Geometric descent method for convex composite minimization

Shixiang Chen∗ Shiqian Ma†

January 19, 2017

Abstract

In this paper, we extend the geometric descent method recently proposed by Bubeck, Lee and Singh [5] to solving nonsmooth and strongly convex composite problems. We prove that the resulting algorithm, GeoPG, converges with a linear rate \((1 - 1/√κ)\), thus achieves the optimal rate among first-order methods, where \(κ\) is the condition number of the problem. Numerical results on linear regression and logistic regression with elastic net regularization show that GeoPG compares favorably with Nesterov’s accelerated proximal gradient method, especially when the problem is ill-conditioned.

1 Introduction

Recently, Bubeck, Lee and Singh proposed a geometric descent method (GeoM) for minimizing a smooth and strongly convex function [5]. They showed that GeoM achieves the same optimal rate as Nesterov’s accelerated gradient method (AGM) [13, 14]. In this paper, we provide an extension of GeoM that can minimize a nonsmooth function in composite form as follows:

\[
\min_{x \in \mathbb{R}^n} F(x) := f(x) + h(x),
\]

where \(f\) is \(\alpha\)-strongly convex and \(\beta\)-smooth (i.e., \(\nabla f\) is Lipschitz continuous with Lipschitz constant \(\beta\)), \(h\) is a closed nonsmooth convex function with simple proximal mapping. Commonly seen examples of \(h\) include \(\ell_1\) norm, \(\ell_2\) norm, nuclear norm, and so on.

If \(h\) vanishes, then the objective function of (1.1) becomes smooth and strongly convex. In this case, it is known that AGM converges with a linear rate \((1 − 1/√κ)\), which is optimal among all first-order methods, where \(κ = \beta/\alpha\) is the condition number. However, AGM lacks clear geometric intuition which makes it difficult to interpret. Recently, there have been many works on attempting to explain AGM or designing new algorithms with the same optimal rate (see, e.g. [17, 1, 5, 12, 19]). In particular, the GeoM method proposed in [5] has a clear geometric intuition that is in the flavor of the ellipsoid method. The follow-up works [4, 7] attempted to improve the performance of GeoM by utilizing the gradient information from the past with a “limited-memory” idea. Moreover, Drusvyatskiy, Fazel and Roy [7] showed how to extend the basic version of GeoM (with convergence rate \((1 − 1/κ)\)) to the composite problem (1.1). However, it was not clear how to extend the optimal GeoM to (1.1) and the authors posed this as an open question. In this paper, we settle this question by proposing the GeoPG (geometric proximal gradient) algorithm that can solve the composite problem (1.1), and show how to incorporate various techniques to improve the performance of this algorithm.

∗Department of SEEM, The Chinese University of Hong Kong, Shatin, NT, Hong Kong. Email: sxchen@se.cuhk.edu.hk
†Department of SEEM, The Chinese University of Hong Kong, Shatin, NT, Hong Kong. Email: sqma@se.cuhk.edu.hk
Notation. We use \( B(c, r^2) = \{ x \| x - c \|^2 \leq r^2 \} \) to denote the ball with center \( c \) and radius \( r \). We use \( \text{Line}(x, y) \) to denote the line that connects \( x \) and \( y \), i.e., \( \{ x + s(y - x), s \in \mathbb{R}^n \} \). We use the operation \( \text{line}_\text{search}(x, y) \) to denote the minimizer of \( f \) on \( \text{Line}(x, y) \), i.e.,

\[
\text{line}_\text{search}(x, y) = \text{argmin}_{z} \{ f(z) \mid z = x + t(y - x), t \in \mathbb{R} \}.
\]

The rest of this paper is organized as follows. In Section 2, we briefly review the GeoM method for solving smooth and strongly convex problem. In Section 3, we provide our GeoPG algorithm for solving nonsmooth problem and analyze its convergence rate. We address two practical issues of the proposed method in Section 4 and incorporate two techniques: backtracking and limited memory, to take care of these issues. In Section 5, we report some numerical results of comparing GeoPG with Nesterov’s accelerated proximal gradient method for solving linear regression and logistic regression problems with elastic net regularization. Finally, we conclude the paper in Section 6.

2 Geometric descent method for smooth problem

The GeoM method [5] solves the smooth and strongly convex problem \( \min_{x} f(x) \), whose optimal solution and optimal value are denoted as \( x^* \) and \( f^* \), respectively. We first briefly describe the basic idea of the ordinary GeoM. Since \( f \) is \( \alpha \)-strongly convex, the following inequality holds

\[
f(x) + \langle \nabla f(x), y - x \rangle + \frac{\alpha}{2} \| y - x \|^2 \leq f(y), \quad \forall x, y \in \mathbb{R}^n. \tag{2.1}
\]

By letting \( y = x^* \) in (2.1), one obtains that

\[
x^* \in B \left( x^{++}, \frac{\| \nabla f(x) \|^2}{\alpha^2} - \frac{2}{\alpha} (f(x) - f^*) \right), \quad \forall x \in \mathbb{R}^n. \tag{2.2}
\]

For any \( x \in \mathbb{R}^n \), we denote \( x^+ = x - \frac{1}{\beta} \nabla f(x) \) and \( x^{++} = x - \frac{1}{\alpha} \nabla f(x) \). Note that the \( \beta \)-smoothness of \( f \) implies

\[
f(x^+) \leq f(x) - \frac{1}{2\beta} \| \nabla f(x) \|^2, \quad \forall x \in \mathbb{R}^n. \tag{2.3}
\]

Combining (2.2) and (2.3) yields

\[
x^* \in B \left( x^{++}, \left( 1 - \frac{1}{\kappa} \right) \frac{\| \nabla f(x) \|^2}{\alpha^2} - \frac{2}{\alpha} (f(x^+) - f^*) \right), \quad \forall x \in \mathbb{R}^n.
\]

As a result, suppose initially we have a ball \( B(x_0, R_0^2) \) that contains \( x^* \), then it follows that

\[
x^* \in B(x_0, R_0^2) \cap B \left( x_0^{++}, \left( 1 - \frac{1}{\kappa} \right) \frac{\| \nabla f(x_0) \|^2}{\alpha^2} - \frac{2}{\alpha} (f(x_0^{++}) - f^*) \right). \tag{2.4}
\]

