Composite Convex Minimization Involving Self-concordant-Like Cost Functions

Quoc Tran-Dinh, Yen-Huan Li, and Volkan Cevher

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

Abstract. The self-concordant-like property of a smooth convex function is a new analytical structure that generalizes the self-concordant notion. While a wide variety of important applications feature the self-concordant-like property, this concept has heretofore remained unexploited in convex optimization. To this end, we develop a variable metric framework of minimizing the sum of a "simple" convex function and a self-concordant-like function. We introduce a new analytic step-size selection procedure and prove that the basic gradient algorithm has improved convergence guarantees as compared to "fast" algorithms that rely on the Lipschitz gradient property. Our numerical tests with real-data sets show that the practice indeed follows the theory.

1 Introduction

In this paper, we consider the following composite convex minimization problem:

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

where f is a nonlinear smooth convex function, while g is a "simple" possibly nonsmooth convex function. Such composite convex problems naturally arise in many applications of machine learning, data sciences, and imaging science. Very often, f measures a data fidelity or a loss function, and g encodes a form of low-dimensionality, such as sparsity or low-rankness.

To trade-off accuracy and computation optimally in large-scale instances of (1), existing optimization methods invariably invoke the additional assumption that the smooth function f also has an L-Lipschitz continuous gradient (cf., [11] for the definition). A highlight is the recent developments on proximal gradient methods, which feature (nearly) dimension-independent, global sublinear convergence rates [3,9,11]. When the smooth f in (1) also has strong regularity [15], the problem (1) is also within the theoretical and practical grasp of proximal-(quasi) Newton algorithms with linear, superlinear, and quadratic convergence rates [5,8,17]. These algorithms specifically exploit second order information or its principled approximations (e.g., via BFGS or L-BFGS updates [13]).

In this paper, we do away with the Lipschitz gradient assumption and instead focus on another structural assumption on f in developing an algorithmic framework for (1), which is defined below.

H.A. Le Thi et al. (eds.), *Model. Comput. & Optim. in Inf. Syst. & Manage. Sci.*,
Advances in Intelligent Systems and Computing 359, DOI: 10.1007/978-3-319-18161-5_14

Definition 1. *A convex function $f \in \mathcal{C}^3(\mathbb{R}^n)$ is called a self-concordant-like function $f \in \mathcal{F}_{\mathrm{scl}}$, if:*

$$|\varphi'''(t)| \leq M_f \varphi''(t) \left\|\mathbf{u}\right\|_2, \tag{2}$$

for $t \in \mathbb{R}$ and $M_f > 0$, where $\varphi(t) := f(\mathbf{x}+t\mathbf{u})$ for any $\mathbf{x} \in \mathrm{dom}(f)$ and $\mathbf{u} \in \mathbb{R}^n$.

Definition 1 mimics the standard self-concordance concept ([10, Definition 4.1.1]) and was first discussed in [1] for model consistency in logistic regression. For composite convex minimization, self-concordant-like functions abound in machine learning, including but not limited to logistic regression, multinomial logistic regression, conditional random fields, and robust regression (cf., the references in [2]). In addition, special instances of geometric programming [6] can also be recast as (1) where $f \in \mathcal{F}_{\mathrm{scl}}$.

The importance of the assumption $f \in \mathcal{F}_{\mathrm{scl}}$ in (1) is twofold. First, it enables us to derive an explicit step-size selection strategy for proximal variable metric methods, enhancing backtracking-line search operations with improved theoretical convergence guarantees. For instance, we can prove that our proximal gradient method can automatically adapt to the local strong convexity of f near the optimal solution to feature linear convergence under mild conditions. This theoretical result is backed up by great empirical performance on real-life problems where the fast Lipschitz-based methods actually exhibit sublinear convergence (cf. Section 4). Second, the self-concordant-like assumption on f also helps us provide scalable numerical solutions of (1) for specific problems where f does not have Lipschitz continuous gradient, such as special forms of geometric programming problems.

Contributions. 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-like 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, and uses an *analytical* step-size to achieve descent in three classes of algorithms: first order methods, second order methods, and quasi-Newton methods.
2. We establish both the global and the local convergence of different variable metric strategies. We pay particular attention to diagonal variable metrics since in this case many of the proximal subproblems can be solved exactly. We derive conditions on when and where these variants achieve locally linear convergence. When the variable metric is the Hessian of f at each iteration, we show that the resulting algorithm locally exhibits quadratic convergence without requiring any globalization strategy such as a backtracking line-search.
3. We apply our algorithms to large-scale real-world and synthetic problems to highlight the strengths and the weaknesses of our variable-metric scheme.

Relation to Prior Work. Many of the composite problems with self-concordant-like f, such as regularized logistics and multinomial logistics, also have Lipschitz continuous gradient. In those specific instances, many theoretically efficient algorithms are applicable [3,5,8,9,11,17]. Compared to these works, our framework has theoretically stronger local convergence guarantees thanks to the specific step-size strategy matched with $f \in \mathcal{F}_{\rm scl}$. The authors of [18] consider composite problems where f is standard self-concordant and proposes a proximal Newton algorithm optimally exploiting this structure. Our structural assumptions and algorithmic emphasis here are different.

Paper Organization. We first introduce the basic definitions and optimality conditions before deriving the variable metric strategy in Section 2. Section 3 proposes our new variable metric framework, describes its step-size selection procedure, and establishes the convergence theory of its variants. Section 4 illustrates our framework in real and synthetic data.

2 Preliminaries

We adopt the notion of self-concordant functions in [10,12] to a different smooth function class. Then we present the optimality condition of problem (1).

