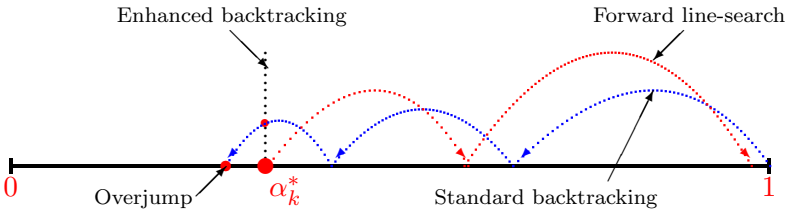


Figure 2: Illustration of step-size selection procedures

"optimal" step-size α_k and increase it towards full-step with a stopping condition based on the objective evaluations. Interestingly, when we analytically calculate the step, we also have access to the side information on whether or not we are within the quadratic convergence region, and hence, we can automatically switch to Step 4 in Algorithm 1. Alternatively, calculation of the analytic step-size can enhance backtracking since the knowledge of α_k reduces the backtracking range from $(0, 1]$ to $(\alpha_k, 1]$ with the side-information as to when to automatically take the full-step without function evaluation.

3.2 A proximal quasi-Newton scheme

Even if the function f is self-concordant, the numerical evaluation of $\nabla^2 f(\mathbf{x})$ can be expensive in many applications. Hence, it is interesting to study proximal quasi-Newton method for solving (1). Our interest in the quasi-Newton methods in this paper is for completeness; we do not provide any algorithmic details or implementations on our quasi-Newton variant.

To this end, we need a symmetric positive definite matrix $\mathbf{H}_k$ that approximates $\nabla^2 f(\mathbf{x}^k)$ at the iteration k. As a result, our main assumption here is that matrix $\mathbf{H}_{k+1}$ at the next iteration $k+1$ satisfies the *secant equation*:

$$\mathbf{H}_{k+1}(\mathbf{x}^{k+1} - \mathbf{x}^k) = \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k). \tag{23}$$

For instance, it is well-known that the sequence of matrices $\{\mathbf{H}_k\}_{k\geq 0}$ updated by the following BFGS formula satisfies the secant equation (23) (Nocedal and Wright, 2006):

$$\mathbf{H}_{k+1} := \mathbf{H}_k + \frac{1}{(\mathbf{y}^k)^T \mathbf{s}^k} \mathbf{y}^k (\mathbf{y}^k)^T - \frac{1}{(\mathbf{s}^k)^T \mathbf{H}_k \mathbf{s}^k} \mathbf{H}_k \mathbf{s}^k (\mathbf{H}_k \mathbf{s}^k)^T, \tag{24}$$

where $\mathbf{s}^k := \mathbf{x}^{k+1} - \mathbf{x}^k$ and $\mathbf{y}^k := \nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k)$. Other methods for updating matrix $\mathbf{H}_k$ can be found in (Nocedal and Wright, 2006), which are not listed here.

In this subsection, we only analyze the full-step proximal quasi-Newton scheme based on the BFGS updates. The global convergence characterization of the BFGS quasi-Newton method can be obtained using our analysis in the next subsection. To this end, we have the following update equation, where the subscript q is used to distinguish the quasi-Newton method:

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \mathbf{d}_q^k. \tag{25}$$

Here we use $\mathbf{d}_q^k$ to stand for the proximal quasi-Newton search direction.

Under certain assumptions, one can prove that the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ generated by (25) converges to $\mathbf{x}^*$ the unique solution of (1). One of the common assumptions used in quasi-Newton methods is the Dennis-Moré condition, see (Dennis and Moré, 1974). Adopting the Dennis-Moré criterion, we impose the following condition in our context:

$$\lim_{k\to\infty} \frac{\left\| \left[\mathbf{H}_k - \nabla^2 f(\mathbf{x}^*)\right](\mathbf{x}^{k+1} - \mathbf{x}^k)\right\|_{\mathbf{x}^*}^*}{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^*}} = 0. \tag{26}$$

Now, we establish the superlinear convergence of the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ generated by (25) as follows:

Theorem 7 *Let matrix $\mathbf{H}_k$ maintains the secant equation (23) and let $\{\mathbf{x}^k\}_{k\geq 0}$ be a sequence generated by scheme (25). Then the following statements hold:*

(a) Suppose, in addition, that the sequence of matrices $\{\mathbf{H}_k\}_{k\geq 0}$ satisfies the Dennis-Moré condition (26) for sufficiently large k. Then the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ converges to the solution $\mathbf{x}^*$ of (1) at a superlinear rate provided that $\|\mathbf{x}^0 - \mathbf{x}^*\|_{\mathbf{x}^*} < 1$.

(b) Suppose that a matrix $\mathbf{H}_0 \succ 0$ is chosen. Suppose further that $\nabla^2 f(\mathbf{x}^k)$ is nonsingular for all $k \geq 0$ (in particular, $\mathrm{dom}(f)$ contains no straight line) then $(\mathbf{y}^k)^T\mathbf{s}^k > 0$ for all $k \geq 0$ and hence the sequence $\{\mathbf{H}_k\}_{k\geq 0}$ generated by (24) is symmetric positive definite and satisfies the secant equation (23). Moreover, if the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ generated by (25) satisfies $\sum_{k=0}^{\infty} \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*} < +\infty$, then this sequence converges to $\mathbf{x}^*$ at a superlinear rate.

The proof of this theorem can be found in the appendix. We note that if the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ locally converges to $\mathbf{x}^*$ at a linear rate w.r.t. the local norm at $\mathbf{x}^*$, i.e. $\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_{\mathbf{x}^*} \leq \kappa \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}$ for some $\kappa \in (0,1)$ and $k \geq 0$, then the condition $\sum_{k=0}^{\infty} \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*} < +\infty$ automatically holds. From (26) we also observe that the matrix $\mathbf{H}_k$ is required to well approximate $\nabla^2 f(\mathbf{x}^*)$ along the direction $\mathbf{d}_q^k$, which is not in the whole space.

3.3 A proximal-gradient method

If we choose matrix $\mathbf{H}_k := \mathbf{D}_k$, where $\mathbf{D}_k$ is a positive diagonal matrix, then the iterative scheme (14) is called the *proximal-gradient* scheme. In this case, we can write (14) as

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}_g^k = (1 - \alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}_g^k, \tag{27}$$

where $\alpha_k \in (0,1]$ is an appropriate step size, $\mathbf{d}_g^k$ is the proximal-gradient search direction and $\mathbf{s}_g^k \equiv \mathbf{s}^k$ as in (15).

The following lemma shows how we can choose the step size α_k corresponding to $\mathbf{D}_k$ such that we obtain a descent direction in the proximal-gradient scheme (27). The proof of this lemma can be found in the appendix.

Lemma 8 *Let $\{\mathbf{x}^k\}_{k\geq 0}$ be a sequence generated by (27). Suppose that the matrix $\mathbf{D}_k \succ 0$ is chosen such that the step size α_k satisfies $\alpha_k := \frac{\beta_k^2}{\lambda_k(\lambda_k+\beta_k^2)} \in (0,1]$ (see below), where $\beta_k := \|\mathbf{d}_g^k\|_{\mathbf{D}_k}$ and $\lambda_k := \|\mathbf{d}_g^k\|_{\mathbf{x}^k}$. Then $\{\mathbf{x}^k\}_{k\geq 0} \subset \mathrm{dom}(F)$ and*

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega\left(\frac{\beta_k^2}{\lambda_k}\right). \tag{28}$$

Moreover, the step-size α_k as defined above is optimal.

From Lemma 8, we observe that $\alpha_k \leq 1$ if $\frac{\lambda_k^2}{\beta_k^2} + \lambda_k \geq 1$. It is obvious that if $\lambda_k \geq 1$ then the last condition is automatically satisfied. We only consider the case $\lambda_k < 1$. If the sequence $\{\mathbf{x}^k\}_{k\geq 0}$ is convergent, then $\lambda_k \to 0^+$ as $k \to \infty$. Then, we relax actually the condition $\frac{\lambda_k^2}{\beta_k^2} + \lambda_k \geq 1$ to $\lambda_k \geq \beta_k$.

We now study the case $\mathbf{D}_k := L_k\mathbb{I}$, where $L_k \geq \underline{L} > 0$ is a positive constant and $\mathbb{I}$ is the identity matrix with dimensions apparent from the context. Hence, $\beta_k^2 = L_k\|\mathbf{d}_g^k\|_2^2$ and

$$\frac{\lambda_k^2}{\beta_k^2} = \frac{(\mathbf{d}_g^k)^T\nabla^2 f(\mathbf{x}^k)\mathbf{d}_g^k}{L_k\|\mathbf{d}_g^k\|_2^2}.$$

However, since

$$\sigma_{\min}(\nabla^2 f(\mathbf{x}^k)) \leq \sigma^k := \frac{(\mathbf{d}_g^k)^T\nabla^2 f(\mathbf{x}^k)\mathbf{d}_g^k}{\|\mathbf{d}_g^k\|_2^2} \leq \sigma_{\max}(\nabla^2 f(\mathbf{x}^k)), \tag{29}$$

the condition $\lambda_k \geq \beta_k$ is equivalent to

$$L_k \leq \sigma_k, \tag{30}$$

where $\sigma_{\min}^k := \sigma_{\min}(\nabla^2 f(\mathbf{x}^k))$ and $\sigma_{\max}^k := \sigma_{\max}(\nabla^2 f(\mathbf{x}^k))$ are the smallest and largest eigenvalue of $\nabla^2 f(\mathbf{x}^k)$, respectively. Under the assumption that $\mathrm{dom}(f)$ contains no straight-line, then we have the Hessian $\nabla^2 f(\mathbf{x}^k) \succ 0$ by (Nesterov, 2004, Theorem 4.1.3), which implies that $\sigma_{\min}^k > 0$. Therefore, in the worst-case, we can choose $L_k := \sigma_{\min}^k$. However, this lower bound may be too conservative. In practice, we can apply a *bisection procedure* to meet the condition (30). It is not difficult to prove via contradiction that the number of bisection steps is upper bounded by a constant.

We note that if g is separable, i.e. $g(\mathbf{x}) := \sum_{i=1}^n g_i(\mathbf{x}_i)$ (e.g. $g(\mathbf{x}) := \rho\,\|\mathbf{x}\|_1$), then we can compute $\mathbf{s}_{\mathbf{D}_k}^k$ in (15) in a component-wise fashion as:

$$(\mathbf{s}_{L_k}^k)_i := \mathcal{P}_{\tau_i^k}^{g_i}\left(\mathbf{x}_i^k - \tau_i^k(\nabla f(\mathbf{x}^k))_i\right),\ i = 1,\ldots,n, \tag{31}$$

where $\tau_i^k := 1/(\mathbf{D}_k)_{ii}$ and $\mathcal{P}_{\tau_i}^{g_i} := (\mathbf{I} + \tau_i \partial g_i)^{-1}$ is the resolvent of g_i. The computation of λ_k only requires one matrix-vector multiplication and one vector inner-product; but it can be reduced by exploiting concrete structure of the smooth part f.

Based on Lemma 8, we describe the proximal-gradient scheme (27) in Algorithm 2. The main computation cost of Algorithm 2 is incurred at Step 2 and in calculating λ_k. If g is separable, then the computation of Step 2 can be done in a *closed form*. One main step

Algorithm 2 (*Proximal-gradient method*)

Inputs: $\mathbf{x}^0 \in \mathrm{dom}(F)$, tolerance $\varepsilon > 0$.

for $k = 0$ **to** $k_{\max}$ **do**

1. Choose an appropriate $\mathbf{D}_k \succ 0$ based on (30).
2. Compute $\mathbf{d}_g^k := \mathcal{P}_{\mathbf{D}_k}^g \left(\mathbf{D}_k \mathbf{x}^k - \nabla f(\mathbf{x}^k)\right) - \mathbf{x}^k$ due to (15).
3. Compute $\beta_k := \|\mathbf{d}_g^k\|_{\mathbf{D}_k}$ and $\lambda_k := \|\mathbf{d}_g^k\|_{\mathbf{x}^k}$.
4. If $e_k := \|\mathbf{d}_g^k\|_2 \leq \varepsilon$ then terminate.
5. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}_g^k$, where $\alpha_k := \frac{\beta_k^2}{\lambda_k(\lambda_k + \beta_k^2)} \in (0, 1]$.

end for

of Algorithm 2 is Step 2, which depends on the cost of prox-operator $\mathcal{P}_{\mathbf{D}_k}^g$. In practice, $\mathbf{D}_k$ is determined by a bisection procedure whenever $\lambda_k < 1$, which requires additional computational cost. Our bisection procedure does not require objective evaluations, which is usually needed in standard back-tracking line-search. We note that computing λ_k at Step 3 does not need to form the full Hessian $\nabla^2 f(\mathbf{x}^k)$, it only requires a directional derivative, which is relatively cheap in applications (Nocedal and Wright, 2006, Chapter 7).

Global and local convergence. The global and local convergence of Algorithm 2 is stated in the following theorems, whose proof can be found in the appendix.

Theorem 9 *Assume that there exists $\underline{L} > 0$ such that $\mathbf{D}_k \succeq \underline{L}\mathbb{I}$ for $k \geq 0$. Let*

$$\mathcal{L}_F(F(\mathbf{x}^0)) := \left\{\mathbf{x} \in \mathrm{dom}(F) \mid F(\mathbf{x}) \leq F(\mathbf{x}^0)\right\}$$

be bounded from below. Then, the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$, generated by Algorithm 2, converges to the unique solution $\mathbf{x}^*$ of (1).

Theorem 10 *Let $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ be the sequence generated by Algorithm 2. Then, for k sufficiently large, if*

$$\frac{\left\|[\mathbf{D}_k - \nabla^2 f(\mathbf{x}^*)]\mathbf{d}_g^k\right\|_{\mathbf{x}^*}^*}{\|\mathbf{d}_g^k\|_{\mathbf{x}^*}} < \frac{1}{2}, \tag{32}$$

then $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ locally converges to $\mathbf{x}^*$ at a linear rate. In particular, if $\mathbf{D}_k := L_k\mathbb{I}$ and $\gamma_* := \max\left\{\left|1 - \frac{L_k}{\sigma_{\min}^*}\right|, \left|1 - \frac{L_k}{\sigma_{\max}^*}\right|\right\} < \frac{1}{2}$ then the condition (32) holds.