Some simple algebraic calculation shows that the squared radius of the minimum enclosing ball of the right hand side of (2.4) is no larger than \( R_0^2 (1 - 1/\kappa) \), i.e., there exists some \( x_1 \in \mathbb{R}^n \) such that \( x^* \in B(x_1, R_0^2 (1 - 1/\kappa)) \). Therefore, the squared radius of the initial ball shrinks by a factor \((1 - 1/\kappa)\). Repeating this process yields a linear convergent sequence \( \{x_k\} \) with convergence rate \((1 - 1/\kappa)\):

\[
\|x_k - x^*\|^2 \leq \left( 1 - \frac{1}{\kappa} \right)^k R_0^2.
\]

The optimal GeoM (with linear convergence rate \((1 - 1/\sqrt{\kappa})\)) is a bit more involved. The optimal GeoM maintains two balls containing \( x^* \) in each iteration, whose centers are \( c_k \) and
For a given step size $t$ and $x$, and $x^{++}_{k+1}$ are obtained as follows. First, $x^{+}_{k+1} = \text{line-search}(c_k, x^+_k)$. Second, $c_{k+1}$ (resp. $R^2_{k+1}$) is the center (resp. squared radius) of the ball (given by Lemma 2.1) that contains

$$B\left(c_k, R^2_k - \frac{\|\nabla f(x_{k+1})\|^2}{\alpha^2}\right) \cap B\left(x^{++}_{k+1}, \left(1 - \frac{1}{\kappa}\right) \frac{\|\nabla f(x_{k+1})\|^2}{\alpha^2}\right).$$

Calculating $c_{k+1}$ and $R_{k+1}$ is simple and we refer to Algorithm 1 of [5] for details. By applying Lemma 2.1 with $x_A = c_k$, $r_A = R_k$, $r_B = \|\nabla f(x_{k+1})\|/\alpha$, $\epsilon = 2 \kappa$ and $\delta = \frac{\alpha}{\kappa} (f(x^+_k) - f(x^*))$, it follows that $R_{k+1} = \left(1 - 1/\sqrt{\kappa}\right)R_k^2$, which further implies that

$$\|x^* - c_k\|^2 \leq \left(1 - \frac{1}{\sqrt{\kappa}}\right) R_0^2,$$

i.e., the optimal GeoM converges with linear rate $(1 - 1/\sqrt{\kappa})$.

**Lemma 2.1** (see [5, 7]). Fix centers $x_A, x_B \in \mathbb{R}^n$ and squared radii $r^2_A, r^2_B > 0$. Also fix $\epsilon \in (0, 1)$ and suppose $\|x_A - x_B\|^2 \geq r^2_B$. There exists a new center $c \in \mathbb{R}^n$ such that for any $\delta > 0$, we have

$$B(x_A, r^2_A - \epsilon r^2_B - \delta) \cap B(x_B, r^2_B (1 - \epsilon) - \delta) \subseteq B(c, (1 - \sqrt{\kappa}) r^2_A - \delta).$$

### 3 Geometric descent method for composite convex problem

Drusvyatskiy, Fazel and Roy [7] extended the ordinary GeoM to solving composite problem (1.1). However, it was not clear how to extend the optimal GeoM to solving (1.1). We resolve this problem in this section.

#### 3.1 Proximal gradient

First, we need some preparation on properties of proximal gradient. It follows from the $\beta$-smoothness of $f$

$$F(y) \leq Q_t(y, x) := f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2t} \|y - x\|^2 + h(y), \ \forall x, y \in \mathbb{R}^n, 0 < t \leq \frac{1}{\beta}.$$

For a given step size $t \in (0, 1/\beta]$, The proximal gradient of $F$ at point $x$ is defined as

$$G_t(x) = (x - x^+)/t,$$

where $x^{+} := \text{Prox}_h(x - t\nabla f(x))$

where the proximal mapping $\text{Prox}_h(\cdot)$ is defined as

$$\text{Prox}_h(x) = \arg\min_z h(z) + \frac{1}{2} \|z - x\|^2.$$

It should be noted that $x^+ = x - tG_t(x)$. It is easy to verify that

$$G_t(x) \in \nabla f(x) + \partial h(x^+),$$

and $x$ is optimal iff $G_t(x) = 0$. Moreover, we define $x^{++} = x - G_t(x)/\alpha$. Note that both $x^+$ and $x^{++}$ are related to some given step size $t$, and we have omitted $t$ whenever there is no ambiguity.

The following lemma is useful to our analysis.
Lemma 3.1. Given point \( x \) and step size \( t \in (0, 1/\beta] \), denote \( x^+ = x - tG_t(x) \). The following inequality holds for any \( y \in \mathbb{R}^n \):

\[
F(y) \geq F(x^+) + \langle G_t(x), y - x \rangle + \frac{t}{2}\|G_t(x)\|^2 + \frac{\alpha}{2}\|y - x\|^2. \tag{3.2}
\]

**Proof.** From the \( \beta \)-smoothness of \( f \), we have

\[
f(x^+) \leq f(x) - t\langle \nabla f(x), G_t(x) \rangle + \frac{t}{2}\|G_t(x)\|^2. \tag{3.3}
\]

Combining (3.3) with (2.1) yields that

\[
F(x^+) \leq f(y) - \langle \nabla f(x), y - x \rangle - \frac{\alpha}{2}\|y - x\|^2 - t\langle \nabla f(x), G_t(x) \rangle + \frac{t}{2}\|G_t(x)\|^2 + h(x^+)
\]

\[
= F(y) - \frac{\alpha}{2}\|y - x\|^2 + \frac{t}{2}\|G_t(x)\|^2 + h(x^+) - h(y) - \langle \nabla f(x) - G_t(x), y - x^+ \rangle - \langle G_t(x), y - x^+ \rangle
\]

\[
\leq F(y) - \frac{\alpha}{2}\|y - x\|^2 + \frac{t}{2}\|G_t(x)\|^2 - \langle G_t(x), y - x^+ \rangle,
\]

where the last inequality is due to the convexity of \( h \) and (3.1). \( \square \)

The following lemma is from [11].