2.1 Basic Definitions

Let $g : \mathbb{R}^n \to \mathbb{R}$ be a proper, lower semicontinuous convex function [16] and $\mathrm{dom}(g)$ denote the domain of g. We use $\partial g(\mathbf{x})$ to denote the subdifferential of g at $\mathbf{x} \in \mathrm{dom}\,(g)$ if g is nondifferentiable at $\mathbf{x}$ and $\nabla g(\mathbf{x})$ to denote its gradient, otherwise. Let $f : \mathbb{R}^n \to \mathbb{R}$ be a $\mathcal{C}^3(\mathrm{dom}(f))$ function (i.e., f is three times continuously differentiable). We denote by $\nabla f(\mathbf{x})$ and $\nabla^2 f(\mathbf{x})$ the gradient and the Hessian of f at $\mathbf{x}$, respectively. Suppose that, for a given $\mathbf{x} \in \mathrm{dom}\,(f)$, $\nabla^2 f(\mathbf{x})$ is positive definite (i.e., $\nabla^2 f(\mathbf{x}) \in \mathcal{S}^n_{++}$), we define the local norm of a given vector $\mathbf{u} \in \mathbb{R}^n$ as $\|\mathbf{u}\|_{\mathbf{x}} := [\mathbf{u}^T\nabla^2 f(\mathbf{x})\mathbf{u}]^{1/2}$. The corresponding dual norm of $\mathbf{u}$, $\|\mathbf{u}\|^*_{\mathbf{x}}$ is defined as $\|\mathbf{u}\|^*_{\mathbf{x}} := \max\left\{\mathbf{u}^T\mathbf{v} \mid \|\mathbf{v}\|_{\mathbf{x}} \le 1\right\} = [\mathbf{u}^T\nabla^2 f(\mathbf{x})^{-1}\mathbf{u}]^{1/2}$.

2.2 Composite Self-Concordant-Like Minimization

Let $f \in \mathcal{F}_{\rm scl}(\mathbb{R}^n)$ and g be proper, closed and convex. The optimality condition for (1) can be concisely written as follows:

$$0 \in \nabla f(\mathbf{x}^\star) + \partial g(\mathbf{x}^\star). \tag{3}$$

Let us denote by $\mathbf{x}^\star$ as an optimal solution of (1). Then, the condition (3) is necessary and sufficient. We also say that $\mathbf{x}^\star$ is *nonsingular* if $\nabla^2 f(\mathbf{x}^\star)$ is positive definite. We now establish the existence and uniqueness of the solution $\mathbf{x}^\star$ of (1), whose proof can be found in [19].

Lemma 1. *Suppose that $f \in \mathcal{F}_{\rm scl}(\mathbb{R}^n)$ satisfies Definition 1 for some $M_f > 0$. Suppose further that $\nabla^2 f(\mathbf{x}) \succ 0$ for some $\mathbf{x} \in \mathrm{dom}(f)$. Then the solution $\mathbf{x}^\star$ of* (1) *exists and is unique.*

For a given symmetric positive definite matrix $\mathbf{H}$, we define a generalized proximal operator $\mathrm{prox}_{\mathbf{H}^{-1}g}$ as:

$$\mathrm{prox}_{\mathbf{H}^{-1}g}(\mathbf{x}) := \arg\min_{\mathbf{z}} \left\{ g(\mathbf{z}) + (1/2)\, \|\mathbf{z} - \mathbf{x}\|_{\mathbf{H}^{-1}}^2 \right\}. \tag{4}$$

Due to the convexity of g, this operator is well-defined and single-valued. If we can compute $\mathrm{prox}_{\mathbf{H}^{-1}g}$ efficiently (e.g., by a closed form or by polynomial time algorithms), then we say that g is *proximally tractable*. Examples of proximal tractability convex functions can be found, e.g., in [14]. Using $\mathrm{prox}_{\mathbf{H}^{-1}g}$, we can write condition (1) as:

$$\mathbf{x}^\star - \mathbf{H}^{-1}\nabla f(\mathbf{x}^\star) \in (\mathbb{I} + \mathbf{H}^{-1}\partial g)(\mathbf{x}^\star) \iff \mathbf{x}^\star = \mathrm{prox}_{\mathbf{H}^{-1}g}(\mathbf{x}^\star - \mathbf{H}^{-1}\nabla f(\mathbf{x}^\star)).$$

This expression shows that $\mathbf{x}^\star$ is a fixed point of $\mathcal{R}_{\mathbf{H}}(\cdot) := \mathrm{prox}_{\mathbf{H}^{-1}g}((\cdot) - \mathbf{H}^{-1}\nabla f(\cdot))$. Based on the fixed point principle, one can expect that the iterative sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ generated by $\mathbf{x}^{k+1} := \mathcal{R}_{\mathbf{H}}(x^k)$ converges to $\mathbf{x}^\star$. This observation is made rigorous below.

3 Our Variable Metric Framework

We first present a generic variable metric proximal framework for solving (1). Then, we specify this framework to obtain three variants: proximal gradient, proximal Newton and proximal quasi-Newton algorithms.

3.1 Generic Variable Metric Proximal Algorithmic Framework

Given $\mathbf{x}^k \in \mathrm{dom}\,(F)$ and an appropriate choice $\mathbf{H}_k \in \mathcal{S}^n_{++}$, since $f \in \mathcal{F}_{\mathrm{scl}}$, one can approximate f at $\mathbf{x}^k$ by the following quadratic model:

$$Q_{\mathbf{H}_k}(\mathbf{x}, \mathbf{x}^k) := f(\mathbf{x}^k) + \langle \nabla f(\mathbf{x}^k), \mathbf{x} - \mathbf{x}^k\rangle + \frac{1}{2}\langle \mathbf{H}_k(\mathbf{x} - \mathbf{x}^k), \mathbf{x} - \mathbf{x}_k\rangle. \tag{5}$$

Our algorithmic approach uses the variable metric forward-backward framework to generate a sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ starting from $\mathbf{x}^0 \in \mathrm{dom}\,(F)$ and update:

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k \tag{6}$$

where $\alpha_k \in (0, 1]$ is a given step-size and $\mathbf{d}^k$ is a search direction defined by:

$$\mathbf{d}^k := \mathbf{s}^k - \mathbf{x}^k, \quad \text{with} \quad \mathbf{s}^k := \mathrm{argmin}_{\mathbf{x}} \left\{ Q_{\mathbf{H}_k}(\mathbf{x}, \mathbf{x}^k) + g(\mathbf{x}) \right\}. \tag{7}$$

In the rest of this section, we explain how to determine the step size α_k in the iterative scheme (6) optimally for special cases of $\mathbf{H}_k$. For this, we need the following definitions:

$$\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k}, \quad r_k := M_f \|\mathbf{d}^k\|_2, \quad \text{and} \quad \beta_k := \|\mathbf{d}^k\|_{\mathbf{H}_k} = \langle \mathbf{H}_k\mathbf{d}^k, \mathbf{d}^k\rangle^{1/2}. \tag{8}$$

3.2 Proximal-Gradient Algorithm

When the variable matrix $\mathbf{H}_k$ is *diagonal* and g is proximally tractable, we can efficiently obtain the solution of the subproblem (7) in a distributed fashion or even in a closed form. Hence, we consider $\mathbf{H}_k = \mathbf{D}_k := \operatorname{diag}(\mathbf{D}_{k,1}, \cdots, \mathbf{D}_{k,n})$ with $\mathbf{D}_{k,i} > 0$, for $i = 1, \cdots, n$. Lemma 2, whose proof is in [19], provides a step-size selection procedure and proves the global convergence of this proximal-gradient algorithm.

Lemma 2. *Let $\{\mathbf{x}^k\}_{k\geq 0}$ be a sequence generated by (6) and (7) starting from $\mathbf{x}^0 \in \operatorname{dom}(F)$. For λ_k, r_k and β_k defined by (8), we consider the step-size α_k as:*

$$\alpha_k := \frac{1}{r_k}\ln\left(1 + \frac{\beta_k^2 r_k}{\lambda_k^2}\right), \tag{9}$$

If $\beta_k^2 r_k \leq (e^{r_k} - 1)\lambda_k^2$, then $\alpha_k \in (0, 1]$ and:

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - \frac{\beta_k^2}{r_k}\left[\left(1 + \frac{\lambda_k^2}{r_k\beta_k^2}\right)\ln\left(1 + \frac{\beta_k^2 r_k}{\lambda_k^2}\right) - 1\right]. \tag{10}$$

Moreover, this step-size α_k is optimal (w.r.t. the worst-case performance).

By our condition, the second term on the right-hand side of (10) is always positive, establishing that the sequence $\{F(\mathbf{x}^k)\}$ is decreasing. Moreover, as $e^{r_k} - 1 \geq r_k$, the condition $\beta_k^2 r_k \leq (e^{r_k} - 1)\lambda_k^2$ can be simplified to $\beta_k \leq \lambda_k$. It is easy to verify that this is satisfied whenever $\mathbf{D}_k \preceq \nabla^2 f(\mathbf{x}^k)$. In such cases, our step-size selection ensures the best decrease of the objective value regarding the self-concordant-like structure of f (and not the actual objective instance). When $\beta_k > \lambda_k$, we scale down $\mathbf{D}_k$ until $\beta_k \leq \lambda_k$. It is easy to prove that the number of backtracking steps to find $\mathbf{D}_{k,i}$ is time constant.

Now, by using our step-size (9), we can describe the proximal-gradient algorithm as in Algorithm 1.

Algorithm 1. (Proximal-gradient algorithm with a *diagonal variable metric*)

```
Initialization: Given $\mathbf{x}^0 \in \operatorname{dom}(F)$, and a tolerance $\varepsilon > 0$.
for $k = 0$ to $k_{\max}$ do
    1. Choose $\mathbf{D}_k \in \mathcal{S}^n_{++}$ (e.g., using $\mathbf{D}_k := L_k\mathbb{I}$, where $L_k$ is given by (11)).
    2. Compute the proximal-gradient search direction $\mathbf{d}^k$ as (7).
    3. Compute $\beta_k := \|\mathbf{d}^k\|_{\mathbf{D}_k}$, $r_k := M_f\|\mathbf{d}^k\|_2$ and $\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k}$.
    4. If $\beta_k \leq \varepsilon$ then terminate.
    5. If $\beta_k^2 r_k \leq (e^{r_k} - 1)\lambda_k^2$, then compute $\alpha_k := \frac{1}{r_k}\ln\left(1 + \frac{\beta_k^2 r_k}{\lambda_k^2}\right)$ and update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k\mathbf{d}^k$. Otherwise, set $\mathbf{x}^{k+1} := \mathbf{x}^k$ and update $\mathbf{D}_{k+1}$ from $\mathbf{D}_k$.
end for
```

We combine the above analysis to obtain the following proximal gradient algorithm for solving (1). The main step in Algorithm 1 is to compute the search direction $\mathbf{d}^k$ at Step 2, which is equivalent to the solution of the convex subproblem (7). The second main step is to compute $\lambda_k = \langle \nabla^2 f(\mathbf{x}^k)\mathbf{d}^k, \mathbf{d}^k\rangle^{1/2}$. This quantity requires the product of Hessian $\nabla^2 f(\mathbf{x}^k)$ of f and $\mathbf{d}^k$, but not the full-Hessian. It is clear that if $\beta_k = 0$ then $\mathbf{d}^k = 0$ and $\mathbf{x}^{k+1} \equiv \mathbf{x}^k$ and we obtain the solution of (1), i.e., $\mathbf{x}^k \equiv \mathbf{x}^\star$. The diagonal matrix $\mathbf{D}_k$ can be updated as $\mathbf{D}_{k+1} := c\mathbf{D}_k$ for a given factor $c > 1$.