We note that $\mathbf{x}^*$ is *unknown*; thus, evaluating γ_* apriori is infeasible in reality. In implementation, one can choose an appropriate value $L_k \geq \underline{L} > 0$ and then adaptively

update L_k based on the knowledge of the eigenvalues of $\nabla^2 f(\mathbf{x}^k)$ near to the solution $\mathbf{x}^*$. Note that the last condition in Theorem 10 leads to $\sigma^*_{\max} < 3\sigma^*_{\min}$. While this seems too imposing, we claim that, for most f and g , this requirement is not too difficult to satisfy (see also the empirical evidence in Subsection 5.2.1) by using the condition (32). The condition (32) can be referred to as a **restricted** approximation gap along the direction $\mathbf{d}_g^k$ for k sufficiently large. For instance, when g is based on the ℓ_1-norm/the nuclear norm, the search direction $\mathbf{d}_g^k$ have at most twice the sparsity/rank of $\mathbf{x}^*$ near the convergence region. Given a subspace generated by all the directions $\mathbf{d}_g^k$, one can prove, via probabilistic assumptions on f that the restricted condition (32) is satisfied with a high probability.

Remark 11 *From the scheme* (27) *we observe that the step size* $\alpha_k < 1$ *may not preserve some of the desiderata on* $\mathbf{x}^{k+1}$ *due to the closed form solution of the prox-operator* $\mathcal{P}^g_{\mathbf{D}_k}$. *For instance, when* g *is based on the* ℓ_1*-norm,* $\alpha_k < 1$, *might increase the sparsity level of the solution as opposed to monotonically increasing it. However, in practice, the numerical values of* α_k *are often* 1 *near the convergence, which maintain properties, such as sparsity, low-rankedness, etc.*

A modification of the proximal-gradient method: If the point $\mathbf{s}_g^k$ generated by (15) belongs to $\mathrm{dom}(F)$, then $F(\mathbf{s}_g^k) < +\infty$. Similarly to the definition of $\mathbf{x}^{k+1}$ in (27), we can define a new trial point

$$\hat{\mathbf{x}}^k := (1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}_g^k. \tag{33}$$

If $F(\mathbf{s}_g^k) \leq F(\mathbf{x}^k)$, then, by the convexity of F, it is easy to show that

$$F(\hat{\mathbf{x}}^k) = F\left((1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}_g^k\right) \leq (1-\alpha_k)F(\mathbf{x}^k) + \alpha_k F(\mathbf{s}_g^k) \overset{F(\mathbf{s}_g^k)\leq F(\mathbf{x}^k)}{\leq} F(\mathbf{x}^k).$$

In this case, based on the function values $F(\mathbf{s}_g^k)$, $F(\hat{\mathbf{x}}^k)$ and $F(\mathbf{x}^k)$ we can eventually choose

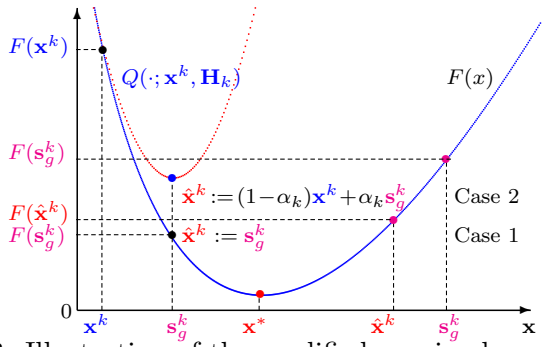


Figure 3: Illustration of the modified proximal-gradient method

the next iteration $\mathbf{x}^{k+1}$ as follows:

$$\mathbf{x}^{k+1} := \begin{cases} \mathbf{s}_g^k & \text{if } \mathbf{s}^k \in \mathrm{dom}(F) \text{ and } F(\mathbf{s}_g^k) < F(\hat{\mathbf{x}}^k) \quad \text{(Case 1)}, \\ \hat{\mathbf{x}}^k & \text{otherwise} \qquad\qquad\qquad\qquad\quad \text{(Case 2)}. \end{cases} \tag{34}$$

The idea of this *greedy* modification is illustrated in Figure 3. We note that here we need to check $\mathbf{s}_g^k \in \mathrm{dom}(F)$ such that $F(\mathbf{s}_g^k) < F(\mathbf{x}^k)$ and additional function evaluations $F(\mathbf{s}_g^k)$ and

$F(\hat{\mathbf{x}}^k)$. However, careful implementations can recycle quantities that enable us to evaluate the objective at $\mathbf{s}_g^k$ and at $\mathbf{x}^{k+1}$ with very little overhead over the calculation of α_k. By using (34), we can specify a modified proximal gradient algorithm for solving (1), whose details we omit here since it is quite similar to Algorithm 2.

4. Concrete instances of our optimization framework

We illustrate three instances of our framework for some of the applications described in Section 1. For concreteness, we describe only the first and second order methods. Quasi-Newton methods based on (L-)BFGS updates or other adaptive variable metrics can be similarly derived in a straightforward fashion.

4.1 Graphical model selection

We customize our optimization framework to solve the graph selection problem (2). For notational convenience, we maintain a matrix variable $\boldsymbol{\Theta}$ instead of vectorizing it. We observe that $f(\boldsymbol{\Theta}) := -\log(\det(\boldsymbol{\Theta})) + \operatorname{tr}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Theta})$ is a standard self-concordant function, while $g(\boldsymbol{\Theta}) := \lambda \left\|\mathbf{vec}(\boldsymbol{\Theta})\right\|_1$ is convex and nonsmooth. The gradient and the Hessian of f can be computed explicitly as $\nabla f(\boldsymbol{\Theta}) := \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Theta}^{-1}$ and $\nabla^2 f(\boldsymbol{\Theta}) := \boldsymbol{\Theta}^{-1} \otimes \boldsymbol{\Theta}^{-1}$, respectively. Next, we formulate our proposed framework to construct two algorithmic variants for (2).

4.1.1 Dual proximal-Newton algorithm

We consider a second order algorithm via a dual solution approach for (15). This approach is first introduced in our earlier work (Tran-Dinh et al., 2013a), which did not consider the new modifications we propose in Section 3.1.

We begin by deriving the following dual formulation of the convex subproblem (15). Let $\mathbf{p}_k := \nabla f(\mathbf{x}^k)$, the convex subproblem (15) can then be written equivalently as

$$\min_{\mathbf{x}\in\mathbb{R}^n}\left\{\frac{1}{2}\mathbf{x}^T\mathbf{H}_k\mathbf{x} + (\mathbf{p}_k - \mathbf{H}_k\mathbf{x}^k)^T\mathbf{x} + g(\mathbf{x})\right\}. \tag{35}$$

By using the min-max principle, we can write (35) as

$$\max_{\mathbf{u}\in\mathbb{R}^n}\min_{\mathbf{x}\in\mathbb{R}^n}\left\{\frac{1}{2}\mathbf{x}^T\mathbf{H}_k\mathbf{x} + (\mathbf{p}_k - \mathbf{H}_k\mathbf{x}^k)^T\mathbf{x} + \mathbf{u}^T\mathbf{x} - g^*(\mathbf{u})\right\}, \tag{36}$$

where g^* is the Fenchel conjugate function of g, i.e. $g^*(\mathbf{u}) := \sup_{\mathbf{x}}\left\{\mathbf{u}^T\mathbf{x} - g(\mathbf{x})\right\}$. Solving the inner minimization in (36) we obtain

$$\min_{\mathbf{u}\in\mathbb{R}^n}\left\{\frac{1}{2}\mathbf{u}^T\mathbf{H}_k^{-1}\mathbf{u} + \tilde{\mathbf{p}}_k^T\mathbf{u} + g^*(\mathbf{u})\right\}, \tag{37}$$

where $\tilde{\mathbf{p}}_k := \mathbf{H}_k^{-1}\mathbf{p}_k - \mathbf{x}^k$. Note that the objective function $\varphi(\mathbf{u}) := g^*(\mathbf{u}) + \frac{1}{2}\mathbf{u}^T\mathbf{H}_k^{-1}\mathbf{u} + \tilde{\mathbf{p}}_k^T\mathbf{u}$ of (37) is strongly convex, one can apply the fast projected gradient methods with a linear convergence rate for solving this problem, see (Nesterov, 2013; Beck and Teboulle, 2009a).

In order to recover the solution of the primal subproblem (15), we note that the solution of the parametric minimization problem in (36) is given by $\mathbf{x}^*(\mathbf{u}) := \mathbf{x}^k - \mathbf{H}_k^{-1}(\mathbf{p}_k + \mathbf{u})$. Let

$\mathbf{u}^*_{\mathbf{x}^k}$ be the optimal solution of (37). We can recover the primal proximal-Newton search direction $\mathbf{d}^k$ of the subproblem (15) as

$$\mathbf{d}_n^k = -\nabla^2 f(\mathbf{x}^k)^{-1}\left(\nabla f(\mathbf{x}^k) + \mathbf{u}^*_{\mathbf{x}^k}\right). \tag{38}$$

To compute the quantity λ_k defined by (18) in Algorithm 1, we use (38) such that

$$\lambda_k = \|\mathbf{d}_n^k\|_{\mathbf{x}^k} = \left\|\nabla f(\mathbf{x}^k) + \mathbf{u}^*_{\mathbf{x}^k}\right\|^*_{\mathbf{x}^k}. \tag{39}$$

Note that computing λ_k by (39) requires the inverse of the Hessian matrix $\nabla^2 f(\mathbf{x}^k)$.

Surprisingly, this dual approach allows us to avoid matrix inversion as well as Cholesky decomposition in computing the gradient $\nabla f(\mathbf{\Theta}_i)$ and the Hessian $\nabla^2 f(\mathbf{\Theta}_i)$ of f in graph selection. An alternative is of course to solve (15) in its primal form. Though, in such case, we need to compute $\mathbf{\Theta}_i^{-1}$ at each iteration i (say, via Cholesky decompositions).

The dual subproblem (37) becomes as:

$$\mathbf{U}^* = \underset{\|\mathbf{vec}(\mathbf{U})\|_\infty \le 1}{\arg\min} \left\{ \frac{1}{2}\mathrm{tr}((\mathbf{\Theta}_i\mathbf{U})^2) + \mathrm{tr}(\widetilde{\mathbf{Q}}\mathbf{U}) \right\}, \tag{40}$$

for the graph selection, where $\widetilde{\mathbf{Q}} := \rho^{-1}[\mathbf{\Theta}_i\widehat{\mathbf{\Sigma}}\mathbf{\Theta}_i - 2\mathbf{\Theta}_i]$. Given the dual solution $\mathbf{U}^*$ of (40), the primal proximal-Newton search direction (i.e. the solution of (15)) is computed as

$$\mathbf{\Delta}_i := -\left((\mathbf{\Theta}_i\widehat{\mathbf{\Sigma}} - \mathbb{I})\mathbf{\Theta}_i + \rho\mathbf{\Theta}_i\mathbf{U}^*\mathbf{\Theta}_i\right). \tag{41}$$

The quantity λ_i defined in (39) can be computed as follows, where $\mathbf{W}_i := \mathbf{\Theta}_i(\widehat{\mathbf{\Sigma}} + \rho\mathbf{U}^*)$:

$$\lambda_i := \left(p - 2\cdot\mathrm{tr}\left(\mathbf{W}_i\right) + \mathrm{tr}\left(\mathbf{W}_i^2\right)\right)^{1/2}. \tag{42}$$

Algorithm 3 summarizes the description above. Overall, this proximal-Newton (PN) algorithm *does not require any matrix inversions or Cholesky decompositions.* It only needs

Algorithm 3 (*Dual PN for graph selection* (`DPNGS`))

Input: Matrix $\widehat{\mathbf{\Sigma}} \succ 0$ and a given tolerance $\varepsilon > 0$. Set $\sigma := 0.25(5 - \sqrt{17})$.
Initialization: Find a starting point $\mathbf{\Theta}_0 \succ 0$.
for $i = 0$ **to** $i_{\max}$ **do**
1. Set $\widetilde{\mathbf{Q}} := \rho^{-1}\left(\mathbf{\Theta}_i\widehat{\mathbf{\Sigma}}\mathbf{\Theta}_i - 2\mathbf{\Theta}_i\right)$.
2. Compute $\mathbf{U}^*$ in (40).
3. Compute λ_i by (42), where $\mathbf{W}_i := \mathbf{\Theta}_i(\widehat{\mathbf{\Sigma}} + \rho\mathbf{U}^*)$.
4. If $\lambda_i \le \varepsilon$ terminate.
5. Compute $\mathbf{\Delta}_i := -\left((\mathbf{\Theta}_i\widehat{\mathbf{\Sigma}} - \mathbb{I})\mathbf{\Theta}_i + \rho\mathbf{\Theta}_i\mathbf{U}^*\mathbf{\Theta}_i\right)$.
6. If $\lambda_i > \sigma$, then set $\alpha_i := (1 + \lambda_i)^{-1}$. Otherwise, set $\alpha_i = 1$.
7. Update $\mathbf{\Theta}_{i+1} := \mathbf{\Theta}_i + \alpha_i\mathbf{\Delta}_i$.

end for

matrix-vector and matrix-matrix calculations, which might be attractive for different computational platforms (such as GPUs or simple parallel implementations). Note however that as we work through the dual problem, the primal solution can be dense even if majority of the entries are rather small (e.g., smaller than 10^{-6}).[5]

We now explain the underlying costs of each step in Algorithm 3, which is useful when we consider different strategies for the selection of the step size α_k. The computation of $\widetilde{\mathbf{Q}}$ and $\mathbf{\Delta}_i$ 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 $\mathcal{O}(p^2)$ complexity. We note here that, while $\mathbf{W}_i$ is a *dense* matrix, the trace operation in the latter case requires only the computation of the diagonal elements of $\mathbf{W}_i^2$. Given $\mathbf{\Theta}_i$, α_i and $\mathbf{\Delta}_i$, the calculation of $\mathbf{\Theta}_{i+1}$ has $\mathcal{O}(p^2)$ complexity. In contrast, evaluation of the objective can be achieved through Cholesky decompositions, which has $\mathcal{O}(p^3)$ time complexity.