**Lemma 3.2** (see Lemma 3.9 of [11]). For \( t \in (0, 1/\beta] \), \( G_t(x) \) is strongly monotone, i.e.,

\[
\langle G_t(x) - G_t(y), x - y \rangle \geq \frac{\alpha}{2}\|x - y\|^2, \forall x, y.
\]

3.2 The GeoPG Algorithm

In this subsection, we describe our geometric proximal gradient descent method (GeoPG) for solving (1.1). For simplicity, throughout this subsection, we fix the step size \( t = 1/\beta \). The key observation is that in order to design GeoPG, in the \( k \)-th iteration, one has to find \( x_k \) that lies on Line\((x_{k-1}, c_{k-1})\) such that the following two inequalities hold:

\[
F(x_k^+) \leq F(x_{k-1}^+) - \frac{t}{2}\|G_t(x_k)\|^2, \text{ and } \|x_k^+ - c_{k-1}\|^2 \geq \frac{1}{\alpha^2}\|G_t(x_k)\|^2. \tag{3.4}
\]

The following lemma provides a sufficient condition for (3.4).

**Lemma 3.3.** (3.4) holds if \( x_k \) satisfies:

\[
\langle x_k^+ - x_k, x_{k-1}^+ - x_k \rangle \leq 0, \text{ and } \langle x_k^+ - x_k, x_k - c_{k-1} \rangle \geq 0. \tag{3.5}
\]

**Proof.** Assume (3.3) holds. By letting \( y = x_{k-1}^+ \) and \( x = x_k \) in (3.2), we have

\[
F(x_k^+) \leq F(x_{k-1}^+) - \langle G_t(x_k), x_{k-1}^+ - x_k \rangle - \frac{t}{2}\|G_t(x_k)\|^2 - \frac{\alpha}{2}\|x_{k-1}^+ - x_k\|^2
\]

\[
= F(x_{k-1}^+) + \frac{1}{t}\langle x_k^+ - x_k, x_{k-1}^+ - x_k \rangle - \frac{t}{2}\|G_t(x_k)\|^2 - \frac{\alpha}{2}\|x_{k-1}^+ - x_k\|^2
\]

\[
\leq F(x_{k-1}^+) - \frac{t}{2}\|G_t(x_k)\|^2,
\]

where the last inequality is due to (3.3). Moreover, from the definition of \( x_k^+ \) and (3.3) it is easy to see

\[
\|x_k^+ - c_{k-1}\|^2 = \|x_k - c_{k-1}\|^2 + \frac{2}{\alpha t}\langle x_k^+ - x_k, x_k - c_{k-1} \rangle + \frac{1}{\alpha^2}\|G_t(x_k)\|^2 \geq \frac{1}{\alpha^2}\|G_t(x_k)\|^2.
\]

\( \square \)
Therefore, we only need to find $x_k$ such that (3.5) holds. To do so, we define the following functions for given $x, c (x \neq c)$, and $t > 0$:

$$\phi_{t,x,c}(z) = \langle z^+ - x, z - c \rangle, \forall z \in \mathbb{R}^n$$

and $\tilde{\phi}_{t,x,c}(s) = \phi_{t,x,c}(x + s(c - x)), \forall s \in \mathbb{R}$.

The functions $\phi_{t,x,c}(z)$ and $\tilde{\phi}_{t,x,c}(s)$ have the following properties.

**Lemma 3.4.** (i) $\phi_{t,x,c}(z)$ is Lipschitz continuous. (ii) $\tilde{\phi}_{t,x,c}(s)$ strictly monotonically increases for $s \in \mathbb{R}$.

**Proof.** We prove (i) first.

$$|\phi_{t,x,c}(z_1) - \phi_{t,x,c}(z_2)| = |\langle z_1^+ - z_1 - (z_2^+ - z_2), x - c \rangle| \leq \| z_1^+ - z_2^+ - (z_1 - z_2) \| \| x - c \|$$

$$\leq (\| \text{prox}_{t\phi}(z_1 - t\nabla f(z_1)) - \text{prox}_{t\phi}(z_2 - t\nabla f(z_2)) \| + \| z_1 - z_2 \|) \| x - c \|$$

$$\leq (2 + t\beta) \| x - c \| \| z_1 - z_2 \|,$$

where the last inequality is due to the non-expansiveness of the proximal mapping operation.

We now prove (ii). For $s_1 < s_2$, let $z_1 = x + s_1(c - x)$ and $z_2 = x + s_2(c - x)$. We have

$$\tilde{\phi}_{t,x,c}(s_2) - \tilde{\phi}_{t,x,c}(s_1) = \langle z_2^+ - z_2 - (z_1^+ - z_1), x - c \rangle = \frac{t}{s_2 - s_1}(G_t(z_2) - G_t(z_1), z_2 - z_1)$$

$$\geq \frac{\alpha t}{2}(s_2 - s_1) \| x - c \|^2 > 0,$$

where the first inequality follows from Lemma 3.2. \hfill \Box

We are now ready to describe how to find $x_k$ such that (3.5) holds. This is summarized in Lemma 3.5.

**Lemma 3.5.** There are two ways to find $x_k$ such that (3.5) holds.

- (i) If $\tilde{\phi}_{t,x_{k-1}^+, c_{k-1}}(1) \leq 0$, then (3.5) holds by setting $x_k := c_{k-1}$; If $\tilde{\phi}_{t,x_{k-1}^+, c_{k-1}}(0) > 0$, then (3.5) holds by setting $x_k := x_{k-1}^+$. If $\tilde{\phi}_{t,x_{k-1}^+, c_{k-1}}(1) > 0$ and $\tilde{\phi}_{t,x_{k-1}^+, c_{k-1}}(0) < 0$, then there exists $s \in [0, 1]$ such that $\tilde{\phi}_{t,x_{k-1}^+, c_{k-1}}(s) = 0$. As a result, (3.5) holds by setting $x_k := x_{k-1}^+ + s(c_{k-1} - x_{k-1}^-)$.