We now explain how the new theory enhances the standard backtracking linesearch approaches. For simplicity, let us assume $\mathbf{D}_k := L_k\mathbb{I}$, where $\mathbb{I}$ is the identity matrix. By a careful inspection of (10), we see that $L_k = \sigma_{\max}(\nabla^2 f(\mathbf{x}^k))$ achieves the maximum guaranteed decrease (in the worst case sense) in the objective. There are many principled ways of approximating this constant based on the secant equation underlying the quasi-Newton methods. In Section 4, we use Barzilai-BenTal's rule:

$$L_k := \frac{\|\mathbf{y}^k\|_2^2}{\langle \mathbf{y}^k, \mathbf{s}^k\rangle}, \text{ where } \mathbf{s}^k := \mathbf{x}^k - \mathbf{x}^{k-1} \text{ and } \mathbf{y}^k := \nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^{k-1}). \quad (11)$$

We then deviate from the standard backtracking approaches. As opposed to, for instance, checking the Armijo-Goldstein condition, we use a *new analytic condition* (i.e., Step 5 of Algorithm 1), which is computationally cheaper in many cases. Our analytic step-size then further refines the solution based on the worst-case problem structure, even if the backtracking update satisfies the Armijo-Goldstein condition.

Surprisingly, our analysis also enables us to also establish local linear convergence as described in Theorem 1 under mild assumptions. The proof can be found in [19].

Theorem 1. *Let $\{\mathbf{x}^k\}_{k\geq 0}$ be a sequence generated by Algorithm 1. Suppose that the sub-level set $\mathcal{L}_F(F(\mathbf{x}^0)) := \{\mathbf{x} \in \mathrm{dom}(F) : F(\mathbf{x}) \leq F(\mathbf{x}^0)\}$ is bounded and $\nabla^2 f$ is nonsingular at some $\mathbf{x} \in \mathrm{dom}(f)$. Suppose further that $\mathbf{D}_k := L_k\mathbb{I} \succeq \tau\mathbb{I}_n$ for given $\tau > 0$. Then, $\{\mathbf{x}^k\}$ converges to $\mathbf{x}^\star$ the solution of (1). Moreover, if $\rho_* := \max\{L_k/\sigma^*_{\min} - 1, 1 - L_k/\sigma^*_{\max}\} < \frac{1}{2}$ for k sufficiently large then the sequence $\{\mathbf{x}^k\}$ locally converges to $\mathbf{x}^\star$ at a linear rate, where $\sigma^*_{\min}$ and $\sigma^*_{\max}$ are the smallest and the largest eigenvalues of $\nabla^2 f(\mathbf{x}^\star)$, respectively.*

Linear convergence: According to Theorem 1, linear convergence is only possible when the condition number κ of the Hessian at the true solution satisfies $\kappa = \sigma^*_{\max}/\sigma^*_{\min} < 3$. 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 Section 4). This is because the proof of Theorem 1 only needs the smallest and the largest eigenvalues of $\nabla^2 f(\mathbf{x}^\star)$, *restricted* to the subspaces of the union of $\mathbf{x}^\star - \mathbf{x}^k$ for k sufficiently large, to satisfy the conditions imposed by ρ_*. For instance, when g is based on the ℓ_1-norm/the nuclear norm, the differences $\mathbf{x}^\star - \mathbf{x}^k$ have at most twice the sparsity/rank of $\mathbf{x}^\star$ near convergence. Given such subspace restrictions, one can prove, via probabilistic assumptions on f (cf., [1]), that the restricted condition number is not only dramatically smaller than the full condition number κ of the Hessian $\nabla^2 f(\mathbf{x}^\star)$, but also it can even be dimension independent with high probability.

3.3 Proximal-Newton Algorithm

The case $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$ deserves a special attention as the step-size selection rule becomes explicit and backtracking-free. The resulting method is a *proximal-Newton* method and can be computationally attractive in certain big data problems due to its low iteration count.

The main step of the proximal-Newton algorithm is to compute the proximal-Newton search direction $\mathbf{d}^k$ as:

$$\mathbf{d}^k := \mathbf{s}^k - \mathbf{x}^k, \text{ where } \mathbf{s}^k := \operatorname*{argmin}_{\mathbf{x}} \left\{ Q_{\nabla^2 f(\mathbf{x}^k)}(\mathbf{x}, \mathbf{x}^k) + g(\mathbf{x}) \right\}. \tag{12}$$

Then, it updates the sequence $\left\{\mathbf{x}^k\right\}$ by:

$$\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k \mathbf{d}^k = (1-\alpha_k)\mathbf{x}^k + \alpha_k \mathbf{s}^k, \tag{13}$$

where $\alpha_k \in (0, 1]$ is the step size. If we set $\alpha_k = 1$ for all $k \geq 0$, then (13) is called the full-step proximal-Newton method. Otherwise, it is a damped-step proximal-Newton method.

First, we show how to compute the step size α_k in the following lemma, which is a direct consequence of Lemma 2 by taking $\mathbf{H}_k \equiv \nabla^2 f(\mathbf{x}^k)$.

Lemma 3. *Let $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ be a sequence generated by the proximal-Newton scheme* (13) *starting from $\mathbf{x}^0 \in \operatorname{dom}(F)$. Let λ_k and r_k be as defined by* (8)*. If we choose the step-size $\alpha_k = r_k^{-1}\ln(1+r_k)$ then:*

$$F(\mathbf{x}^{k+1}) \leq F(\mathbf{x}^k) - r_k^{-1}\lambda_k^2\left[\left(1 + r_k^{-1}\right)\ln(1+r_k) - 1\right]. \tag{14}$$

Moreover, this step-size α_k is optimal (*w.r.t. the worst-case performance*)*.*

Next, Theorem 2 proves the local quadratic convergence of the full-step proximal-Newton method, whose proof can be found in [19].