To compute (40), we can use the fast proximal-gradient method (FPGM) (Nesterov, 2013; Beck and Teboulle, 2009a) with step size $1/L$ where L is the Lipschitz constant of the gradient of the objective function in (40). It is easy to observe that $L := \gamma^2_{\max}(\mathbf{\Theta}_i)$ where $\gamma_{\max}(\mathbf{\Theta}_i)$ is the largest eigenvalue of $\mathbf{\Theta}_i$. For sparse $\mathbf{\Theta}_i$, we can approximately compute $\gamma_{\max}(\mathbf{\Theta}_i)$ is $O(p^2)$ 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 $O(p^2)$ time. Since FPGM requires a constant number of iterations $k_{\max}$ (independent of p) to achieve an ε_{in} solution accuracy, the time-complexity for the solution in (40) is $O(k_{\max}M)$, where M is the cost of matrix multiplication. We have also implemented block coordinate descent and active set methods which scale $O(p^2)$ in practice when the solution is quite sparse.

Overall, the major operation with general proximal maps in the algorithm is typically the matrix-matrix multiplications of the form $\mathbf{\Theta}_i\mathbf{U}\mathbf{\Theta_i}$, where $\mathbf{\Theta}_i$ and $\mathbf{U}$ are symmetric positive definite. This operation can naturally be computed (e.g., in a GPU) in a parallel or distributed manner. For more details of such computations we refer the reader to (Bertsekas and Tsitsiklis, 1989). It is important to note that without Cholesky decompositions used in objective evaluations, the basic DPNGS approach theoretically scales with the cost of matrix-matrix multiplications.

4.1.2 Proximal-gradient algorithm

Since $g(\mathbf{\Theta}) := \rho\,\|\mathbf{vec}(\mathbf{\Theta})\|_1$ and $\nabla f(\mathbf{\Theta}_i) = \mathbf{vec}(\widehat{\mathbf{\Sigma}} - \mathbf{\Theta}_i^{-1})$, the subproblem (15) becomes

$$\mathbf{\Delta}_{i+1} := \mathcal{T}_{\tau_i\rho}\left(\mathbf{\Theta}_i - \tau_i(\widehat{\mathbf{\Sigma}} - \mathbf{\Theta}_i^{-1})\right) - \mathbf{\Theta}_i, \tag{43}$$

where $\mathcal{T}_\tau : \mathbb{R}^{p\times p} \to \mathbb{R}^{p\times p}$ is the component-wise matrix thresholding operator which is defined as $\mathcal{T}_\tau(\mathbf{\Theta}) := \max\{0, |\mathbf{\Theta}| - \tau\}$. We also note that the computation of $\mathbf{\Delta}_{i+1}$ requires a matrix inversion $\mathbf{\Theta}_i^{-1}$. Since $\mathbf{\Theta}_i$ is positive definite, one can apply Cholesky decompositions to compute $\mathbf{\Theta}_i^{-1}$ in $O(p^3)$ operations. To compute the quantity λ_i, we have $\lambda_i := \|\mathbf{\Delta}_i\|_{\mathbf{\Theta}_i} = \left\|\mathbf{\Theta}_i^{-1}\mathbf{\Delta}_i\right\|_2$. We also choose $L_i := 0.5\left\|\nabla^2 f(\mathbf{\Theta}_i)\right\|_2 = 0.5\left\|\mathbf{\Theta}_i^{-1}\right\|_2^2$. The above are summarized in Algorithm 4.

5. In our MATLAB code, we made no attempts to sparsify of the primal solution. The overall efficiency can be improved via thresholding tricks, both in terms of time-complexity (e.g., less number of iterations) and matrix estimation quality.

Algorithm 4 (*Proximal-gradient method for graph selection* (`ProxGrad1`))

Initialization: Choose a starting point $\boldsymbol{\Theta}_0 \succ 0$.
for $i = 0$ **to** $i_{\max}$ **do**
1. Compute $\boldsymbol{\Theta}_i^{-1}$ via Cholesky decomposition.
2. Choose L_i satisfying (30) and set $\tau_i := L_i^{-1}$.
3. Compute the search direction $\boldsymbol{\Delta}_i$ as (43).
4. Compute $\beta_i := L_i \left\|\mathbf{vec}(\boldsymbol{\Delta}_i)\right\|_2$ and $\lambda_i := \left\|\boldsymbol{\Theta}_i^{-1}\boldsymbol{\Delta}_i\right\|_2$.
5. Determine the step size $\alpha_i := \frac{\beta_i}{\lambda_i(\lambda_i+\beta_i)}$.
6. Update $\boldsymbol{\Theta}_{i+1} := \boldsymbol{\Theta}_i + \alpha_i\boldsymbol{\Delta}_i$.

end for

The per iteration complexity is dominated by matrix-matrix multiplications and Cholesky decompositions for matrix inversion calculations. In particular, Step 1 requires a Cholesky decomposition with $O(p^3)$ time-complexity. Step 2 requires to compute ℓ_2-norm of a symmetric positive matrix, which can be done by a power-method in $O(p^2)$ time-complexity. The complexity of Steps 3, 4 and 6 requires $O(p^2)$ operations. Step 2 may require additional bisection steps as mentioned in Algorithm 2 whenever $\lambda_k < 1$.

4.2 Poisson intensity reconstruction

We now describe a variant of Algorithm 2; a similar instance based on Algorithm 1 can be easily devised and we omit the details here. First, we can easily check that the function $\tilde{f}(\mathbf{x}) := \sum_{i=1}^m \left(\mathbf{a}_i^T\mathbf{x} - y_i \log(\mathbf{a}_i^T\mathbf{x})\right)$ in (3) is convex and self-concordant with parameter $M_{\tilde{f}} := 2 \cdot \max\left\{\frac{1}{\sqrt{y_i}} \mid y_i > 0, i = 1, \ldots, m\right\}$. We define the functions f and g as:

$$f(\mathbf{x}) := \frac{M_{\tilde{f}}^2}{4}\tilde{f}(\mathbf{x}), \quad g(\mathbf{x}) := \frac{M_{\tilde{f}}^2}{4}\left(\rho\phi(\mathbf{x}) + \delta_{\{\mathbf{u} \mid \mathbf{u}\geq 0\}}(\mathbf{x})\right), \tag{44}$$

where f and g satisfy Assumption 1. Thus, the problem in (3) can be equivalently transformed into (1). Here, the gradient and the Hessian of f satisfy:

$$\nabla f(\mathbf{x}) = \frac{M_{\tilde{f}}^2}{4}\sum_{i=1}^m \left(1 - \frac{y_i}{\mathbf{a}_i^T\mathbf{x}}\right)\mathbf{a}_i \quad \text{and} \quad \nabla^2 f(\mathbf{x}) = \frac{M_{\tilde{f}}^2}{4}\sum_{i=1}^m \frac{y_i}{(\mathbf{a}_i^T\mathbf{x})^2}\mathbf{a}_i\mathbf{a}_i^T, \tag{45}$$

respectively. For a given vector $\mathbf{d} \in \mathbb{R}^n$, the local norm $\|\mathbf{d}\|_{\mathbf{x}}$ can then be written as:

$$\|\mathbf{d}\|_{\mathbf{x}} := \left(\mathbf{d}^T\nabla^2 f(\mathbf{x})\mathbf{d}\right)^{1/2} = \frac{M_{\tilde{f}}}{2}\left(\sum_{i=1}^m \frac{y_i(\mathbf{a}_i^T\mathbf{d})^2}{(\mathbf{a}_i^T\mathbf{x})^2}\right)^{1/2}. \tag{46}$$

Computing this quantity requires one matrix-vector multiplication and $\mathcal{O}(m)$ operations.

For the Poisson model, the subproblem (15) is expressed as follows:

$$\min_{\mathbf{x}\geq 0}\left\{\frac{1}{2}\|\mathbf{x} - \mathbf{w}^k\|_2^2 + \rho_k\phi(\mathbf{x})\right\}, \tag{47}$$

where $\mathbf{w}^k := \mathbf{x}^k - L_k^{-1}\nabla f(\mathbf{x}^k)$ and $\rho_k := \frac{\rho M_{\tilde{f}}^2}{4L_k}$. As a penalty function ϕ in the Poisson intensity reconstruction, we use the Total Variation-norm (TV-norm), defined as $\phi(\mathbf{x}) := \|\mathbf{D}\mathbf{x}\|_1$ (isotropic) or $\phi(\mathbf{x}) := \|\mathbf{D}\mathbf{x}\|_{1,2}$ (anti-isotropic), where $\mathbf{D}$ is a forward linear operator (Chambolle and Pock, 2011; Beck and Teboulle, 2009b). For both TV-norm regularizers, the method proposed in (Beck and Teboulle, 2009b) can solve (47) efficiently.

The above discussion leads to Algorithm 5. We note that the constant L_k at Step 2 of this algorithm can be estimated based on different rules. In our implementation below, we initialize L_k at a Barzilai-Borwein step size, i.e., $L_k := \frac{(\nabla f(\mathbf{x}^k)-\nabla f(\mathbf{x}^{k-1}))^T(\mathbf{x}^k-\mathbf{x}^{k-1})}{\|\mathbf{x}^k-\mathbf{x}^{k-1}\|_2^2}$ and may perform a few backtracking iterations on L_k to ensure the condition (30) whenever $\lambda_k < 1$.

Algorithm 5 (`ProxGrad` *for Poisson intensity reconstruction* (`ProxGrad2`))

Inputs: $\mathbf{x}^0 \geq 0$, $\varepsilon > 0$ and $\rho > 0$.
Compute $M_{\tilde{f}} := 2\max\left\{\frac{1}{\sqrt{\mathbf{y}_i}} \mid \mathbf{y}_i > 0, i = 1, \ldots, m\right\}$.
for $k = 0$ **to** $k_{\max}$ **do**

1. Evaluate the gradient of f as (45).
2. Compute an appropriate value $L_k > 0$ that satisfies (30).
3. Compute $\rho_k := 0.25\rho M_{\tilde{f}}^2 L_k^{-1}$ and $\mathbf{w}^k := \mathbf{x}^k - L_k^{-1}\nabla f(\mathbf{x}^k)$.
4. Compute $\mathbf{s}_g^k$ by solving (47) and then compute $\mathbf{d}_g^k := \mathbf{s}_g^k - \mathbf{x}^k$.
5. Compute $\beta_k := L_k\|\mathbf{d}_g^k\|_2^2$ and $\lambda_k := \|\mathbf{d}_g^k\|_{\mathbf{x}^k}$ as (46).
6. If $e_k := L_k^{-1}\sqrt{\beta_k} \leq \varepsilon$ then terminate.
7. Determine the step size $\alpha_k := \frac{\beta_k}{\lambda_k(\lambda_k+\beta_k)}$.
8. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k\mathbf{d}_g^k$.

end for

Note that we can modify Step 8 in Algorithm 5 by using the update scheme (34) to obtain a new variant of this algorithm. We omit the details here.

4.3 Heteroscedastic LASSO

We focus on a convex formulation of the unconstrained LASSO problem with unknown variance studied in (Städler et al., 2012) as:

$$(\boldsymbol{\beta}^*, \sigma^*) := \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p, \sigma\in\mathbb{R}_{++}} \left\{ -\log(\sigma) + \frac{1}{2n}\|\mathbf{X}\boldsymbol{\beta} - \sigma\mathbf{y}\|_2^2 + \rho\|\boldsymbol{\beta}\|_1 \right\}. \tag{48}$$

However, our algorithm can be applied to solve the multiple unknown variance case considered in (Dalalyan et al., 2013).

By letting $\mathbf{x} := (\boldsymbol{\beta}^T, \sigma)^T \in \mathbb{R}^{p+1}$, $f(\mathbf{x}) := -\log(\sigma) + \frac{1}{2n}\|\mathbf{X}\boldsymbol{\beta} - \sigma\mathbf{y}\|_2^2$. Then, it is easy to see that the function f is standard self-concordant. Hence, we can apply Algorithm 2 to solve this problem. To highlight the salient differences in the code, we note the following:

- Define $\mathbf{z} := \mathbf{X}\boldsymbol{\beta} - \sigma\mathbf{y}$, then the gradient vector of function f can be computed as

$$\nabla f(\mathbf{x}) := \left(\frac{1}{n}\mathbf{z}^T\mathbf{X}, -\frac{1}{\sigma} - \frac{1}{n}\mathbf{y}^T\mathbf{z}\right)^T.$$

This computation requires two matrix vector multiplications and one inner product.

- The quantity λ_k can be explicitly computed as

$$\lambda_k := \left(\left(\frac{1}{\sigma_k^2} + \frac{1}{n} \mathbf{y}^T \mathbf{y} \right) (\mathbf{d}_\sigma^k)^2 + \frac{1}{n} \mathbf{z}_k^T \mathbf{z}_k - \frac{2}{n} \mathbf{d}_\sigma^k \mathbf{y}^T \mathbf{z}_k \right)^{1/2},$$

where $\mathbf{z}_k := \mathbf{X}\mathbf{d}_{\boldsymbol{\beta}}^k$ and $\mathbf{d}_g^k := ((\mathbf{d}_{\boldsymbol{\beta}}^k)^T, \mathbf{d}_\sigma^k)^T$ is the search direction. This quantity requires one matrix-vector multiplication and two inner products. Moreover, this matrix-vector product can be reused to compute the gradient for the next iteration.

The final algorithm is very similar to Algorithm 5 and hence we omit the details.

5. Numerical experiments

In this section, we illustrate our optimization framework via numerical experiments on the variants discussed in Section 4. We only focus on proximal gradient and Newton variants and encourage the interested reader to try out the quasi-Newton variants for their own applications. All the tests are performed in MATLAB 2011b running on a PC Intel Xeon X5690 at 3.47GHz per core with 94Gb RAM.[6]

5.1 Proximal-Newton method in action

By using the graph selection problem, we first show that the modifications on the proximal-Newton method provides advantages in practical convergence as compared to state-of-the-art strategies and provides a safeguard for line-search procedures in optimization routines. We then highlight the impact of different subsolvers for (35) in the practical convergence of the algorithms.