- (ii) If $\tilde{\phi}_{t,x_{k-1}^+, c_{k-1}}(0) \geq 0$, then (3.5) holds by setting $x_k := x_{k-1}^+$. If $\tilde{\phi}_{t,x_{k-1}^+, c_{k-1}}(0) < 0$, then there exists $s \geq 0$, such that $\tilde{\phi}_{t,x_{k-1}^+, c_{k-1}}(s) = 0$. As a result, (3.5) holds by setting $x_k := x_{k-1}^+ + s(c_{k-1} - x_{k-1}^-)$.

**Proof.** Case (i) directly follows from the Mean-Value Theorem. Case (ii) follows from the monotonicity and continuity of $\tilde{\phi}_{t,x_{k-1}^+, c_{k-1}}$ from Lemma 3.3. \hfill \Box

It is indeed very easy to find $x_k$ satisfying the two cases in Lemma 3.5. Specifically, for case (i) of Lemma 3.5, we can use bisection method to find the zero of $\tilde{\phi}_{t,x_{k-1}^+, c_{k-1}}$ in the closed interval $[0, 1]$. In practice, we found that the Brent-Dekker method [3, 6] performs much better than bisection method, so we used the Brent-Dekker method in our numerical experiments. For case (ii) of Lemma 3.5, we can use the semi-smooth Newton method to find the zero of $\tilde{\phi}_{t,x_{k-1}^+, c_{k-1}}$ in the interval $[0, +\infty)$. The semi-smooth Newton method updates $s_k$ via

$$s_{k+1} = s_k - \left[ \Phi_{t,x_{k-1}^+, c_{k-1}}(s) \right]^{-1} \tilde{\phi}_{t,x_{k-1}^+, c_{k-1}}(s),$$

where $\Phi_{t,x_{k-1}^+, c_{k-1}}(s)$ is the generalized derivative of $\tilde{\phi}_{t,x_{k-1}^+, c_{k-1}}$, which can be computed by $\Phi_{t,x_{k-1}^+, c_{k-1}}(s) = \| d \|^2 - (d, G(u)(I - t\nabla^2 g(z))d)$, where $d = c_{k-1} - x_{k-1}^+$, $G(u)$ is the generalized
derivative of Prox$_t(u)$ and $u = z - t\nabla f(z)$. In our numerical experiment, we implemented the global semi-smooth Newton method \cite{9, 10} and obtained very encouraging results.

These two procedures are described in Algorithms 1 and 2 respectively. Based on the discussions above, we know that $x_k$ generated by these two algorithms satisfies (3.5) and hence (3.4).

**Algorithm 1** The first procedure to find $x_k$ from given $x_{k-1}^+$ and $c_{k-1}$

1. if $\langle (x_{k-1}^+)^+ - x_{k-1}^+, x_{k-1}^+ - c_{k-1} \rangle \geq 0$ then
2. Set $x_k = x_{k-1}^+$
3. else if $\langle c_{k-1} - c_{k-1}, x_{k-1}^+ - c_{k-1} \rangle \leq 0$ then
4. Set $x_k = c_{k-1}$
5. else
6. Use Brent-Dekker method to find $s \in [0, 1]$ such that $\tilde{\phi}_{t,x_{k-1}^+,c_{k-1}}(s) = 0$ and set $x_k = x_{k-1}^+ + s(c_{k-1} - x_{k-1}^+)$.
7. end if

**Algorithm 2** The second procedure to find $x_k$ from given $x_{k-1}^+$ and $c_{k-1}$

1. if $\langle (x_{k-1}^+)^+ - x_{k-1}^+, x_{k-1}^+ - c_{k-1} \rangle \geq 0$ then
2. Set $x_k = x_{k-1}^+$
3. else
4. Use the global semi-smooth Newton method \cite{9, 10} to find the root $s \in [0, +\infty)$ of $\tilde{\phi}_{t,x_{k-1}^+,c_{k-1}}(s)$, and set $x_k = x_{k-1}^+ + s(c_{k-1} - x_{k-1}^+)$.
5. end if

We are now ready to present our GeoPG algorithm for solving (1.1). This is described in Algorithm 3. Note that the Step 6 can be easily conducted as shown in \cite[Algorithm 1]{9}.

**Algorithm 3** GeoPG: geometric proximal gradient descent for composite convex minimization

Require: Parameters $\alpha$, $\beta$, and initial point $x_0$.

1. Set $c_0 = x_0^+$, $R_0^2 = \frac{\|G_t(x_0)\|^2}{\alpha^2}(1 - 1/\kappa)$
2. for $k = 1, 2, \ldots$ do
3. Use Algorithm 1 or 2 to find $x_k$.
4. Set $x_A := x_k^+ = x_k - \frac{G_t(x_k)}{\alpha}$, and $R_A^2 = \frac{\|G_t(x_k)\|^2}{\alpha^2}(1 - 1/\kappa)$.
5. Set $x_B = c_k$, and $R_B^2 = R_{k-1}^2 - \frac{2}{\alpha}(F(x_{k-1}) - F(x_k^+))$.
6. Compute $B(c_k, R_B^2)$: the minimum enclosing ball of $B(x_A, R_A^2) \cap B(x_B, R_B^2)$. This can be done using Algorithm 1 in \cite{9}
7. end for

### 3.3 Convergence analysis of GeoPG

We are now ready to present our main convergence result for GeoPG. The result is stated in Theorem 3.6. For simplicity, we still assume that the fixed step size $t = 1/\beta$ is used.