Theorem 2. *Suppose that the sequence $\left\{\mathbf{x}^k\right\}_{k\geq 0}$ is generated by* (13) *with full-step, i.e., $\alpha_k = 1$ for $k \geq 0$. If $r_k \leq \ln(4/3) \approx 0.28768207$ then it holds that:*

$$\left(\lambda_{k+1}/\sqrt{\sigma_{\min}^{k+1}}\right) \leq 2M_f\left(\lambda_k/\sqrt{\sigma_{\min}^k}\right)^2, \tag{15}$$

where $\sigma_{\min}^k$ is the smallest eigenvalue of $\nabla^2 f(\mathbf{x}^k)$. Consequently, if we choose $\mathbf{x}^0$ such that $\lambda_0 \leq \sigma_{\min}(\nabla^2 f(\mathbf{x}^0))\ln(4/3)$, then the sequence $\left\{\lambda_k/\sqrt{\sigma_{\min}^k}\right\}$ converges to zero at a quadratic rate.

Theorem 2 rigorously establishes where we can take full steps and still have quadratic convergence. Based on this information, we propose the proximal-Newton algorithm as in Algorithm 2.

The most remarkable feature of Algorithm 2 is that it does not require any globalization strategy such as backtracking line search for global convergence.

Complexity Analysis. First, we estimate the number of iterations needed when $\lambda_k \leq \sigma$ to reach the solution $\mathbf{x}^k$ such that $\frac{\lambda_k}{\sqrt{\sigma_k}} \leq \varepsilon$ for a given tolerance $\varepsilon > 0$.

Algorithm 2. (Prototype proximal-Newton algorithm)

```
**Initialization:** Given $\mathbf{x}^0 \in \mathrm{dom}\,(F)$ and $\sigma \in (0, \sigma_{\min}(\nabla^2 f(\mathbf{x}^0))\ln(4/3)]$.
**for** $k = 0$ **to** $k_{\max}$ **do**
  1. Compute $\mathbf{s}^k$ by(12). Then, define $\mathbf{d}^k := \mathbf{s}^k - \mathbf{x}^k$ and $\lambda_k := \|\mathbf{d}^k\|_{\mathbf{x}^k}$.
  2. If $\lambda_k \leq \varepsilon$, then terminate.
  3. If $\lambda_k > \sigma$, then compute $r_k := M_f\|\mathbf{d}^k\|_2$ and $\alpha_k := \frac{1}{r_k}\ln(1 + r_k)$; else $\alpha_k := 1$.
  4. Update $\mathbf{x}^{k+1} := \mathbf{x}^k + \alpha_k\mathbf{d}^k$.
**end for**
```

Based on the conclusion of Theorem 2, we can show that the number of iterations of Algorithm 2 when $\lambda_k > \sigma$ does not exceed $k_{\max} := \left\lfloor \log_2\left(\frac{\ln(2M_f\varepsilon)}{\ln(2\sigma)}\right)\right\rfloor$. Finally, we estimate the number of iterations needed when $\lambda_k > \sigma$. From Lemma 3, we see that for all $k \geq 0$ we have $\lambda_k \geq \sigma$ and $r_k \geq \sigma$. Therefore, the number of iterations is $\left\lfloor \frac{F(\mathbf{x}^0) - F(\mathbf{x}^\star)}{\psi(\sigma)}\right\rfloor$, where $\psi(\tau) := \tau\left((1 + \tau^{-1})\ln(1 + \tau) - 1)\right) > 0$.

3.4 Proximal Quasi-Newton Algorithm

In many applications, estimating the Hessian $\nabla^2 f(\mathbf{x}^k)$ can be costly even though the Hessian is given in a closed form (cf., Section 4). In such cases, variable metric strategies employing approximate Hessian can provide computation-accuracy tradeoffs. Among these approximations, applying quasi-Newton methods with BFGS updates for $\mathbf{H}_k$ would ensure its positive definiteness. Our analytic step-size procedures with backtracking automatically applies to the BFGS proximal-quasi Newton method, whose algorithm details and convergence analysis are omitted here.

4 Numerical Experiments

We use a variety of different real-data problems to illustrate the performance of our variable metric framework using a MATLAB implementation. We pick two advanced solvers for comparison: TFOCS [4] and PNOPT [8]. TFOCS hosts accelerated first order methods. PNOPT provides a several proximal-(quasi) Newton implementations, which has been shown to be quite successful in logistic regression problems [8]. Both use sophisticated backtracking linesearch enhancements. We benchmark all algorithms with performance profiles [7].

A performance profile is built based on a set $\mathcal{S}$ of n_s algorithms (solvers) and a collection $\mathcal{P}$ of n_p problems. We first build a profile based on computational time. We denote by $T_{p,s} :=$ *computational time required to solve problem p by solver s*. We compare the performance of algorithm s on problem p with the best performance of any algorithm on this problem; that is we compute the performance ratio $r_{p,s} := \frac{T_{p,s}}{\min\{T_{p,\hat{s}} : \hat{s} \in \mathcal{S}\}}$. Now, let $\tilde{\rho}_s(\tilde{\tau}) := \frac{1}{n_p}\mathrm{size}\left\{p \in \mathcal{P} : r_{p,s} \leq \tilde{\tau}\right\}$ for $\tilde{\tau} \in \mathbb{R}_+$. The function $\tilde{\rho}_s : \mathbb{R} \to [0, 1]$ is the probability for solver s that a performance ratio is within a factor $\tilde{\tau}$ of the best possible ratio. We use the term "performance profile" for the distribution function $\tilde{\rho}_s$ of a performance metric. In the following numerical examples, we plotted the performance profiles in $\log_2$-scale, i.e. $\rho_s(\tau) := \frac{1}{n_p}\text{size}\{p \in \mathcal{P} : \log_2(r_{p,s}) \leq \tau := \log_2 \tilde{\tau}\}$.