5.1.1 Comparison of different step-size selection procedures

We apply four different step-size selection procedures in our proximal-Newton framework to solve problem (2). Specifically, we test the algorithm based on the following configuration:

(i) We implement Algorithm 3 in MATLAB using FISTA (Beck and Teboulle, 2009a) to solve the dual subproblem with the following stopping criterion: $\|\boldsymbol{\Theta}_{i+1} - \boldsymbol{\Theta}_i\|_F \leq 10^{-8} \times \max\{\|\boldsymbol{\Theta}_{i+1}\|_F, 1\}$.

(ii) We consider four different globalization procedures, whose details can be found in Section 3.1: *a*) `NoLS` which uses the analytic step size $\alpha_k^* = (1+\lambda_k)^{-1}$, *b*) `BtkLS` which is an instance of the proximal-Newton framework of (Lee et al., 2012) and uses the standard backtracking line-search based on the Amirjo's rule, *c*) `E-BtkLS` which is based on the standard backtracking line-search enhanced by the lower bound α_k^* and, *d*) `FwLS` as the forward line-search by starting from α_k^* and increasing the step size until either infeasibility or the objective value does not improve.

6. We also provide MATLAB implementations of the examples in this section as a software package (SCOPT) at `http://lions.epfl.ch/software`.

(*iii*) We test our implementation on four problem cases: The first problem is a synthetic examples of size $p = 10$, where the data is generated as in (Kyrillidis and Cevher, 2013). We run this test for 10 times and report computational primitives in average. Three remaining problems are based on real data from `http://ima.umn.edu/~maxxa007/send_SICS/`, where the regularization parameters are chosen as the standard values (cf., Tran-Dinh et al. (2013a); Lee et al. (2012); Hsieh et al. (2011)).

The numerical results are summarized in Table 2. Here, #iter denotes the (average) number of iterations, #chol represents the (average) number of Cholesky decompositions and #Mm is the (average) number of matrix-matrix multiplications.

Table 2: Metadata for the line search strategy comparison

	Synthetic ($\rho = 0.01$)			Arabidopsis ($\rho = 0.5$)			Leukemia ($\rho = 0.1$)			Hereditary ($\rho = 0.1$)		
LS Scheme	#iter	#chol	#Mm	#iter	#chol	#Mm	#iter	#chol	#Mm	#iter	#chol	#Mm
`NoLS`	25.4	-	3400	18	-	1810	44	-	9842	72	-	20960
`BtkLS`	25.5	37.0	2436	11	25	718	15	50	1282	19	63	2006
`E-BtkLS`	25.5	36.2	2436	11	24	718	15	49	1282	15	51	1282
`FwLS`	18.1	26.2	1632	10	17	612	12	34	844	14	44	1126

We can see that our new step-size selection procedure `FwLS` shows superior empirical performance as compared to the rest: The standard approach `NoLS` usually starts with pessimistic step-sizes which are designed for worst-case problem structures. Therefore, we find it advantageous to continue with a forward line-search procedure. Whenever it reaches the quadratic convergence, no Cholesky decompositions are required. This makes a difference, compared to standard backtracking line-search `BtkLS` where we need to evaluate the objective value at every iteration. While there is no free lunch, the cost of computing λ_k is $\mathcal{O}(p^2)$ in `FwLS`, which turns out to be quite cheap in this application. The `E-BtkLS` combines both backtrack line-search and our analytic step-size $\alpha_k^* := (1 + \lambda_k)^{-1}$, which outperforms `BtkLS` as the regularization parameter becomes smaller. Finally, we note that the `NoLS` variant needs more iterations but it does not require any Cholesky decompositions, which might be advantageous in homogeneous computational platforms.

5.1.2 Impact of different solvers for the subproblems

As mentioned in the introduction, an important step in our second order algorithmic framework is the solution of the subproblem (15). If the variable matrix $\mathbf{H}_k$ is not diagonal, computing $\mathbf{s}_{\mathbf{H}_k}^k$ corresponds to solving a convex subproblem. For a given regularization term g, we can exploit different existing approaches to tackle this problem. We illustrate that the overall framework to be quite robust against the solution accuracy of the individual subsolver.

In this test, we consider the broad used ℓ_1-norm function as the regularizer. Hence, (15) collapses to an unconstrained LASSO problem; cf. (Wright et al., 2009). To this end, we implement the proximal-Newton algorithm to solve the graph learning problem (2) where $g(\mathbf{x}) := \rho \|\mathbf{x}\|_1$. To show the impact of the subsolver in (2), we implement the following methods, which are all available in our software package SCOPT:

Table 3: METADATA FOR THE SUBSOLVER EFFICIENCY COMPARISON

SUB-SOLVERS	Estrogen ($p = 692$)			Arabidopsis ($p = 834$)			Leukemia ($p = 1255$)			Hereditary($p = 1869$)		
	#iter	#chol	time[s]	#iter	#chol	time[s]	#iter	#chol	time[s]	#iter	#chol	time[s]
	$\rho = 0.5$											
	#nnz $= 0.022p^2$			#nnz $= 0.030p^2$			#nnz $= 0.022p^2$			#nnz $= 0.020p^2$		
`pFISTA`	9	29	13.10	10	35	24.76	9	31	286.57	17	80	1608.66
`pFISTA[gpu]`	9	29	10.70	10	35	16.81	9	31	231.97	17	80	1265.97
`dFISTA`	8	16	4.66	10	17	10.92	14	22	50.19	14	27	147.86
`dFISTA[gpu]`	8	16	4.16	10	17	7.89	14	22	43.53	14	27	120.16
`FastAS`	7	24	28.69	8	27	96.93	9	31	532.11	11	40	1682.28
`BCDC`	8	25	90.35	9	28	227.27	9	31	549.80	12	47	3452.82
`MatQUIC`	11	29	21.61	10	35	50.67	10	35	119.06	14	44	891.29
`ProxGrad1`	175	175	8.82	226	226	17.78	230	230	44.06	660	660	350.52
	$\rho = 0.1$											
	#nnz $= 0.072p^2$ ($\sim 6\%$)			#nnz $= 0.074p^2$			#nnz $= 0.065p^2$			#nnz $= 0.063p^2$		
`pFISTA`	34	101	357.25	57	148	1056.90	143	242	7490.27	-	-	-
`pFISTA[gpu]`	34	101	300.90	57	148	730.07	143	242	6083.06	-	-	-
`dFISTA`	14	32	12.51	12	35	15.53	12	34	38.73	14	44	150.03
`dFISTA[gpu]`	14	32	11.18	12	35	11.18	12	34	33.45	14	44	121.37
`FastAS`	-	-	-	-	-	-	-	-	-	-	-	-
`BCDC`	13	48	1839.17	15	50	4806.62	-	-	-	-	-	-
`MatQUIC`	30	88	573.87	36	95	1255.13	36	95	4260.97	-	-	-
`ProxGrad1`	4345	4345	224.95	6640	6640	532.77	9225	9225	1797.49	-	-	-

(i) `pFISTA` and `dFISTA`: in these cases, we use the FISTA algorithm (Beck and Teboulle, 2009a) for solving the primal (35) and the dual subproblem (37). Morever, to speedup the computations, we further run these methods on the GPU [NVIDIA Quadro 4000].

(ii) `FastAS`: this method corresponds to the exact implementation of the fast active-set method proposed in (Kim and Park, 2010) for solving the primal-dual (35).

(iii) `BCDC`: here, we consider the block-coordinate descent method implemented in (Hsieh et al., 2011) for solving the primal subproblem (35).

We also compare the above variants of the *proximal-Newton approach* with (i) the proximal-gradient method (Algorithm 4) denoted by `ProxGrad1` and (ii) a precise MATLAB implementation of QUIC (`MatQUIC`), as described in (Hsieh et al., 2011). For the proximal-Newton and `MatQUIC` approaches, we terminate the execution if the maximum number of iterations exceeds 200 or the total execution time exceeds the 5 hours. The maximum number of iterations in `ProxGrad1` is set to 10^4.

The results are reported in Table 3. Overall, we observe that `dFISTA` shows superior performance across the board in terms of computational time and the total number of Cholesky decompositions required. Here, #nnz represents the number of nonzero entries in the final solution. The notation "$-$" indicates that the algorithms exceed either the maximum number of iterations or the time limit (5 hours).

If the parameter ρ is relatively large (i.e., the solution is expected to be quite sparse), `FastAS`, `BCDC` and `MatQUIC` perform well and converge in a reasonable time. This is expected

since all three approaches vastly rely on the sparsity of the solution: the sparser the solution is, the faster their computations are performed, as restricted on the active set of variables. However, when ρ is small, the performance of these methods significantly degrade due to the increased number of active (non-zero) entries.

Aside from the above, `ProxGrad1` performs well in terms of computational time, as compared to the rest of the methods. Unfortunately, the number of Cholesky decompositions in this method can become as many as the number of iterations, which indicates a computational bottleneck in high-dimensional problem cases. Moreover, when ρ is small, this method also slows down and requires more iterations to converge.

On the other hand, we also note that `pFISTA` is rather sensitive to the accuracy of the subsolver within the quadratic convergence region. In fact, while `pFISTA` reaches medium scale accuracies in a manner similar to `dFISTA`, it spends most of its iterations trying to achieve the higher accuracy values. However, this could also be an artifact of our MATLAB implementation.

5.2 Proximal-gradient algorithm in action

In this subsection, we illustrate the performance of proximal gradient algorithm in practice on various problems with different regularizers.

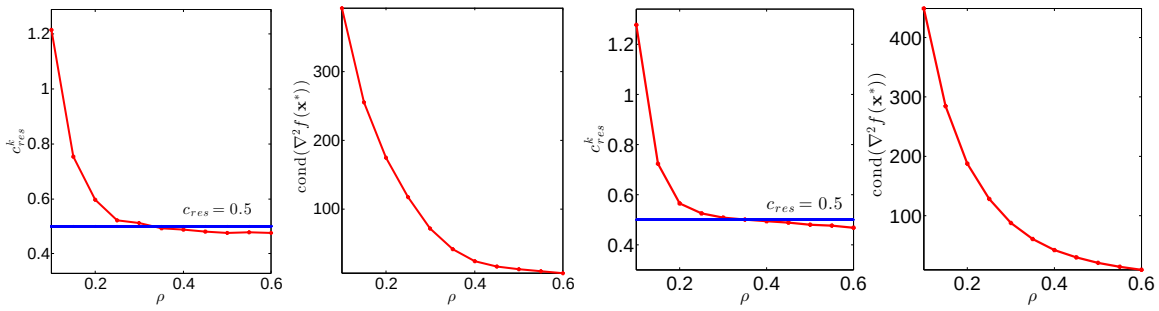
5.2.1 Linear convergence

To show the linear convergence of `ProxGrad1` (Algorithm 2) in practice, we consider the following numerical test. Our experiment is based on the `Lymph` and `Estrogen` problems downloaded from `http://ima.umn.edu/~maxxa007/send_SICS/`. For both problem cases, we use different values for ρ as $\rho = [0.1 : 0.05 : 0.6]$ in MATLAB notation. For each configuration, we measure the quantity

$$c_{\text{res}}^k := \frac{\left\|(\mathbf{D}_k - \nabla^2 f(\mathbf{x}^*))\mathbf{d}_g^k\right\|_{\mathbf{x}^*}^*}{\|\mathbf{d}_g^k\|_{\mathbf{x}^*}}, \tag{49}$$

for few last iterations. This quantity can be referred to as the restricted approximation gap of D_k to $\nabla^2 f(\mathbf{x}^*)$ along the proximal-gradient direction $\mathbf{d}_g^k$. We first run the proximal-Newton method up to 10^{-16} accuracy to obtain the solution $\mathbf{x}^*$ and then run the proximal-gradient algorithm up to 10^{-8} accuracy to compute c_{res}^k and the norm $\left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}$. From the proof of Theorem 10, we can show that if $\mathrm{c}_{\text{res}}^k < 0.5$ for sufficiently large k, then the sequence $\left\{\mathbf{x}^k\right\}$ locally converges to $\mathbf{x}^*$ at a linear rate. We note that this condition is much weaker than the last condition given in Theorem 10 but more difficult to interpret. Note that the requirement in Theorem 10 leads to a restriction on the condition number of $\nabla^2 f(\mathbf{x}^*)$ to be less than 3. We perform this test on two problem instances with 11 different values of the regularization parameter and then compute the median of c_{res}^k for each problem. Figure 4 shows the median of the restricted approximation gap c_{res}^k and the real condition number of $\nabla^2 f(\mathbf{x}^*)$, respectively.

As expected, we observe that the real condition number of $\nabla^2 f(\mathbf{x}^*)$ increases as the regularization parameter decreases. Moreover, the last condition given in Theorem 10 does not hold in this example. However, if we look at the restricted condition number computed



(a) `Lymph` dataset ($p = 578$) (b) `Estrogen` dataset ($p = 692$)

Figure 4: For each test case: **(Left)** Restricted approximation gap c_{res}^k **(Right)** The actual condition number of $\nabla^2 f(\mathbf{x}^*)$.

by (49), we can observe that for $\rho \gtrsim 0.3$, this value is strictly smaller than 0.5. In this case, the local linear convergence is actually observed in practice.

While $c_{\text{res}}^k < 0.5$ is only a sufficient condition and can possibly be improved, we find it to be a good indicator of the convergence behavior. Figure 5 shows the last 100 iterations of our gradient method for the `Lymph` problem with $\rho = 0.15$ and $\rho = 0.55$. The number of iterations needed to achieve the final solution in these cases is 1525 and 140, respectively. In the former case, the calculated restricted condition number is above 0.5 and the final convergence rate suffers. For instance, the contraction factor κ in the estimate $\left\|\mathbf{x}^{k+1} - \mathbf{x}^*\right\|_{\mathbf{x}^*} \leq \kappa \left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}$ is close to 1 when $\rho = 0.15$, while it is smaller when $\rho = 0.55$. We can observe from Figure 5 (left) that the error $\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}$ drops rapidly at the last few iterations due to the affect of the bisection procedure, where we check the condition (30) for $\lambda_k < 1$.