**Theorem 3.6.** Given initial point $x_0$, we set $R_0^2 = \frac{\|G_t(x_0)\|^2}{\alpha^2}(1 - 1/\kappa)$. Suppose sequence $\{(x_k, c_k, R_k)\}$ is generated by Algorithm 3. Suppose $x^*$ is the optimal solution of (1.1) and $F^*$ is the optimal objective value. For any $k \geq 0$, one has $x^* \in B(c_k, R_k^2)$, and $R_{k+1} \leq (1-1/\sqrt{\kappa})R_k^2$. 


and thus
\[ \|x^* - c_k\|^2 \leq (1 - 1/(\sqrt{\kappa}))^k R_0^2, \]
and
\[ F(x^*_{k+1}) - F^* \leq \frac{\alpha}{2}(1 - 1/(\sqrt{\kappa}))^k R_0^2. \] (3.6)

Proof. We prove a stronger result by induction that for every \( k \geq 0 \), one has
\[ x^* \in B \left( c_k, R_k^2 - \frac{2}{\alpha} (F(x^*_k) - F^*) \right). \] (3.7)

By letting \( y = x^* \) and \( t = 1/\beta \) in (3.2) we have
\[ \|x^* - x^+\|^2 \leq (1 - \frac{1}{\kappa}) \frac{\|G(x)^2\|}{\alpha^2} - \frac{2}{\alpha} (F(x^+) - F^*), \]
which implies that
\[ x^* \in B \left( x^+, \frac{\|G(x)^2\|}{\alpha^2} (1 - \frac{1}{\kappa}) - \frac{2}{\alpha} (F(x^+) - F^*) \right). \] (3.8)

Setting \( x = x_0 \) in (3.8) shows that (3.7) holds for \( k = 0 \). We now assume that (3.7) holds for some \( k \geq 0 \). Combining (3.7) and the first inequality of (3.4) yields,
\[ x^* \in B \left( c_k, R_k^2 - \frac{1}{\alpha^2 \kappa} \|G_t(x_{k+1})\|^2 - \frac{2}{\alpha} (F(x^+_k) - F^*) \right). \] (3.9)

By setting \( x = x_{k+1} \) in (3.8) we get
\[ x^* \in B \left( x^+_{k+1}, \frac{\|G_t(x_{k+1})\|^2}{\alpha^2} (1 - \frac{1}{\kappa}) - \frac{2}{\alpha} (F(x^+_{k+1}) - F^*) \right). \] (3.10)

We now apply Lemma 2.1 to (3.9) and (3.10). Specifically, we set \( x_B = x^+_{k+1}, \ x_A = c_k, \ \epsilon = \frac{1}{\kappa}, \ r_A = R_k, \ r_B = \frac{\|G_t(x_{k+1})\|}{\alpha}, \ \delta = \frac{2}{\alpha} (F(x^+_k) - F^*), \) and note that \( \|x_A - x_B\|^2 \geq \epsilon r_B^2 \) because of the second inequality of (3.4), then Lemma 2.1 indicates that there exists \( c_{k+1} \), such that
\[ x^* \in B \left( c_{k+1}, (1 - 1/(\sqrt{\kappa})) R_k^2 - \frac{2}{\alpha} (F(x^+_{k+1}) - F^*) \right), \] (3.11)
i.e., (3.7) holds for \( k+1 \) with \( R_{k+1}^2 \leq (1 - 1/(\sqrt{\kappa})) R_k^2 \). Note that \( c_{k+1} \) is the center of the minimum enclosing ball of the intersection of the two balls in (3.9) and (3.10), and can be computed in the same way as Algorithm 1 of [5]. From (3.11) we obtain that
\[ \|x^* - c_{k+1}\|^2 \leq (1 - 1/(\sqrt{\kappa})) R_k^2 \leq (1 - 1/(\sqrt{\kappa}))^k R_0^2. \]
Moreover, (3.9) indicates that
\[ F(x^+_{k+1}) - F^* \leq \frac{\alpha}{2} R_k^2 \leq \frac{\alpha}{2} (1 - 1/(\sqrt{\kappa})) R_0^2. \]

4 Practical Issues

4.1 GeoPG with backtracking

In practice, the Lipschitz constant \( \beta \) might be unknown to us. In this subsection, we describe a backtracking strategy for GeoPG in which \( \beta \) is not needed.

Note that the inequality (3.2) holds because of (3.3), which holds when \( t \in (0, 1/\beta) \). If \( \beta \) is unknown, we can perform backtracking on \( t \) such that (3.3) holds, which is a common practice for proximal gradient method (see, e.g., [2, 16, 15]). Note that the key step in our analysis of GeoPG is to guarantee that the two inequalities in (3.4) hold. According to Lemma 3.3, the second inequality in (3.4) holds as long as we use Algorithm 1 or Algorithm 2 to find \( x_k \), and it does not need the knowledge of \( \beta \). However, the first inequality in (3.4) requires \( t \leq 1/\beta \), because its proof in Lemma 3.3 needs (3.2). As a result, we need to perform backtracking on \( t \).
Theorem 4.1. Suppose sequence \( \{x_k, c_k, R_k, t_k\} \) is generated by Algorithm 4. For any \( k \geq 0 \), one has \( x^* \in B(c_k, R_k^2) \), and \( R_{k+1}^2 \leq (1 - \sqrt{\alpha t_k})^2 R_k^2 \), and thus
\[
\|x^* - c_k\|^2 \leq \sum_{i=0}^{k} (1 - \sqrt{\alpha t_i})^2 R_0^2 \leq (1 - \sqrt{\alpha t_{\min}})^k R_0^2.
\]

4.2 GeoPG with limited memory

The basic idea of GeoM is that in each iteration we maintain two balls \( B(y_1, r_1^2) \) and \( B(y_2, r_2^2) \) that both contain \( x^* \), and then compute the minimum enclosing ball of their intersection, which is expected to be smaller than both \( B(y_1, r_1^2) \) and \( B(y_2, r_2^2) \). One very intuitive idea that can possibly improve the performance of GeoM is to maintain more balls from the past, because their intersection should be smaller than the intersection of two balls. This idea has been
proposed by [1] and [7]. Specifically, [1] suggests to keep all the balls from past iterations, and then compute the minimum enclosing ball of their intersection. For a given bounded set \( Q \), the center of its minimum enclosing ball is known as the Chebyshev center, and is defined as the solution to the following problem:

\[
\min_{y} \max_{x \in Q} \|y - x\|^2 = \min_{y} \max_{x \in Q} \|y\|^2 - 2y^T x + \text{Tr}(xx^T). \tag{4.1}
\]

At this moment, let us assume that (4.1) can be solved for \( Q := \cap_{i=1}^m B(y_i, r_i^2) \). Now we can design a limited memory GeoPG algorithm (L-GeoPG). Specifically, in the \( k \)-th iteration of L-GeoPG, \( c_k \) is computed as the Chebyshev center of the intersection of the following balls:

\[
B(x_{k-m+1}^{++}, r_{k-m+1}^2), B(x_{k-m+2}^{++}, r_{k-m+2}^2), \ldots, B(x_k^{++}, r_k^2), B(c_{k-1}, R_{k-1}^2). \]

L-GeoPG is described in Algorithm 5.