4.1 Sparse Logistic Regression

We consider the classical logistic regression problem of the form [20]:

$$\min_{\mathbf{x},\mu} \left\{ N^{-1} \sum_{j=1}^{N} \log\left(1 + e^{-y_j(\langle \mathbf{w}^{(j)}, \mathbf{x}\rangle + \mu)}\right) + \rho N^{-1/2} \|\mathbf{x}\|_1 \right\}, \tag{16}$$

where $\mathbf{x} \in \mathbb{R}^p$ is an unknown vector, μ is an unknown bias, and $y^{(j)}$ and $\mathbf{w}^j$ are observations where $j = 1, \cdots, N$. The logistic term in (16) is self-concordant-like with $M_f := \max \|\mathbf{w}^{(j)}\|_2$ [1]. In this case, the smooth term in (16) has Lipschitz gradient, hence several fast algorithms are applicable.

Figure 1 illustrates the performance profiles for computational time (left) and the number of prox-operations (right) using the 36 medium size problems[1]. For comparison, we use TFOCS-N07, which is Nesterov's 2007 two prox-method; and TFOCS-AT, which is Auslender and Teboulle's accelerated method, PNOPT with L-BFGS updates, and our algorithms: proximal gradient and proximal-Newton. From these performance profiles, we can observe that our proximal gradient is the best one in terms of computational time and the number of prox-operations. In terms of time, proximal-gradient solves upto 83.3% of problems with the best performance, while these numbers in TFOCS-N07 and PNOPT-LBFGS are 2.7%. Proximal Newton algorithm solves 11.1% problems with the best performance. In prox-operations, proximal-gradient is also the best one in 75% of problems.

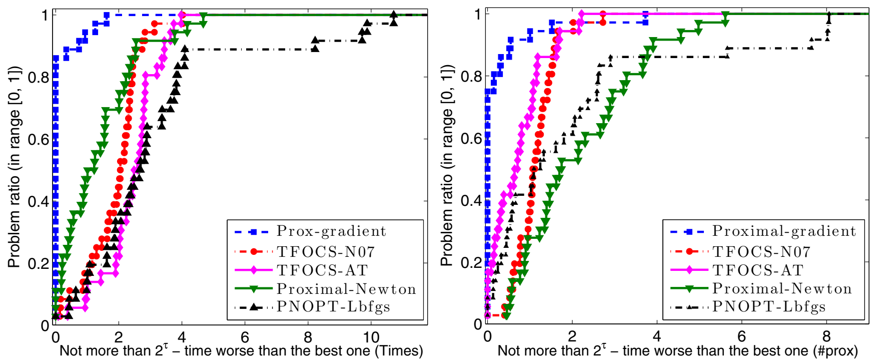

Fig. 1. Computational time (*left*) and number of prox-operations (*right*)

[1] Available at `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

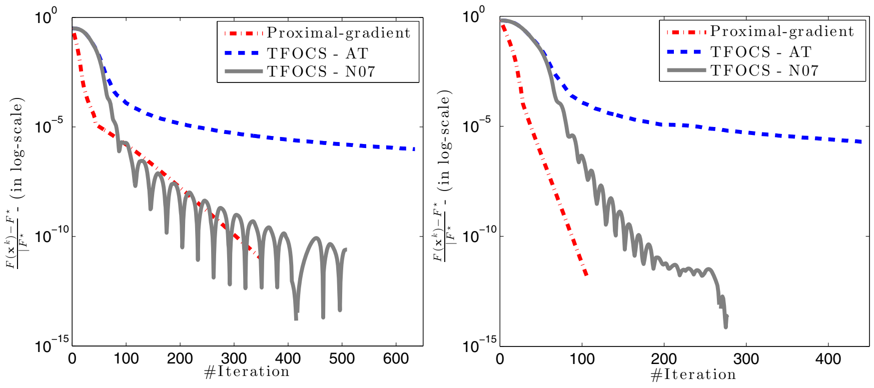

Fig. 2. *Left*: `rcv1_train.binary`, and *Right:* `real-sim`.

We now show an example convergence behavior of our proximal-gradient algorithm via two large-scale problems with $\rho = 0.1$. The first problem is `rcv1_train.binary` with the size $p = 20242$ and $N = 47236$ and the second one is `real-sim` with the size $p = 72309$ and $N = 20958$. For comparison, we use TFOCS-N07 and TFOCS-AT. For this example, PNOPT (with Newton, BFGS, and L-BFGS options) and our proximal-Newton do not scale and are omitted.

Figure 2 shows that our simple gradient algorithm locally exhibits linear convergence whereas the fast method TFOCS-AT shows a sublinear convergence rate. The variant TFOCS-N07 is the Nesterov's dual proximal algorithm, which exhibits oscillations but performs comparable to our proximal gradient method in terms of accuracy, time, and the total number of prox operations. The computational time and the number of prox-operations in these both problems are given as follows: Proximal-gradient: (15.67s, 698), (13.71s, 152); TFOCS-AT: (20.57s, 678), (33.82s, 466); TFOCS-N07: (17.09s, 1049), (22.08s, 568), respectively. For these data sets, the relative performance of the algorithms is surprisingly consistent across various regularization parameters.