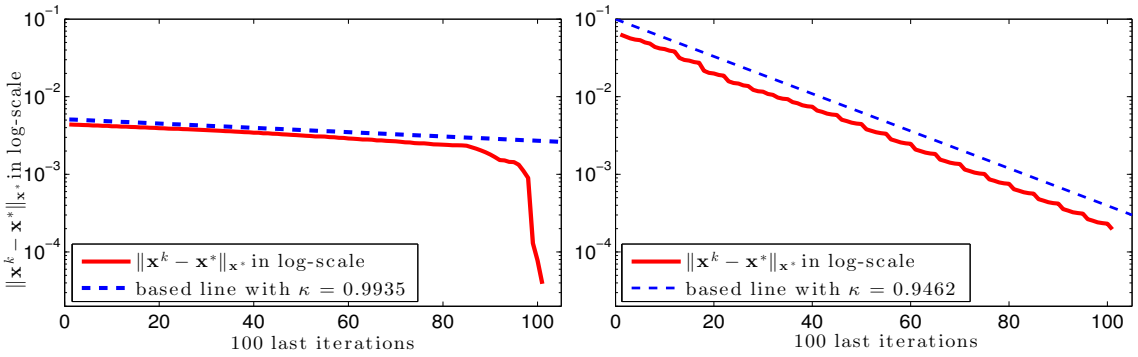


Figure 5: Linear convergence of `ProxGrad1` for `Lymph`: **Left**: $\rho = 0.15$ and **Right**: $\rho = 0.55$.

5.2.2 TV_{ℓ_1}-REGULARIZER

In this experiment, we consider the Poisson intensity reconstruction problem, where the regularizer g, the TV_{ℓ_1}-norm which is called the *anisotropic*-TV; as an example, cf. (Beck and Teboulle, 2009b). Hence, we implement Algorithm 5 (`ProxGrad2`) to solve (3), improve it using the greedy step-size modification as described in Section 3.3 (`ProxGrad2g`), and

compare its performance with the state-of-the-art Sparse Poisson Intensity Reconstruction Algorithms (SPIRAL-TAP) toolbox (Harmany et al., 2012).

As a termination criterion, we have $\|\mathbf{d}_g^k\|_2 \leq 10^{-5}\max\{1, \|\mathbf{x}^k\|_2\}$ or when the objective value does not significantly change after 5 successive iterations, i.e., for each k, $\left|f(\mathbf{x}^{k+j}) - f(\mathbf{x}^k)\right| \leq 10^{-8}\max\{1, \left|f(\mathbf{x}^k)\right|\}$ for $j = 1, \ldots, 5$.

We first illustrate the convergence behavior of the three algorithms under comparison. We consider two image test cases: `house` and `cameraman`, and we set the regularization parameter of the TV_{ℓ_1}-norm to $\rho = 2.5 \times 10^{-5}$. Figure 9 illustrate the convergence of the algorithms both in iteration count and the timing.

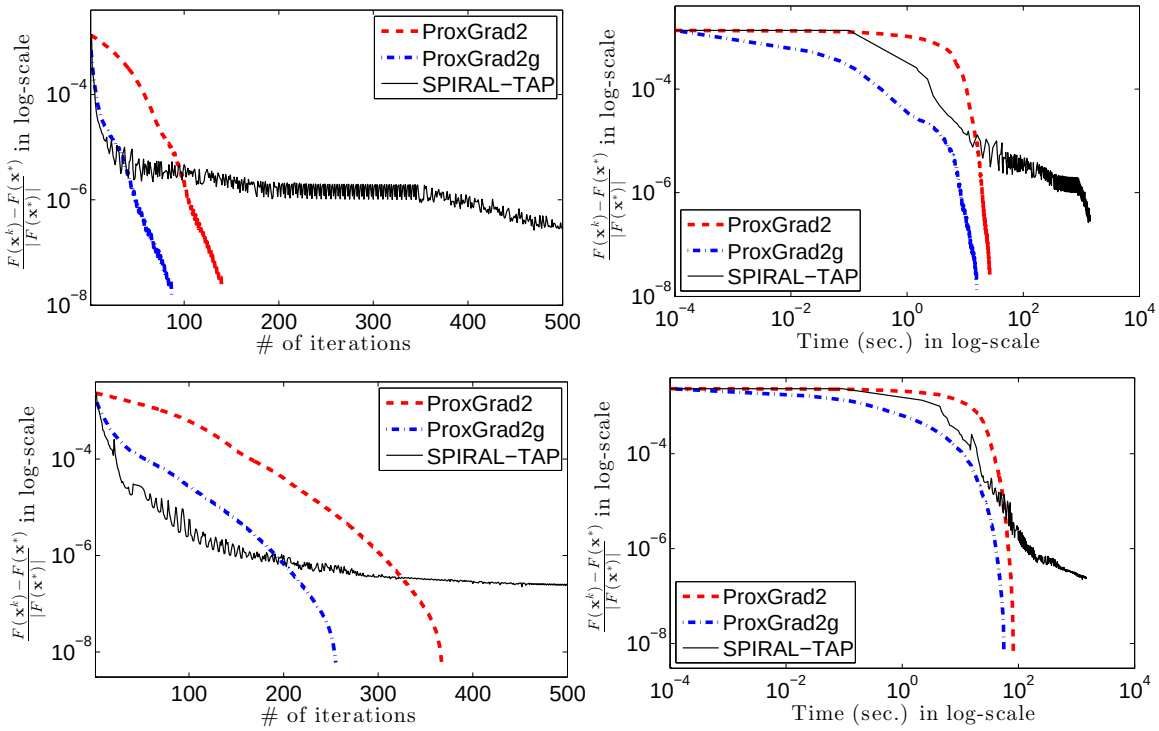


Figure 6: Convergence of three algorithms for `house` (top) and `cameraman` (bottom). **Left**: in iteration scale **Right**: in time log-scale.

Overall, `ProxGrad2g` exhibits the best convergence behavior in terms of iterations and time. Due to the inaccurate solutions of the subproblem (47), the methods might exhibit oscillations. Since SPIRAL-TAP employs a Barzilai-Borwein step-size and performs a line-search procedure up to very small step-size, the objective value is not sufficiently decreased; as a result of this, we observe more oscillations in the objective value.

In stark contrast, `ProxGrad2` and `ProxGrad2g` use the Barzilai-Borwein step-size as an initial-guess for computing a search direction and then use the step-size correction procedure to ensure that the objective function decreases a certain amount at each iteration. This strategy turns out to be more effective since milder oscillations in the objective values are observed in practice (which are due to the inaccuracy of the TV-proximal mapping).

Finally, we test the performance of `ProxGrad2`, `ProxGrad2g` and `SPIRAL-TAP` on 4 different image cases: `barbara, cameraman, house` and `lena`. We set ρ to two different

Figure 7: The reconstructed images for `barbara` ($\rho = 2.5 \times 10^{-5}$)

values: $\rho \in \{10^{-5},\ 2.5 \cdot 10^{-5}\}$. These values are chosen in order to obtain the best visual reconstructions (e.g., see Figure 7) and are previously used in (Harmany et al., 2012). The summary results reported in Table 4. Here, AC denotes the multiplicative factor in time acceleration of `ProxGrad2` as compared to `SPIRAL-TAP`, and ΔF is the difference between the corresponding obtained objective values between `ProxGrad2` and `SPIRAL-TAP` (a positive ΔF means that `SPIRAL-TAP` obtains a higher objective value at termination).

Table 4: The results and performance of three algorithms

	`ProxGrad2g / ProxGrad2 / SPIRAL-TAP`											
Image	$\rho \times 10^{-5}$	#iteration			CPU time [s]			AC		$F^k_{\min}$	ΔF	
`house`	1.0	116	256	500	27.45	56.95	1658.00	60	29	-10718352.93	0.31	0.70
(256 × 256)	2.5	92	244	500	18.18	50.26	1431.94	79	28	-10711758.80	3.20	3.32
`barbara`	1.0	200	324	500	46.92	77.77	1204.36	26	15	-7388497.47	0.05	0.30
(256 × 256)	2.5	164	268	500	36.45	67.98	1620.95	44	24	-7377424.50	1.90	2.02
`cameraman`	1.0	396	516	500	99.56	117.75	389.79	4	3	-9186631.65	0.19	0.07
(256 × 256)	2.5	256	368	500	59.75	85.25	1460.62	24	17	-9175307.33	2.29	2.31
`lena`	1.0	152	220	500	27.43	41.31	1212.69	44	29	-5797053.79	0.10	0.10
(204 × 204)	2.5	304	184	500	59.20	36.77	1132.04	19	31	-5789554.53	1.52	1.25

From Table 4 we observe that both `ProxGrad2` and `ProxGrad2g` are superior to `SPIRAL-TAP`, both in terms of CPU time and the final objective value in majority of problems. As the table shows, `ProxGrad2g` can be 4 to 79 times faster than `SPIRAL-TAP`. Moreover, it reports a better objective values in all cases.

5.2.3 A comparison to standard gradient methods based on $\mathcal{F}_L$ assumption

In this subsection, we use the LASSO problem (48) with unknown variance as a simple test case to illustrate the improvements over the "standard" methods. Note that the standard Lipschitz gradient assumption no longer holds in this example due to the log-term $\log(\sigma)$. For this comparison, we dub our algorithm as `ProxGrad3(g)` and compare it against a state-of-the-art TFOCS software package in Becker et al. (2011). The input data is synthetically generated based on the linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta}^* + \mathbf{s}$, where $\boldsymbol{\beta}$ is the true sparse parameter vector; $\mathbf{X}$ is a Gaussian $n \times p$ matrix and $\mathbf{s} \sim \mathcal{N}(0, \sigma^2)$, where $\sigma = 0.01$. In TFOCS, we

configure the Nesterov's accelerated algorithm with two proximal operations (TFOCS-N07) and adaptive restart as well as the standard gradient method (TFOCS-GRA). Both options use a backtracking step-size selection procedure due to the presence of the logarithmic term in the objective.

As we can see in Figure 9 and Table 5 that `ProxGrad3g` performs the best and manages to converge to a high accuracy solution at a linear rate in both examples. Interestingly, we find the per iteration complexity of `ProxGrad3g` is similar to `ProxGrad3` and TFOCS-GRA. In terms of per iteration cost, TFOCS-N07 is the most expensive one as it uses dual prox operations and adaptive restart, and requires more backtracking operations. Hence, while it takes less iterations as compared to the TFOCS-GRA, it performs worse in terms of timing. For illustration purposes, we ran the algorithms to high accuracy. However, if a typical stopping criteria such as 10^{-6} is used, our algorithm `ProxGrad3g` obtains $\times 3$ to $\times 8$ speed-ups over the standard gradient algorithm with backtracking enhancements.

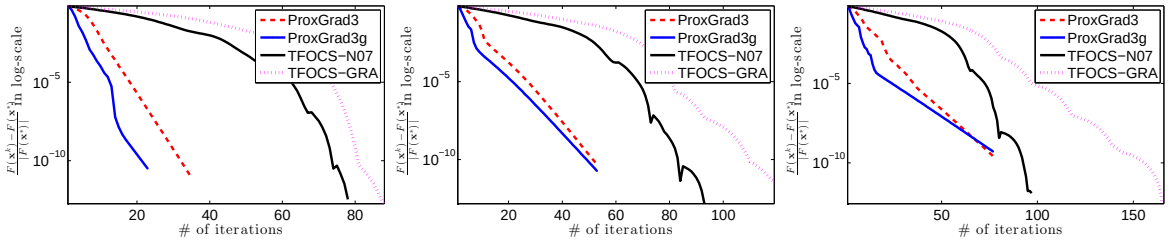


Figure 8: Convergence plots of algorithms under comparison for $n = 3000$ and $p = 10000$. From left to right, $\rho = 10^{-3}, \frac{2}{3} \cdot 10^{-4}, 5 \cdot 10^{-4}$.

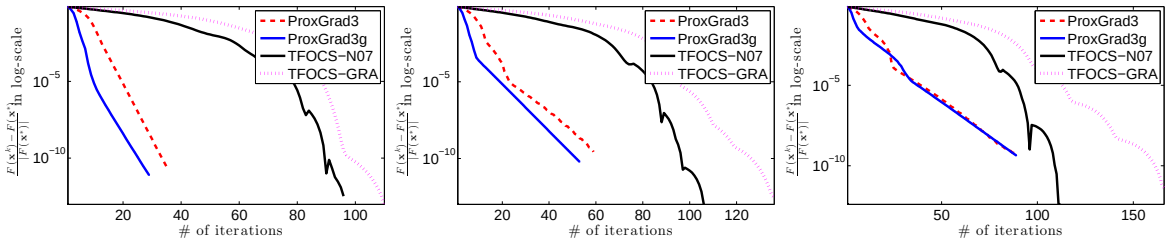


Figure 9: Convergence plots of algorithms under comparison for $n = 15000$ and $p = 50000$. From left to right, $\rho = 2 \cdot 10^{-4}, \frac{4}{3} \cdot 10^{-4}, 10^{-4}$.

6. Conclusions

We propose a variable metric method for minimizing convex functions that are compositions of proximity functions with self-concordant smooth functions. Our framework does not rely on the usual Lipschitz gradient assumption on the smooth part for its convergence theory. A highlight of this work is the new set of analytic step-size selection and correction procedures, which are best matched to the underlying problem structures. Our empirical results illustrate that the new theory leads to significant improvements in the practical performance of the algorithmic instances when tested on a variety of different applications.

Table 5: Metadata on the Lasso problem with unknown variance

Problem	ProxGrad3 / ProxGrad3g / TFOCS-N07 / TFOCS-GRA										
$(3000, 10000)$	#iteration				CPU time [s]				$\|\boldsymbol{\beta}\|_0$	$\|\widehat{\boldsymbol{\beta}}\|_0$	Overlap (%)
$\rho = 10^{-3}$	36	24	79	88	1.0096	0.7862	3.2759	1.7648	360	166	44.72
$\rho = \frac{2}{3} \cdot 10^{-4}$	54	54	94	119	1.2974	1.2918	3.6499	2.4002		378	92.22
$\rho = 5 \cdot 10^{-4}$	78	78	97	166	1.7420	1.7513	3.7794	3.3416		412	100
$(15000, 50000)$	#iteration				CPU time [s]				$\|\boldsymbol{\beta}\|_0$	$\|\widehat{\boldsymbol{\beta}}\|_0$	Overlap (%)
$\rho = 2 \cdot 10^{-4}$	36	30	99	110	21.7937	19.3241	82.3298	46.0475	1800	845	44.98
$\rho = \frac{4}{3} \cdot 10^{-4}$	60	54	108	136	31.7884	29.1194	89.4279	57.9088		1886	87.91
$\rho = 10^{-4}$	90	90	113	166	44.2692	44.0611	95.3060	70.0946		2201	100

Appendix A. Technical proofs

We provide the detailed proofs of the theoretical results in the main text here.