**Algorithm 5** L-GeoPG: Limited-memory GeoPG

**Require**: Parameters \( \alpha, \beta \), memory size \( m > 0 \) and initial point \( x_0 \).

1. Set \( c_0 = x_0^{++}, r_0^2 = \frac{\|G_0(x_0)\|^2}{\alpha^2} (1 - 1/\kappa) \), and \( t = 1/\beta \).
2. **for** \( k = 1, 2, \ldots \) **do**
3. Use Algorithm 1 or 2 to find \( x_k \).
4. Compute \( r_k^2 = \frac{\|G_k(x_k)\|^2}{\alpha^2} (1 - 1/\kappa) \)
5. Compute \( B(c_k, R_k^2) \): an enclosing ball of the intersection of \( B(c_{k-1}, R_{k-1}^2) \) and \( Q_k := \cap_{i=k-m+1}^k B(x_i^{++}, r_i^2) \) (if \( k \leq m \), then set \( Q_k := \cap_{i=1}^k B(x_i^{++}, r_i^2) \)), by solving (4.1)
6. **end for**

**Remark 4.2**. The backtracking technique can also be incorporated in L-GeoPG. We denote the resulting algorithm as L-GeoPG-B.

L-GeoPG has the same linear convergence rate as GeoPG, as we show in Theorem 4.3.

**Theorem 4.3**. Consider L-GeoPG algorithm. For any \( k \geq 0 \), one has \( x^* \in B(c_k, R_k^2) \), and \( R_k^2 \leq (1 - 1/\sqrt{\kappa}) R_{k-1}^2 \), and thus

\[
\|x^* - c_k\|^2 \leq (1 - 1/\sqrt{\kappa})^k R_0^2.
\]

**Proof**. Note that \( Q_k := \cap_{i=k-m+1}^k B(x_i^{++}, r_i^2) \subset B(x_k^{++}, r_k^2) \). Thus, the minimum enclosing ball of \( B(c_{k-1}, R_{k-1}^2) \cap B(x_k^{++}, r_k^2) \) is an enclosing ball of \( B(c_{k-1}, R_{k-1}^2) \cap Q_k \). The proof then follows from the proof of Theorem 3.6 and we omit it for brevity.

Now we come back to the issue of computing Chebyshev center. Apparently, computing the minimum enclosing ball of \( Q_k \) is not easy in general, and it is suggested in [4] to compute the volumetric center [18] of \( Q_k \) as the center of the enclosing ball. In this paper, we propose to compute the relaxed Chebyshev center (RCC) instead. RCC was proposed by Eldar, Beck and Teboulle in [8] and is defined as the solution to the following problem

\[
\min_{y} \max_{(x, \triangle) \in \Gamma} \|y\|^2 - 2y^T x + \text{Tr}(\triangle)
\]

\[
= \max_{(x, \triangle) \in \Gamma} \min_{y} \|y\|^2 - 2y^T x + \text{Tr}(\triangle)
\]

\[
= \max_{(x, \triangle) \in \Gamma} -\|x\|^2 + \text{Tr}(\triangle), \tag{4.2}
\]
where $\Gamma = \{(x, \Delta) : x \in Q, \Delta \succeq xx^\top\}$. Note that (4.2) is a relaxation to (4.1) and more importantly, it is convex. If $Q = \cap_{i=1}^m B(c_i, r_i^2)$, then the dual problem of (4.2) is:

$$
\min \|C\lambda\|_2^2 - \sum_{i=1}^m \lambda_i \|c_i\|_2^2 + \sum_{i=1}^m \lambda_i r_i^2
$$

s.t. $\sum_{i=1}^m \lambda_i \geq 1$, $\lambda_i \geq 0$, $i = 1, \ldots, m,$

(4.3)

where $C := [c_1, \ldots, c_m]$. Note that (4.3) minimizes a quadratic function over a simplex, and is not difficult to solve. It is known that the optimal solutions of (4.2) and (4.3) are linked by $x^* = C\lambda^*$. In our experiment, we compute the RCC in Step 5 of Algorithm 5 instead of solving (4.1).

5 Numerical experiment

In this section, we compare our GeoPG algorithm with Nesterov’s accelerated proximal gradient (APG) method for solving two nonsmooth problems: linear regression and logistic regression, both with an elastic net regularization. Because of the elastic net term, strong convexity parameter $\alpha$ is known. However, we assume that $\beta$ is unknown and we implement backtracking for both GeoPG and APG, i.e., we test GeoPG-B and APG-B (APG with backtracking). We do not target to compare with other efficient algorithms for solving these two problems. Our main purpose here is to illustrate the performance of this new first-order method GeoPG. Further improvement of this algorithm and comparison with other state-of-the-art methods will be a future research topic.

The initial points were set to zero. GeoPG-B was terminated if $\|G_t(x_k^+)\|_\infty \leq \text{tol}$ and APG-B was terminated if $\|G_t(x_k)\|_\infty \leq \text{tol}$, where $\text{tol}$ is the accuracy tolerance. The parameters used in backtracking were set to $\eta = 0.5$ and $\gamma = 0.9$. In GeoPG-B, we used Algorithm 2 to find $x_k$, because we found that the performance of Algorithm 2 is slightly better than Algorithm 1 in practice. The codes were written in Matlab and run on a standard PC with 3.20 GHz I5 Intel microprocessor and 16GB of memory.