4.2 Restricted Condition Number in Practice

The convergence plots in Figure 2 indicate that the linear convergence condition in Theorem 1 may be satisfied. *In fact, in all of our tests, the proximal gradient algorithm exhibits locally linear convergence.* Hence, to see if Remark 1 is grounded in practice, we perform the following test on the `a#a` dataset[1], consisting of small to medium problems. We first solve each problem with the proximal-Newton method up to 16 digits of accuracy to obtain $\mathbf{x}^\star$, and we calculate $\nabla^2 f(\mathbf{x}^\star)$. We then run our proximal gradient algorithm until convergence, and during its linear convergence, we record $\|\nabla^2 f(\mathbf{x}^\star)(\mathbf{x}^\star - \mathbf{x}^k)\|^2/\|\mathbf{x}^\star - \mathbf{x}^k\|_2^2$, and take the ratios of the maximum and the minimum to estimate the restricted condition number for each problem.

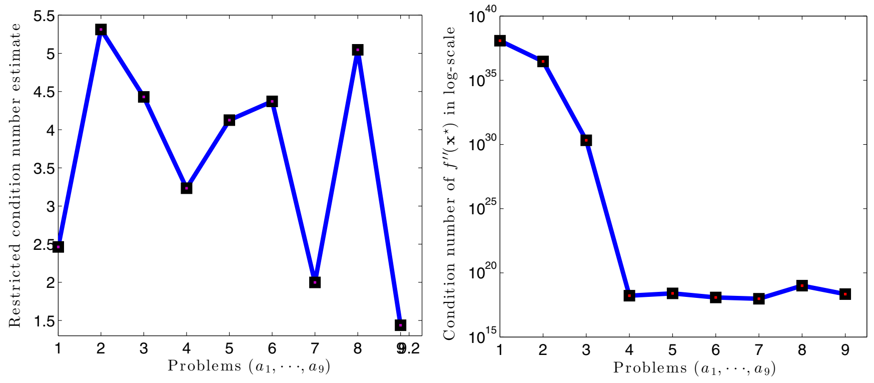

Fig. 3. Restricted condition number (*left*), and condition number (*right*) estimates

Figure 3 illustrates that while the condition number of the Hessian $\nabla^2 f(\mathbf{x}^\star)$ can be extremely large, as the algorithm navigates to the optimal solution $\mathbf{x}^\star$ through sparse subspaces, the restricted condition number estimates are in fact very close to 3. Given that algorithm still exhibit linear convergence for the cases $\# = 2, 3, 4, 5, 6, 8$ (where our condition cannot be met), we believe that the tightness of our convergence condition is an artifact of our proof and may be improved.

4.3 Sparse Multinomial Logistic Regression

For sparse multimonomial logistic regression, the underlying problem is formulated in the form of (1), which the objective function f is given as:

$$f(\mathbf{X}) := N^{-1} \sum_{j=1}^{N} \left[\log \left(1 + \sum_{i=1}^{m} e^{\langle \mathbf{w}^{(j)}, \mathbf{X}^{(i)} \rangle} \right) - \sum_{i=1}^{m} \mathbf{y}_i^{(j)} \langle \mathbf{w}^{(j)}, \mathbf{X}^{(i)} \rangle \right]. \qquad (17)$$

where $\mathbf{X}$ can be considered as a matrix variable of size $m \times p$ formed from $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(m)}$. Other vectors, $\mathbf{y}^{(j)}$ and $\mathbf{w}^{(j)}$ are given as input data for $j = 1, \ldots, N$. The function f has closed form gradient as well as Hessian. However, forming a full hessian matrix $\nabla^2 f(\mathbf{x})$ is especially costly in large scale problems when $N \gg 1$. In this case, proximal-quasi-Newton methods are more suitable. First, we show in Lemma 4 that f satisfies Definition 1, whose proof is in [19].

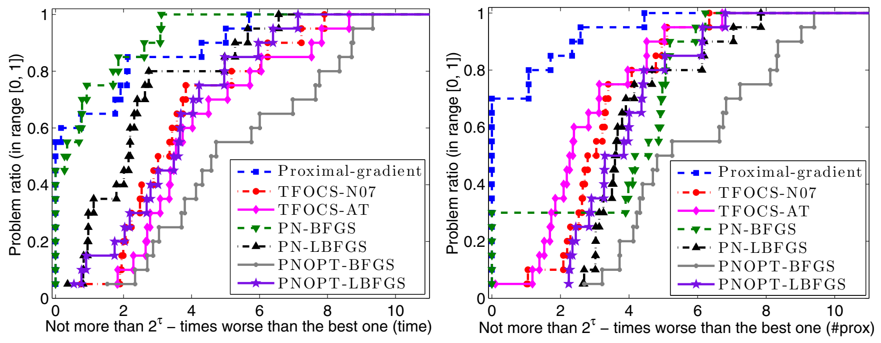

Fig. 4. Computational time (*left*), and number of prox-operations (*right*)

Lemma 4. *The function f defined by* (17) *is convex and self-concordant-like in the sense of Definition 1 with the parameter* $M_f := \sqrt{6}N^{-1}\max_{j=1,\dots,N}\|\mathbf{w}^{(j)}\|_2$.

The performance profiles of 20 small-to-medium size problems[1] are shown in Figure 4 in terms of computational time (left) as well as number of prox-operations (right), respectively. Both proximal-gradient method and proximal-Newton method with BFGS have good performance. They can solve unto 55% and 45% problems with the best time performance, respectively. These methods are also the best in terms of prox-operations (70% and 30%).

4.4 A Sytlized Example of a Non-Lipschitz Gradient Function for (1)

We consider the following convex composite minimization problem by modifying one of the canonical examples of geometric programming [6]:

$$\min_{\mathbf{x}\in\Omega}\left\{f(\mathbf{x}) := \sum_{i=1}^{m} e^{\mathbf{a}_i^T\mathbf{x}+b_i} + \mathbf{c}^T\mathbf{x}\right\} + g(\mathbf{x}), \tag{18}$$

where Ω is a simple convex set, $\mathbf{a}_i, \mathbf{c} \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$ are random, and g is the ℓ_1-norm. After some algebra, we can show that f satisfies Definition 1 with $M_f := \max\{\|\mathbf{a}_i\|_2 : 1 \le i \le m\}$. Unfortunately, f does not have Lipschitz continuous gradient in $\mathbb{R}^n$.