A.1 Proof of Lemma 3

Since g is convex, we have

$$g(\mathbf{y}) \geq g(\mathbf{x}) + \mathbf{v}^T(\mathbf{y} - \mathbf{x}), \quad \forall \mathbf{v} \in \partial g(\mathbf{x}).$$

By adding this inequality to (9) and notting that $F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}),\ \forall \mathbf{x}$, we obtain

$$\begin{aligned} F(\mathbf{y}) &\geq F(\mathbf{x}) + (\nabla f(\mathbf{x}) + \mathbf{v})^T (\mathbf{y} - \mathbf{x}) + \omega(\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}}) \\ &\geq F(\mathbf{x}) - \lambda(\mathbf{x}) \|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} + \omega(\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}}). \end{aligned} \tag{50}$$

Here, the last inequality is due to the generalized Cauchy-Schwartz inequality and $\lambda(\mathbf{x}) := \|\nabla f(\mathbf{x}) + \mathbf{v}\|_{\mathbf{x}}^*$. Let $\mathcal{L}_F(F(\mathbf{x})) := \{\mathbf{y} \in \mathrm{dom}(F) \mid F(\mathbf{y}) \leq F(\mathbf{x})\}$ be a sublevel set of F. Then, for any $\mathbf{y} \in \mathcal{L}_F(F(\mathbf{x}))$, we have $F(\mathbf{y}) \leq F(\mathbf{x})$ which leads to

$$\lambda(\mathbf{x}) \|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} \geq \omega(\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}}),$$

due to (50). Since ω is convex and strictly increasing, the equation $\lambda(\mathbf{x})t - \omega(t) = 0$ has unique solution $t^* > 0$, if $\lambda(\mathbf{x}) < 1$. Therefore, for any $0 \leq t \leq t^*$, we have $\|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} \leq t^*$. This implies that $\mathcal{L}_F(F(\mathbf{x}))$ is bounded. Hence, $\mathbf{x}^*$ exists due to the well-known Weierstrass theorem. The uniqueness of $\mathbf{x}^*$ follows from the monotonicity of $\omega(\cdot)$. □

A.2 Proofs of global convergence: Theorem 4 and Theorem 9

In this subsection, we provide the proofs of Theorem 4, Lemma 8 and Theorem 9 in a unified fashion. We first provide a key result quantifying the improvement of the objective as a function of the step-size α_k.

Maximum decrease of the objective function: Let $\beta_k := \|\mathbf{d}^k\|_{\mathbf{H}^k}$, $\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k}$ and

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k = (1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}^k,$$

where $\alpha_k := \frac{\beta_k^2}{\lambda_k(\lambda_k+\beta_k^2)} \in (0,1]$. We will prove below that the following holds at each iteration of the algorithms

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega\left(\frac{\beta_k^2}{\lambda_k}\right). \tag{51}$$

Moreover, the choice of α_k is *optimal* (in the worse-case sense).

Proof Indeed, since g is convex and $\alpha_k \in (0,1]$, we have $g(\mathbf{x}^{k+1}) = g\left((1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}^k\right) \leq (1-\alpha_k)g(\mathbf{x}^k) + \alpha_k g(\mathbf{s}^k)$, which leads to

$$g(\mathbf{x}^{k+1}) - g(\mathbf{x}^k) \leq \alpha_k (g(\mathbf{s}^k) - g(\mathbf{x}^k)). \tag{52}$$

Combining (52) with the self-concordant property (10) of f, we obtain

$$\begin{aligned} F(\mathbf{x}^{k+1}) &\leq F(\mathbf{x}^k) + \nabla f(\mathbf{x}^k)^T(\mathbf{x}^{k+1}-\mathbf{x}^k) + \omega_*\left(\|\mathbf{x}^{k+1}-\mathbf{x}^k\|_{\mathbf{x}^k}\right) + \alpha_k\left(g(\mathbf{s}^k) - g(\mathbf{x}^k)\right) \\ &\overset{(16)}{\leq} F(\mathbf{x}^k) + \alpha_k \nabla f(\mathbf{x}^k)^T \mathbf{d}^k + \omega_*\left(\alpha_k \|\mathbf{d}^k\|_{\mathbf{x}^k}\right) + \alpha_k\left(g(\mathbf{s}^k) - g(\mathbf{x}^k)\right). \end{aligned} \tag{53}$$

Since $\mathbf{s}^k$ is the unique solution of (15), by using the optimality condition (17), we get

$$\begin{aligned} -\nabla f(\mathbf{x}^k) - \mathbf{H}_k(\mathbf{s}^k - \mathbf{x}^k) \in \partial g(\mathbf{s}^k) \Rightarrow \\ -\nabla f(\mathbf{x}^k)^T(\mathbf{s}^k - \mathbf{x}^k) - \|\mathbf{s}^k - \mathbf{x}^k\|^2_{\mathbf{H}_k} \in (\mathbf{s}^k - \mathbf{x}^k)^T \partial g(\mathbf{s}^k). \end{aligned} \tag{54}$$

Combining (54) with $g(\mathbf{x}^k) - g(\mathbf{s}^k) \geq \mathbf{v}^T(\mathbf{x}^k - \mathbf{s}^k),\ \mathbf{v} \in \partial(\mathbf{s}^k)$, due to the convexity of $g(\cdot)$, we have

$$g(\mathbf{s}^k) - g(\mathbf{x}^k) \leq -\nabla f(\mathbf{x}^k)^T(\mathbf{s}^k - \mathbf{x}^k) - \|\mathbf{s}^k - \mathbf{x}^k\|^2_{\mathbf{H}_k}. \tag{55}$$

Using (55) in (53) together with the definitions of β_k and λ_k, we obtain

$$F(\mathbf{x}^{k+1}) \overset{(16)}{\leq} F(\mathbf{x}^k) - \alpha_k \beta_k^2 + \omega_*\left(\alpha_k \lambda_k\right). \tag{56}$$

Let us consider the function $\varphi(\alpha) := \alpha\beta_k^2 - \omega_*(\alpha\lambda_k)$. By the definition of $\omega_*(\cdot)$, we can easily show that $\varphi(\alpha)$ attains the maximum

$$\alpha_k := \frac{\beta_k^2}{\lambda_k(\lambda_k + \beta_k^2)},$$

provided that $\alpha_k \in (0,1]$. Moreover, $\varphi(\alpha_k) = \omega(\beta_k^2/\lambda_k)$, which proves (51). Since α_k maximizes φ over $[0,1]$, this value is optimal. ■

Proof of Theorem 4: Since $\mathbf{H}_k := \nabla^2 f(\mathbf{x}^k)$, we observe $\beta_k := \|\mathbf{d}^k\|_{\mathbf{H}_k} \equiv \|\mathbf{d}^k\|_{\mathbf{x}_k} =: \lambda_k$, where $\mathbf{d}^k \equiv \mathbf{d}_n^k$. In this case, the step size α_k in (51) becomes $\alpha_k = \frac{\lambda_k}{1+\lambda_k}$ which is in $(0,1)$. Moreover, (51) reduces to

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \omega(\lambda_k),$$

which is indeed (21).

Finally, we assume that, for a given $\sigma \in (0,1)$, we have $\lambda_k \geq \sigma$ for $0 \leq k \leq k_{\max} - 1$. Since ω strictly increases, it follows from (21) by induction that

$$F(\mathbf{x}^*) \leq F(\mathbf{x}^k) \leq F(\mathbf{x}^0) - \sum_{j=0}^{k-1} \omega(\lambda_j) \leq F(\mathbf{x}^0) - k\omega(\sigma).$$

This estimate shows that the number of iterations to reach $\lambda_k < \sigma$ is at least $k_{\max} = \left\lfloor \frac{F(\mathbf{x}^0)-F(\mathbf{x}^*)}{\omega(\sigma)} \right\rfloor + 1$. □

Proof of Lemma 8: Proof of Lemma 8 immediately follows from (51) by taking $\mathbf{H}_k \equiv \mathbf{D}_k$ and $\mathbf{d}^k \equiv \mathbf{d}_g^k$. □

Proof of Theorem 9: We consider the sequence $\left\{F(\mathbf{x}^k)\right\}_{k\geq 0}$. By Lemma 8, this sequence is nonincreasing. Moreover, $F(\mathbf{x}^0) \geq F(\mathbf{x}^k) \geq F(\mathbf{x}^*)$ for all $k \geq 0$. As a result, the sequence $\left\{F(\mathbf{x}^k)\right\}_{k\geq 0}$ converges to a finite value F^*. By Lemma 8, we can derive

$$\sum_{j=0}^{\infty} \omega\left(\frac{\|\mathbf{d}_g^j\|_{\mathbf{D}_j}^2}{\|\mathbf{d}_g^j\|_{\mathbf{x}^j}}\right) \leq F(\mathbf{x}^0) - F^* < +\infty.$$

Since the function $\omega(\tau) = \tau - \ln(1+\tau) \geq \frac{\tau^2}{4}$ for $\tau \in (0,1]$ is increasing, this implies that $\lim_{j\to\infty} \|\mathbf{d}_g^j\|_2^2 / \|\mathbf{d}_g^j\|_{\mathbf{x}^j} = 0$ due to the fact that $\mathbf{D}_k \succeq \underline{L}\mathbb{I} \succ 0$. Since $\mathcal{L}_F(F(\mathbf{x}^0))$ is bounded, by applying Zangwill's convergence theorem in (Zangwill, 1969), we can show that every limit point $\mathbf{x}^*$ of the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ is the stationary point of (11). Since $\mathbf{x}^*$ is unique, the whole sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ converges to $\mathbf{x}^*$. □

A.3 Proofs of local convergence: Theorem 5, Theorem 7 and Theorem 10

We first provide a fixed-point representation of the optimality conditions and prove some key estimates used in the sequel.

Optimality conditions as fixed-point formulations: Let f be a given standard self-concordant function, g be a given proper, lower semicontinuous and convex function, and $\mathbf{H}_k$ be a given symmetric positive definite matrix. Besides the two key inequalities (9) and (10), we also need the following inequality (Nesterov and Nemirovski, 1994; Nesterov, 2004, Theorem 4.1.6) in the proofs below:

$$\left(1 - \|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}\right)^2 \nabla^2 f(\mathbf{x}) \preceq \nabla^2 f(\mathbf{y}) \preceq \left(1 - \|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}}\right)^{-2} \nabla^2 f(\mathbf{x}), \tag{57}$$

for any $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$ such that $\|\mathbf{y}-\mathbf{x}\|_{\mathbf{x}} < 1$.

For a fixed $\bar{\mathbf{x}} \in \mathrm{dom}(F)$, where $F := f + g$, we redefined the following operators:

$$P_{\bar{\mathbf{x}}}^g(\mathbf{z}) := \left(\nabla^2 f(\bar{\mathbf{x}}) + \partial g\right)^{-1}(\mathbf{z}), \quad S_{\bar{\mathbf{x}}}(\mathbf{z}) := \nabla^2 f(\bar{\mathbf{x}})\mathbf{z} - \nabla f(\mathbf{z}), \tag{58}$$

and

$$\mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \mathbf{z}) := \left(\nabla^2 f(\bar{\mathbf{x}}) - \mathbf{H}_k\right)(\mathbf{z} - \mathbf{x}^k). \tag{59}$$

Here, $P^g_{\bar{\mathbf{x}}}$ and $S_{\bar{\mathbf{x}}}$ can be considered as a generalized proximal operator of g and the gradient step of f, respectively. While $\mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \cdot)$ measures the error between $\nabla^2 f(\bar{\mathbf{x}})$ and $\mathbf{H}_k$ along the direction $z - \mathbf{x}^k$.

Next, given $\mathbf{s}^k$ is the unique solution of (15), we characterize the optimality condition of the original problem (1) and the subproblem (15) based on the $P^g_{\bar{\mathbf{x}}}$, $S_{\bar{\mathbf{x}}}$ and $\mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \cdot)$ operators. From (17), we have

$$S_{\bar{\mathbf{x}}}(\mathbf{x}^k) + \mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \mathbf{s}^k) \in \nabla^2 f(\bar{\mathbf{x}})\mathbf{s}^k + \partial g(\mathbf{s}^k).$$

By the definition of $P^g_{\bar{\mathbf{x}}}$ in (58), the above expression leads to

$$\mathbf{s}^k = P^g_{\bar{\mathbf{x}}}\left(S_{\bar{\mathbf{x}}}(\mathbf{x}^k) + \mathbf{e}_{\bar{\mathbf{x}}}(\mathbf{H}_k, \mathbf{s}^k)\right). \tag{60}$$

By replacing $\bar{\mathbf{x}}$ with $\mathbf{x}^*$, i.e., the unique solution of (1), into (60) we obtain

$$\mathbf{s}^k = P^g_{\mathbf{x}^*}\left(S_{\mathbf{x}^*}(\mathbf{x}^k) + \mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k)\right). \tag{61}$$

Moreover, if we replace $\mathbf{H}_k$ by $\nabla^2 f(\mathbf{x}^*)$ in the above fixed-point expression, we finally have

$$\mathbf{x}^* = P^g_{\mathbf{x}^*}\left(S_{\mathbf{x}^*}(\mathbf{x}^*)\right). \tag{62}$$

Formulas (60) to (62) represent the fixed-point formulation of the optimality conditions.

Key estimates: Let $\mathbf{r}_k := \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}$ and λ_k be defined by (18). For any $\alpha_k \in (0, 1]$, we prove the following estimates:

$$\|\mathbf{s}_n^{k+1} - \mathbf{s}_n^k\|_{\mathbf{x}^k} \leq \frac{\alpha_k^2\lambda_k^2}{1 - \alpha_k\lambda_k} + \frac{2\alpha_k\lambda_k - \alpha_k^2\lambda_k^2}{(1 - \alpha_k\lambda_k)^2}\|\mathbf{d}^{k+1}\|_{\mathbf{x}^k}, \tag{63}$$

$$\|\mathbf{s}^k - \mathbf{x}^*\|_{\mathbf{x}^*} \leq \frac{\mathbf{r}_k^2}{1 - \mathbf{r}_k} + \|(\mathbf{H}_k - \nabla^2 f(\mathbf{x}^*))\mathbf{d}^k\|^*_{\mathbf{x}^*}, \tag{64}$$

provided that $\alpha_k\lambda_k < 1$ and $\mathbf{r}_k < 1$.