5.1 Linear regression with elastic net regularization

In this subsection, we compare GeoPG-B and APG-B for solving linear regression with elastic net regularization, a problem from machine learning and statistics [20]:

$$
\min_{x \in \mathbb{R}^n} \frac{1}{2p} \|Ax - b\|^2 + \frac{\alpha}{2} \|x\|^2 + \mu \|x\|_1,
$$

(5.1)

where $A \in \mathbb{R}^{p \times n}$, $b \in \mathbb{R}^p$, $\alpha, \mu > 0$ are weighting parameters.

We first compare these two algorithms on some synthetic data. In our experiments, entries of $A$ were drawn randomly from the standard Gaussian distribution, the solution $\bar{x}$ was a sparse vector with 10% nonzero entries whose locations are uniformly random and whose values follow the Gaussian distribution $3 \ast \mathcal{N}(0, 1)$, and $b = A \ast \bar{x} + n$, where the noise $n$ follows the Gaussian distribution $0.01 \ast \mathcal{N}(0, 1)$. Moreover, since we assume that the strong convexity parameter of (5.1) is equal to $\alpha$, when $p > n$, we manipulate $A$ such that the smallest eigenvalue of $A^\top A$ is equal to 0. Specifically, when $p > n$, we truncate the smallest eigenvalue of $A^\top A$ to 0, and obtain the new $A$ by eigenvalue decomposition of $A^\top A$. We set $\text{tol} = 10^{-8}$.

In Tables 1, 2 and 3 we report the comparison results of GeoPG-B and APG-B for solving different instances of (5.1). We use “f-ev”, “g-ev”, “p-ev” and “MVM” to denote the number of evaluations of objective function, gradient, proximal mapping of $\ell_1$ norm, and matrix-vector multiplications, respectively. The CPU times are in seconds. We use “-” to denote that the
algorithm does not converge in $10^5$ iterations. We tested different values of $\alpha$, which reflect different condition numbers of the problem. We also tested different values of $\mu$, which was set to $\mu = (10^{-3}, 10^{-4}, 10^{-5})/p \times \|A^T b\|_\infty$, respectively. “f-diff” denotes the absolute difference of the objective values returned by the two algorithms.

From Tables 1, 2 and 3 we see that GeoPG-B is more efficient than APG-B in terms of CPU time when $\alpha$ is small. For example, Table 1 indicates that GeoPG-B is faster than APG-B when $\alpha \leq 10^{-4}$, Table 2 indicates that GeoPG-B is faster than APG-B when $\alpha \leq 10^{-6}$, and Table 3 shows that GeoPG-B is faster than APG-B when $\alpha \leq 10^{-8}$. Since a small $\alpha$ corresponds to a large condition number, we can conclude that in this case GeoPG-B is more preferable than APG-B for ill-conditioned problems. Note that “f-diff” is very small in all cases, which indicates that the solutions returned by GeoPG-B and APG-B are very close.

We also conducted tests on three real datasets downloaded from the SVMLIB repository: a9a, RCV1 and Gisette, among which a9a and RCV1 are sparse and Gisette is dense. The size and sparsity (percentage of nonzero entries) of these three datasets are (32561 × 123, 11.28%), (20242 × 47236, 0.16%) and (6000 × 5000, 99.1%), respectively. The results are reported in Tables 4, 5 and 6, where $\alpha \leq 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}$ and $\mu = 10^{-3}, 10^{-4}, 10^{-5}$. We see from these tables that GeoPG-B is faster than APG-B on these real datasets when $\alpha$ is small, i.e., when the problem is more ill-conditioned.

5.2 Logistic regression with elastic net regularization

In this subsection, we compare the performance of GeoPG-B and APG-B for solving the following logistic regression problem with elastic net regularization.

$$
\min_{x \in \mathbb{R}^n} \frac{1}{p} \sum_{i=1}^{p} \log(1 + \exp(-b_i \cdot a_i^T x)) + \frac{\alpha}{2} \|x\|^2 + \mu\|x\|_1,
$$

(5.2)

where $a_i \in \mathbb{R}^n$ and $b_i \in \{\pm 1\}$ are the feature vector and class label of the $i$-th sample, respectively, and $\alpha, \mu > 0$ are weighting parameters.

We first compare GeoPG-B and APG-B for solving (5.2) on some synthetic data. In our experiments, each $a_i$ was drawn randomly from the standard Gaussian distribution, the linear model parameter $\bar{x}$ was a sparse vector with 10% nonzero entries whose locations are uniformly random and whose values follow the Gaussian distribution $3 * \mathcal{N}(0, 1)$, and $\ell = A * \bar{x} + n$, where noise $n$ follows the Gaussian distribution $0.01 * \mathcal{N}(0, 1)$. Then, we generate class labels as bernoulli random variables with the parameter $1/(1 + \exp(\ell_i))$. We set $tol = 10^{-8}$.

In Tables 7, 8 and 9 we report the comparison results of GeoPG-B and APG-B for solving different instances of (5.2). From results in these tables we again observe that GeoPG-B is faster than APG-B when $\alpha$ is small, i.e., when the condition number is large.

We also tested GeoPG-B and APG-B for solving (5.2) on the three real datasets a9a, RCV1 and Gisette from SVMLIB, and the results are reported in Tables 10, 11 and 12. We again have the similar observations as before, i.e., GeoPG-B is faster than APG-B for more ill-conditioned problems. Moreover, people usually set $\alpha$ as $10^{-8}$ or even smaller values in practice. Therefore, our GeoPG-B shows great potential when it is applied to solving real problems in practice.

5.3 Numerical results of L-GeoPG-B

In this subsection, we test the GeoPG with limited memory described in Algorithm 5 on solving (5.2) on Gisette dataset. Since we still need to use backtracking technique, we actually tested L-GeoPG-B. The results for different memory size $m$ are reported in Table 13. Note that $m = 0$ corresponds to the original GeoPG-B without memory. The subproblem (13) is solved using the function “quadprog” in Matlab.