We implement our proximal-gradient algorithm and compare it with TFOCS and PNOPT-LBFGS. However, TFOCS breaks down in running this example due to the estimation of Lipschitz constant, while PNOPT is rather slow. Several tests on synthetic data show that our algorithm outperforms PNOPT-LBFGS. As an example, we show the convergence behavior of both these methods in Figure 5 where we plot the accuracy of the objective values w.r.t. the number of prox-operators for two cases of $\varepsilon = 10^{-6}$ and $\varepsilon = 10^{-12}$, respectively. As we can see from this figure that our prox-gradient method requires many fewer prox-operations to achieve a very high accuracy compared to PNOPT. Moreover, our method is also 20 to 40 times faster than PNOPT in this numerical test.

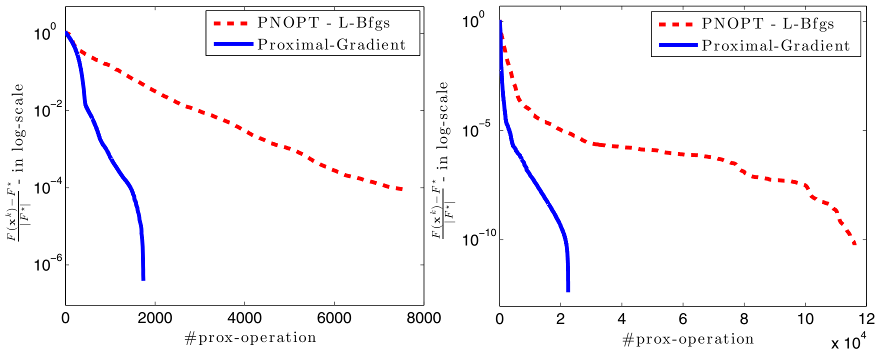

Fig. 5. Relative objective values w.r.t. #prox: *left*: $\varepsilon = 10^{-6}$, and *right*: $\varepsilon = 10^{-12}$

5 Conclusions

Convex optimization efficiency relies significantly on the structure of the objective functions. In this paper, we propose a variable metric method for minimizing the sum of a self-concordant-like convex function and a proximally tractable convex function. Our framework is applicable in several interesting machine learning problems and do not rely on the usual Lipschitz gradient assumption on the smooth part for its convergence theory. A highlight of this work is the new analytic step-size selection procedure that enhances backtracking procedures. Thanks to this new approach, we can prove that the basic gradient variant of our framework has improved local convergence guarantees under certain conditions while the tuning-free proximal Newton method has locally quadratic convergence. While our assumption on the restricted condition number in Theorem 1 is not deterministically verifiable *a priori*, we provide empirical evidence that it can hold in many practical problems. Numerical experiments on different applications that have both self-concordant-like and Lipschitz gradient properties demonstrate that the gradient algorithm based on the former assumption can be more efficient than the fast algorithms based on the latter assumption. As a result, we plan to look into fast versions of our gradient scheme as future work.

References

1. Bach, F.: Self-concordant analysis for logistic regression. Electron. J. Statist. 4, 384–414 (2010)
2. Bach, F.: Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression (2013)
3. Beck, A., Teboulle, M.: A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. SIAM J. Imaging Sciences 2(1), 183–202 (2009)
4. Becker, S., Candès, E.J., Grant, M.: Templates for convex cone problems with applications to sparse signal recovery. Mathematical Programming Computation 3(3), 165–218 (2011)

5. Becker, S., Fadili, M.J.: A quasi-Newton proximal splitting method. In: Adv. Neural Information Processing Systems (2012)
6. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
7. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. 91, 201–213 (2002)
8. Lee, J.D., Sun, Y., Saunders, M.A.: Proximal Newton-type methods for convex optimization. SIAM J. Optim. 24(3), 1420–1443 (2014)
9. Mine, H., Fukushima, M.: A minimization method for the sum of a convex function and a continuously differentiable function. J. Optim. Theory Appl. 33, 9–23 (1981)
10. Nesterov, Y.: Introductory lectures on convex optimization: a basic course. Applied Optimization, vol. 87. Kluwer Academic Publishers (2004)
11. Nesterov, Y.: Gradient methods for minimizing composite objective function. Math. Program. 140(1), 125–161 (2013)
12. Nesterov, Y., Nemirovski, A.: Interior-point Polynomial Algorithms in Convex Programming. Society for Industrial Mathematics (1994)
13. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer Series in Operations Research and Financial Engineering. Springer (2006)
14. Parikh, N., Boyd, S.: Proximal algorithms. Foundations and Trends in Optimization 1(3), 123–231 (2013)
15. Robinson, S.M.: Strongly Regular Generalized Equations. Mathematics of Operations Research 5(1), 43–62 (1980)
16. Rockafellar, R.T.: Convex Analysis. Princeton Mathematics Series, vol. 28. Princeton University Press (1970)
17. Schmidt, M., Roux, N.L., Bach, F.: Convergence rates of inexact proximal-gradient methods for convex optimization. In: NIPS, Granada, Spain (2011)
18. Tran-Dinh, Q., Kyrillidis, A., Cevher, V.: Composite self-concordant minimization. J. Mach. Learn. Res. 15, 1–54 (2014) (accepted)
19. Tran-Dinh, Q., Li, Y.-H., Cevher, V.: Composite convex minimization involving self-concordant-like cost functions. LIONS-Tech. Report., 1–19 (2015), `http://arxiv.org/abs/1502.01068`
20. Yuan, Y.X.: Recent advances in numerical methods for nonlinear equations and nonlinear least squares. Numerical Algebra, Control and Optimization 1(1), 15–34 (2011)