Proof First, by using the nonexpansiveness of $P^g_{\mathbf{x}^k}$ in Lemma (2), it follows from (60) that

$$\begin{aligned}
\|\mathbf{s}^{k+1} - \mathbf{s}^k\|_{\mathbf{x}^k} &= \left\|P^g_{\mathbf{x}^k}(S_{\mathbf{x}^k}(\mathbf{x}^{k+1}) + \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_{k+1}, \mathbf{s}^{k+1})) - P^g_{\mathbf{x}^k}(S_{\mathbf{x}^k}(\mathbf{x}^k) + \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_k, \mathbf{s}^k))\right\|_{\mathbf{x}^k} \\
&\overset{(8)}{\leq} \left\|S_{\mathbf{x}^k}(\mathbf{x}^{k+1}) + \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_k, \mathbf{s}^k) - S_{\mathbf{x}^*}(\mathbf{x}^*)\right\|^*_{\mathbf{x}^*} \\
&\overset{(i)}{\leq} \left\|\nabla f(\mathbf{x}^{k+1}) - \nabla f(\mathbf{x}^k) - \nabla^2 f(\mathbf{x}^k)(\mathbf{x}^{k+1} - \mathbf{x}^k)\right\|^*_{\mathbf{x}^k} \\
&+ \left\|\mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_{k+1}, \mathbf{s}^{k+1}) - \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_k, \mathbf{s}^k)\right\|^*_{\mathbf{x}^k} \\
&\overset{(ii)}{=} \left\|\int_0^1 \left(\nabla^2 f(\mathbf{x}^k + \tau(\mathbf{x}^{k+1} - \mathbf{x}^k)) - \nabla^2 f(\mathbf{x}^k)\right)(\mathbf{x}^{k+1} - \mathbf{x}^k)d\tau\right\|^*_{\mathbf{x}^k} \\
&+ \left\|\mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_{k+1}, \mathbf{s}^{k+1}) - \mathbf{e}_{\mathbf{x}^k}(\mathbf{H}_k, \mathbf{s}^k)\right\|^*_{\mathbf{x}^k},
\end{aligned} \tag{65}$$

where (i) and (ii) are due to the triangle inequality and the mean-value theorem, respectively.

Second, we estimate the first term in (65). For this purpose, we define

$$\begin{aligned}\boldsymbol{\Sigma}_k &:= \int_0^1 \left(\nabla^2 f(\mathbf{x}^k + \tau(\mathbf{x}^{k+1} - \mathbf{x}^k)) - \nabla^2 f(\mathbf{x}^k)\right) d\tau,\\ \mathbf{M}_k &:= \nabla^2 f(\mathbf{x}^k)^{-1/2} \boldsymbol{\Sigma}_k \nabla^2 f(\mathbf{x}^k)^{-1/2}.\end{aligned} \tag{66}$$

Based on the proof of (Nesterov, 2004, Theorem 4.1.14), we can show that

$$\|\mathbf{M}_k\|_2 \leq \frac{\left\|\mathbf{x}^{k+1} - \mathbf{x}^k\right\|_{\mathbf{x}^k}}{1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}}.$$

Using this estimate, the definition (66) and noting that $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}^k$, we obtain

$$\begin{aligned}\|\boldsymbol{\Sigma}_k(\mathbf{x}^{k+1} - \mathbf{x}^k)\|^*_{\mathbf{x}^k} &= \|\mathbf{M}_k(\mathbf{x}^{k+1} - \mathbf{x}^k)\|_{\mathbf{x}^k}\\ &\overset{(i)}{\leq} \|\mathbf{M}_k\|_2 \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}\\ &\leq \frac{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2_{\mathbf{x}^k}}{1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}}\\ &= \frac{\alpha_k^2 \|\mathbf{d}^k\|^2_{\mathbf{x}^k}}{1 - \alpha_k \|\mathbf{d}^k\|_{\mathbf{x}^k}},\end{aligned} \tag{67}$$

where (i) is due to the Cauchy-Schwartz inequality.

Third, we consider the second term in (65) for $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$. By the definition of $\mathbf{e}_{\bar{\mathbf{x}}}$, it is obvious that $\mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^k), \mathbf{s}^k) = 0$. Hence, we have

$$\begin{aligned}\mathcal{T}_2 &:= \left\|\mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^{k+1}), \mathbf{s}^{k+1}) - \mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^k), \mathbf{s}^k)\right\|^*_{\mathbf{x}^k}\\ &= \left\|\mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^{k+1}), \mathbf{s}^{k+1})\right\|^*_{\mathbf{x}^k}\\ &= \left\|\left(\nabla^2 f(\mathbf{x}^{k+1}) - \nabla^2 f(\mathbf{x}^k)\right)\mathbf{d}^{k+1}\right\|^*_{\mathbf{x}^k}.\end{aligned} \tag{68}$$

We now define the following quantity, whose spectral norm we bound below

$$\mathbf{N}_k := \nabla^2 f(\mathbf{x}^k)^{-1/2} \left(\nabla^2 f(\mathbf{x}^{k+1}) - \nabla^2 f(\mathbf{x}^k)\right) \nabla^2 f(\mathbf{x}^k)^{-1/2}. \tag{69}$$

By applying (57) with $\mathbf{x} = \mathbf{x}^k$ and $\mathbf{y} = \mathbf{x}^{k+1}$, we can bound the spectral norm of $\mathbf{N}_k$ as follows

$$\begin{aligned}\|\mathbf{N}_k\|_2 &\leq \max\left\{1 - \left(1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}\right)^2, \left(1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k}\right)^{-2} - 1\right\}\\ &= \frac{2\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k} - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2_{\mathbf{x}^k}}{(1 - \|\mathbf{x}^{k+1} - \mathbf{x}^k\|_{\mathbf{x}^k})^2}.\end{aligned} \tag{70}$$

Therefore, from (68) we can obtain the following estimate

$$\begin{aligned}(\mathcal{T}_2)^2 &= \mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^{k+1}), \mathbf{s}^{k+1})^T \nabla^2 f(\mathbf{x}^k)^{-1} \mathbf{e}_{\mathbf{x}^k}(\nabla^2 f(\mathbf{x}^{k+1}), \mathbf{s}^{k+1}) \\ &= (\mathbf{d}^{k+1})^T \ \nabla^2 f(\mathbf{x}^k)^{1/2} \ \mathbf{N}_k^2 \ \nabla^2 f(\mathbf{x}^k)^{1/2} \ \mathbf{d}^{k+1} \\ &\leq \|\mathbf{N}_k\|_2^2 \, \|\mathbf{d}^{k+1}\|_{\mathbf{x}^k}^2. \end{aligned} \tag{71}$$

By substituting (70) into (71) and noting that $\alpha_k \mathbf{d}^k = \mathbf{x}^{k+1} - \mathbf{x}^k$, we obtain

$$\mathcal{T}_2 \leq \frac{2\alpha_k \left\|\mathbf{d}^k\right\|_{\mathbf{x}^k} - \alpha_k^2 \left\|\mathbf{d}^k\right\|_{\mathbf{x}^k}^2}{(1 - \alpha_k \left\|\mathbf{d}^k\right\|_{\mathbf{x}^k})^2} \|\mathbf{d}^{k+1}\|_{\mathbf{x}^k}. \tag{72}$$

Now, by substituting (67) and (72) into (65) and noting that $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$, $\mathbf{s}^k \equiv \mathbf{s}_n^k$, $\mathbf{d}^k \equiv \mathbf{d}_n^k$ and $\lambda_k \equiv \left\|\mathbf{d}_n^k\right\|_{\mathbf{x}^k}$, we obtain

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

which is indeed (63).

Similarly to Proof of (65) and (67), we have

$$\begin{aligned}\|\mathbf{s}^k - \mathbf{x}^*\|_{\mathbf{x}^*} &\overset{(62)}{=} \left\| P_{\mathbf{x}^*}^g(S_{\mathbf{x}^*}(\mathbf{x}^k) + \mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k)) - P_{\mathbf{x}^*}^g(S_{\mathbf{x}^*}(\mathbf{x}^*)) \right\|_{\mathbf{x}^*} \\ &\overset{(8)}{\leq} \left\| S_{\mathbf{x}^*}(\mathbf{x}^k) + \mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k) - S_{\mathbf{x}^*}(\mathbf{x}^*) \right\|_{\mathbf{x}^*}^* \\ &\leq \left\| \nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^*) - \nabla^2 f(\mathbf{x}^*)(\mathbf{x}^k - \mathbf{x}^*) \right\|_{\mathbf{x}^*}^* + \left\| \mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k) \right\|_{\mathbf{x}^*}^* \\ &= \left\| \int_0^1 \left(\nabla^2 f(\mathbf{x}^* + \tau(\mathbf{x}^k - \mathbf{x}^*)) - \nabla^2 f(\mathbf{x}^*) \right) (\mathbf{x}^k - \mathbf{x}^*) d\tau \right\|_{\mathbf{x}^*}^* + \left\| \mathbf{e}_{\mathbf{x}^*}(\mathbf{H}_k, \mathbf{s}^k) \right\|_{\mathbf{x}^*}^* \\ &\overset{(67)}{\leq} \frac{\left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}^2}{1 - \left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}} + \left\| \left(\mathbf{H}_k - \nabla^2 f(\mathbf{x}^*)\right) \mathbf{d}^k \right\|_{\mathbf{x}^*}^*, \end{aligned} \tag{73}$$

which is indeed (64) since $\mathbf{r}_k = \left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}$. ■

Proof of Theorem 5: Since $\mathbf{x}^k = \mathbf{s}_n^k - \mathbf{d}_n^k$ due to (20), we have $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}_n^k = \mathbf{s}_n^k - (1 - \alpha_k)\mathbf{d}_n^k$, which leads to

$$\mathbf{d}_n^{k+1} = \mathbf{s}_n^{k+1} - \mathbf{x}^{k+1} = \mathbf{s}_n^{k+1} - \mathbf{s}_n^k + (1 - \alpha_k)\mathbf{d}_n^k.$$

By applying the triangle inequality to the above expression, we have

$$\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k} = \|\mathbf{s}_n^{k+1} - \mathbf{s}_n^k + (1 - \alpha_k)\mathbf{d}_n^k\|_{\mathbf{x}^k} \leq \|\mathbf{s}_n^{k+1} - \mathbf{s}_n^k\|_{\mathbf{x}^k} + (1 - \alpha_k)\|\mathbf{d}_n^k\|_{\mathbf{x}^k}. \tag{74}$$

Substituting (63) into (74) we obtain

$$\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k} \leq \frac{\alpha_k^2 \lambda_k^2}{1 - \alpha_k \lambda_k} + \frac{2\alpha_k \lambda_k - \alpha_k^2 \lambda_k^2}{(1 - \alpha_k \lambda_k)^2} \|\mathbf{d}^{k+1}\|_{\mathbf{x}^k} + (1 - \alpha_k)\lambda_k.$$

Rearranging this inequality we get

$$\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k} \leq \left(\frac{(1-\alpha_k\lambda_k)\left(1-\alpha_k+(2\alpha_k^2-\alpha_k)\lambda_k\right)}{1-4\alpha_k\lambda_k+2\alpha_k^2\lambda_k^2} \right) \lambda_k, \tag{75}$$

provided that $1-4\alpha_k\lambda_k+2\alpha_k^2\lambda_k^2 > 0$. Now, by applying (57) with $\mathbf{x}=\mathbf{x}^k$ and $\mathbf{y}=\mathbf{x}^{k+1}$, one can show that

$$\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^{k+1}} \leq \frac{\|\mathbf{d}_n^{k+1}\|_{\mathbf{x}^k}}{1-\alpha_k\|\mathbf{d}_n^k\|_{\mathbf{x}^k}}. \tag{76}$$

We note that $1-4\alpha_k\lambda_k+2\alpha_k^2\lambda_k^2 > 0$ if $\alpha_k\lambda_k < 1-1/\sqrt{2}$. By combining (75) and (76) we obtain

$$\lambda_k \left\|\mathbf{d}_n^{k+1}\right\|_{\mathbf{x}^{k+1}} \leq \left(\frac{1-\alpha_k+(2\alpha_k^2-\alpha_k)\lambda_k}{1-4\alpha_k\lambda_k+2\alpha_k^2\lambda_k^2} \right) \lambda_k,$$

which is (22).

Next, we consider the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ generated by dub-step proximal Newton method (20) with the step size $\alpha_k = (1+\lambda_k)^{-1}$. Then, (22) is transformed into

$$\lambda_{k+1} \leq \frac{2\lambda_k}{1-2\lambda_k-\lambda_k^2}\lambda_k. \tag{77}$$

Assuming $\lambda_k \leq \bar{\sigma} := \sqrt{5}-2$, we can easily deduce that $\frac{2\lambda_k}{1-2\lambda_k-\lambda_k^2} \leq 1$ and thus, $\lambda_{k+1} \leq \lambda_k$. By induction, if $\lambda_0 \leq \bar{\sigma}$ then, $\lambda_{k+1} \leq \lambda_k$ for all $k \geq 0$. Moreover, we have $\lambda_{k+1} \leq \frac{2}{1-2\bar{\sigma}-\bar{\sigma}^2}\lambda_k^2$, which shows that the sequence $\{\lambda_k\}_{k\geq 0}$ converges to zero at a quadratic rate, which completes the proof of part b).

Finally, since $\alpha_k = 1$, the estimate (22) reduces to $\lambda_{k+1} \leq \frac{\lambda_k^2}{1-4\lambda_k+2\lambda_k^2}$. By the same argument as in the proof of part b), we can show that the sequence $\{\lambda_k\}_{k\geq 0}$ converges to zero at a quadratic rate. □

Proof of Theorem 7: We first prove the statement (a). Since $\mathbf{x}^{k+1} \equiv \mathbf{s}_q^k$ due to (25), from (64) we have

$$\mathbf{r}_{k+1} \leq \frac{\mathbf{r}_k^2}{1-\mathbf{r}_k} + \left\| \left(\mathbf{H}_k - \nabla^2 f(\mathbf{x}^*)\right)(\mathbf{x}^{k+1}-\mathbf{x}^k) \right\|_{\mathbf{x}^*}^*. \tag{78}$$

Now, by using the condition (26), we can easily show that the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ converges super-linearly to $\mathbf{x}^*$ provided that $\left\|\mathbf{x}^0-\mathbf{x}^*\right\|_{\mathbf{x}^*} \leq \rho_0 < 1$.

Next, it is well-known (see, e.g., Nocedal and Wright (2006)) that if matrix $\mathbf{H}_k$ is positive definite and $(\mathbf{y}^k)^T(\mathbf{s}^k) > 0$ then the matrix $\mathbf{H}_{k+1}$ updated by (24) is also positive definite. Indeed, we have $(\mathbf{y}^k)^T(\mathbf{s}^k) = \int_0^1 (\mathbf{s}^k)^T\nabla^2 f(\mathbf{x}^k+t\mathbf{s}^k)\mathbf{s}^k dt$. Therefore, under the condition $\left\|\mathbf{s}^k\right\|_{\mathbf{x}^k} < 1$, we can show that $(\mathbf{y}^k)^T(\mathbf{s}^k) \geq (\mathbf{s}^k)^T\nabla^2 f(\mathbf{x}^k)\mathbf{s}^k = \left\|\mathbf{s}^k\right\|_{\mathbf{x}^k}^2 > 0$. By multiplying (24) by $\mathbf{s}^k$ we can easily see that $\mathbf{H}_{k+1}$ satisfies the secant equation (23). The statement (b) is proved.

Finally, we are equipped to prove (c). We estimate $\left\|\mathbf{y}^k - \nabla^2 f(\mathbf{x}^*)\mathbf{s}^k\right\|_{\mathbf{x}^*}^*$ as follows

$$\|\mathbf{y}^k - \nabla^2 f(\mathbf{x}^*)\mathbf{s}^k\|_{\mathbf{x}^*}^* \leq \frac{\mathbf{r}_k + \mathbf{r}_{k+1}}{(1-\mathbf{r}_k)(1-\mathbf{r}_{k+1})}\|\mathbf{s}^k\|_{\mathbf{x}^*}. \tag{79}$$

Now, by assumption that $\sum_{k=0}^{\infty}\mathbf{r}_k < +\infty$, we obtain from (79) that $\sum_{k=0}^{\infty}\varepsilon_k < +\infty$, where $\varepsilon_k := \frac{\mathbf{r}_k+\mathbf{r}_{k+1}}{(1-\mathbf{r}_k)(1-\mathbf{r}_{k+1})}$. By applying (Byrd and Nocedal, 1989, Theorem 3.2.), we can show that the Dennis-Moré condition (26) is satisfied. This implies that the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ generated by scheme (25) converges super-linearly to $\mathbf{x}^*$. □

Proof of Theorem 10: For $\left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*} < 1$, from (64), we have

$$\|\mathbf{s}_g^k - \mathbf{x}^*\|_{\mathbf{x}^*} \leq \frac{\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}^2}{1 - \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}} + \left\|\left(\mathbf{D}_k - \nabla^2 f(\mathbf{x}^*)\right)\mathbf{d}^k\right\|_{\mathbf{x}^*}^*. \tag{80}$$

Now, using the condition $\left\|\left(\mathbf{D}_k - \nabla^2 f(\mathbf{x}^*)\right)\mathbf{d}^k\right\|_{\mathbf{x}^*}^* \leq \frac{1}{2}\|\mathbf{d}_g^k\|_{\mathbf{x}^*}$, (80) implies

$$\begin{aligned}\|\mathbf{s}_g^k - \mathbf{x}^*\|_{\mathbf{x}^*} &\leq \frac{\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}^2}{1 - \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}} + \gamma\|\mathbf{d}_g^k\|_{\mathbf{x}^*} \\ &\leq \frac{\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}^2}{1 - \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}} + \gamma\|\mathbf{s}_g^k - \mathbf{x}^*\|_{\mathbf{x}^*} + \gamma\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*},\end{aligned}$$

where $\gamma \in (0, 1/2)$. Rearranging this inequality, we obtain

$$\|\mathbf{s}_g^k - \mathbf{x}^*\|_{\mathbf{x}^*} \leq \frac{1}{1-\gamma}\left(\gamma + \frac{\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}}{1 - \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}}\right)\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}. \tag{81}$$

Now, since $\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k\mathbf{d}_g^k = (1-\alpha_k)\mathbf{x}^k + \alpha_k\mathbf{s}_g^k$, we can further estimate from (81) as

$$\begin{aligned}\|\mathbf{x}^{k+1} - \mathbf{x}^*\|_{\mathbf{x}^*} &\leq (1-\alpha_k)\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*} + \alpha_k\|\mathbf{s}_g^k - \mathbf{x}^*\|_{\mathbf{x}^*} \\ &\leq \left[1 - \alpha_k + \frac{\alpha_k}{1-\gamma}\left(\gamma + \frac{\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}}{1 - \|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}}\right)\right]\|\mathbf{x}^k - \mathbf{x}^*\|_{\mathbf{x}^*}.\end{aligned} \tag{82}$$

Let us define $\tilde{\psi}_k := (1-\alpha_k) + \frac{\alpha_k}{1-\gamma}\left(\gamma + \frac{\left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}}{1 - \left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}}\right)$. Then, for $\gamma < \frac{1}{2}$, $\tilde{\psi}_k < 1$ if $\left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*} < \frac{1-2\gamma}{2(1-\gamma)}$. Therefore, by induction, if we choose $\left\|\mathbf{x}^0 - \mathbf{x}^*\right\|_{\mathbf{x}^*} < \frac{1-2\gamma}{2(1-\gamma)}$, then $\left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*} < \frac{1-2\gamma}{2(1-\gamma)}$ for all $k \geq 0$. Moreover, $\left\|\mathbf{x}^{k+1} - \mathbf{x}^*\right\|_{\mathbf{x}^*} \leq \tilde{\psi}_k\left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}$ for $k \geq 0$ and $\tilde{\psi}_k \in [0, 1)$. This implies that $\left\{\left\|\mathbf{x}^k - \mathbf{x}^*\right\|_{\mathbf{x}^*}\right\}_{k\geq 0}$ linearly converges to zero with the factor $\tilde{\psi}_k$.

Finally, we assume that $\mathbf{D}_k := L_k\mathbb{I}$, the quantity in (69) satisfies

$$\mathbf{N}_* := \nabla^2 f(\mathbf{x}^*)^{-1/2}\left(\nabla^2 f(\mathbf{x}^*) - \mathbf{H}_k\right)\nabla^2 f(\mathbf{x}^*)^{-1/2} = \mathbb{I} - L_k\nabla^2 f(\mathbf{x}^*)^{-1}.$$

Then, we can easily observe that

$$\|\mathbf{N}_*\|_2 = \left\|\mathbb{I} - L_k\nabla^2 f(\mathbf{x}^*)^{-1}\right\|_2 \leq \max\left\{\left|1 - \frac{L_k}{\sigma_{\min}^*}\right|, \left|1 - \frac{L_k}{\sigma_{\max}^*}\right|\right\} := \gamma_*, \tag{83}$$

where $\sigma^*_{\min}$ (respectively, $\sigma^*_{\max}$) is the smallest (respectively, largest) eigenvalue of $\nabla^2 f(\mathbf{x}^*)$. Using the estimate (83), we can derive

$$\begin{aligned}\left\| \left(\mathbf{D}_k - \nabla^2 f(\mathbf{x}^*)\right) \mathbf{d}_g^k \right\|_{\mathbf{x}^*}^* &\overset{(83)}{\leq} \|\mathbf{N}_*\|_2 \|\mathbf{s}^k - \mathbf{x}^k\|_{\mathbf{x}^*} \\ &\leq \gamma_* \|\mathbf{d}_g^k\|_{\mathbf{x}^*},\end{aligned}$$

which proves the last conclusion of Theorem 10. □

References

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

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

A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Trans Image Process.*, 18(11):2419–2434, 2009b.

S. Becker and M.J. Fadili. A quasi-Newton proximal splitting method. In *Proceedings of Neutral Information Processing Systems Foundation*, 2012.

S. Becker, J. Bobin, and E.J. Candès. NESTA: a fast and accurate first-order method for sparse recovery. *SIAM J. Imaging Science*, 4(1):1–39, 2011.

A. Ben-Tal and A.K. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM, 2001.

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

J.F. Bonnans. Local Analysis of Newton-Type Methods for Variational Inequalities and Nonlinear Programming. *Appl. Math. Optim*, 29:161–186, 1994.

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

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

R. H. Byrd and J. Nocedal. A tool for the analysis of quasi-newton methods with application to unconstrained minimization. *SIAM J. Numer. Anal.*, 26(3):727–739, 1989.

E. Candes and T. Tao. The Dantzig selector: Statistical estimation when p is much larger than n. *Annals of Statistics*, 35(6):2313–2351, 2007.

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

E. Chouzenoux, J.-C. Pesquet, and A. Repetti. Variable metric forward-backward algorithm for minimizing the sum of a differentiable function and a convex function. *Tech. Report.*, xx:1–22, 2013.

P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Model. Simul.*, 4:1168–1200, 2005.

A. S. Dalalyan, M. Hebiri, K. Meziani, and J. Salmon. Learning heteroscedastic models by convex programming under group sparsity. *Proc. of the International conference on Machine Learning*, pages 1–8, 2013.

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

J.E. Dennis and J.J. Moré. A characterization of superlinear convergence and its application to quasi-newton methods. *Math. Comp.*, 28(126):549–560, 1974.

J. Eckstein and D. Bertsekas. On the Douglas - Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.*, 55:293–318, 1992.

F. Facchinei and J.-S. Pang. *Finite-dimensional variational inequalities and complementarity problems*, volume 1-2. Springer-Verlag, 2003.

D. Goldfarb and S. Ma. Fast alternating linearization methods of minimization of the sum of two convex functions. *Math. Program., Ser. A*, pages 1–34, 2012. doi: DOI10.1007/s10107-012-0530-2.

T. Goldstein and S. Osher. The split Bregman method for l_1-pegularized problems. *SIAM J. Imaging Sciences*, 2(2):323–343, 2009.

T. Goldstein, B. O'Donoghue, and S. Setzer. Fast alternating direction optimization methods. Tech. report., Department of Mathematics, University of California, Los Angeles, USA, May 2012.

M. Grant, S. Boyd, and Y. Ye. Disciplined convex programming. In L. Liberti and N. Maculan, editors, *Global Optimization: From Theory to Implementation*, Nonconvex Optimization and its Applications, pages 155–210. Springer, 2006.

Z.T. Harmany, R.F. Marcia, and R. M. Willett. This is spiral-tap: Sparse poisson intensity reconstruction algorithms theory and practice,. *IEEE Transactions on Image Processing*, Submitted:1–13, 2012.

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

J. Kim and H. Park. Fast active-set-type algorithms for ℓ_1-regularized linear regression. In *Proceedings of the 13th International Conference on Artificial Intelligience and Statistics (AISTATS)*, volume 9, pages 397–404, Sardinia, Italy, 2010.

A. Kyrillidis and V. Cevher. Fast proximal algorithms for self-concordant function minimization with application to sparse graph selection. *Proc. of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1–5, 2013.

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

J. Löefberg. Yalmip : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004. URL `http://users.isy.liu.se/johanl/yalmip`.

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

H. Mine and M. Fukushima. A minimization method for the sum of a convex function and a continuously differentiable function. *J. Optim. Theory Appl.*, 33:9–23, 1981.

A.S. Nemirovski and M.J. Todd. Interior-point methods for optimization. *Acta Numerica*, pages 191–234, 2009. doi: 10.1017/S0962492906370018.

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

Y. Nesterov. Excessive gap technique in non-smooth convex minimization. *SIAM J. Optim.*, 16(1):235–249, 2005a.

Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1):127–152, 2005b.

Y. Nesterov. Barrier subgradient method. *Math. Program., Ser. B*, 127:31–56, 2011.

Y. Nesterov. Gradient methods for minimizing composite objective function. *Math. Program.*, 140(1):125–161, 2013.

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

Y. Nesterov and M.J. Todd. Self-scaled barriers and interior-point methods for convex programming. *Math. Oper. Research*, 22(1):1–42, 1997.

L. M. Brice no Arias and P. L. Combettes. A monotone + skew splitting model for composite monotone inclusions in duality. *SIAM J. Optim.*, 21(4):1230–1250, 2011.

J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition, 2006.

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

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

S. M. Robinson. Strongly Regular Generalized Equations. *Mathematics of Operations Research, Vol. 5, No. 1 (Feb., 1980), pp. 43-62*, 5:43–62, 1980.

R. T. Rockafellar. *Convex Analysis*, volume 28 of *Princeton Mathematics Series*. Princeton University Press, 1970.

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

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

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

M. Schmidt, N.L. Roux, and F. Bach. Convergence rates of inex- act proximal-gradient methods for convex optimization. *NIPS, Granada, Spain*, 2011.

N. Städler, P. Bülmann, and S. Van de Geer. l_1-penalization for mixture regression models. *Tech. Report.*, pages 1–35, 2012.

Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. A proximal newton framework for composite minimization: Graph learning without Cholesky decompositions and matrix inversions. *JMLR W&CP*, 28(2):271–279, 2013a.

Q. Tran-Dinh, A. Kyrillidis, and V. Cevher. A path following method for composite self-concordant barrier minimization. Technical report, LIONS, EPFL, 2013b.

Q. Tran-Dinh, C. Savorgnan, and M. Diehl. Combining lagrangian decomposition and excessive gap smoothing technique for solving large-scale separable convex optimization problems. *Compt. Optim. Appl.*, 55(1):75–211, 2013c. doi: 10.1007/s10589-012-9515-6.

E. van den Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM J. Sci. Comput.*, 31(2):890–912, 2008.

S. J. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Processing*, 57:2479–2493, 2009.

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

W.I. Zangwill. *Nonlinear Programming*. Prentice Hall, 1969.