From Table 13 we see that roughly speaking, L-GeoPG-B performs better for larger memory size, and in most cases, the performance of L-GeoPG-B with $m = 100$ is the best among
the reported results. This indicates that the limited-memory idea indeed helps improve the performance of GeoPG.

6 Conclusion

In this paper, we proposed a GeoPG algorithm for solving nonsmooth composite convex problems, which is an extension of the recent algorithm GeoM for solving smooth problems. We proved that GeoPG has the same optimal rate as Nesterov’s accelerated gradient method for solving strongly convex problem. Backtracking technique was adopted to handle the case when the Lipschitz constant is unknown. Limited-memory GeoPG was also considered to improve the practical performance of GeoPG. Numerical results on linear regression and logistic regression with elastic net regularization demonstrated the efficiency of GeoPG. It will be interesting to see how to extend GeoM and GeoPG to solving non-strongly convex problems, and how to further speed up the practical performance of GeoPG, and we leave these questions in a future work.

Acknowledgement

The authors are grateful to Yin Tat Lee for insightful comments and discussions.

| $\alpha$ | iter | cpu | f-ev | g-ev | p-ev | MVM | iter | cpu | f-ev | g-ev | p-ev | MVM | f-diff |
|---------|------|-----|------|------|------|-----|------|-----|------|------|------|-----|-------|
|        |      |     |      |      |      |     |      |     |      |      |      |     |       |
| $10^{-2}$ | 172  | 1.0 | 354  | 326  | 194  | 384 | 156  | 1.1 | 457  | 348  | 352  | 398 | 8.5e-14 |
| $10^{-4}$ | 538  | 2.8 | 1116 | 1020 | 611  | 1203| 95   | 0.7 | 267  | 240  | 245  | 247 | 6.4e-14 |
| $10^{-6}$ | 905  | 4.9 | 1868 | 1715 | 1029 | 2030| 94   | 0.7 | 260  | 249  | 254  | 247 | 5.0e-14 |
| $10^{-8}$ | 1040 | 5.4 | 2146 | 2003 | 1182 | 2332| 95   | 0.7 | 263  | 258  | 263  | 247 | 1.4e-14 |
| $10^{-10}$| 964  | 5.0 | 2002 | 1805 | 1095 | 2154| 95   | 0.7 | 263  | 267  | 272  | 247 | 2.1e-14 |
| $10^{-2}$ | 175  | 0.9 | 356  | 332  | 197  | 392 | 168  | 1.2 | 493  | 384  | 388  | 432 | 1.3e-13 |
| $10^{-4}$ | 687  | 3.6 | 1414 | 1304 | 779  | 1539| 145  | 1.0 | 411  | 392  | 397  | 377 | 1.5e-14 |
| $10^{-6}$ | 999  | 5.1 | 2086 | 1676 | 1134 | 2225| 140  | 1.0 | 371  | 384  | 394  | 354 | 6.5e-14 |
| $10^{-8}$ | 1122 | 5.8 | 2348 | 1827 | 1275 | 2499| 143  | 1.0 | 374  | 420  | 429  | 365 | 1.8e-15 |
| $10^{-10}$| 1142 | 5.9 | 2388 | 1858 | 1298 | 2545| 143  | 1.0 | 374  | 449  | 458  | 365 | 6.2e-15 |
| $10^{-2}$ | 168  | 0.9 | 346  | 314  | 189  | 374 | 113  | 0.8 | 328  | 252  | 256  | 296 | 1.4e-14 |
| $10^{-4}$ | 911  | 4.8 | 1836 | 1853 | 1035 | 2064| 207  | 1.5 | 603  | 587  | 592  | 535 | 4.1e-14 |
| $10^{-6}$ | 2293 | 11.9| 4744 | 3936 | 2605 | 5132| 191  | 1.4 | 523  | 596  | 602  | 492 | 3.8e-14 |
| $10^{-8}$ | 3979 | 20.5| 8266 | 5923 | 4526 | 8899| 199  | 1.4 | 500  | 713  | 728  | 501 | 9.8e-14 |
| $10^{-10}$| 4503 | 23.3| 9364 | 6668 | 5123 | 10068| 185  | 1.3 | 456  | 624  | 639  | 465 | 5.9e-14 |

Table 1: GeoPG-B and APG-B for solving linear regression with elastic net regularization. $p = 4000, n = 2000$
Table 3: GeoPG-B and APG-B for solving linear regression with elastic net regularization. $p = 2000, n = 2000$

| $\alpha$ | iter | cpu | f-ev | g-ev | p-ev | MVM | iter | cpu | f-ev | g-ev | p-ev | MVM | f-diff |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mu = 1.50e - 02$ |
| $10^{-2}$ | 244 | 0.7 | 498 | 475 | 276 | 548 | 304 | 1.3 | 889 | 690 | 694 | 774 | 3.4e-13 |
| $10^{-4}$ | 1900 | 4.8 | 3690 | 3582 | 2046 | 4048 | 545 | 2.4 | 1569 | 1298 | 1308 | 1378 | 1.3e-12 |
| $10^{-6}$ | 9706 | 26.0 | 19722 | 20445 | 11040 | 21926 | 557 | 2.3 | 1598 | 1328 | 1339 | 1415 | 2.8e-12 |
| $10^{-8}$ | 20056 | 53.7 | 40528 | 43361 | 22817 | 45427 | 561 | 2.3 | 1614 | 1332 | 1344 | 1416 | 2.4e-12 |
| $10^{-10}$ | 29352 | 53.9 | 41426 | 44159 | 23298 | 46357 | 565 | 2.3 | 1626 | 1373 | 1385 | 1436 | 2.4e-12 |

| $\mu = 1.50e - 03$ |
| $10^{-2}$ | 241 | 0.6 | 496 | 463 | 273 | 540 | 280 | 1.2 | 813 | 634 | 638 | 716 | 1.4e-14 |
| $10^{-4}$ | 1926 | 5.1 | 3968 | 3708 | 2188 | 4319 | 1218 | 5.0 | 3560 | 2875 | 2802 | 3073 | 2.0e-11 |
| $10^{-6}$ | 12502 | 32.7 | 25658 | 24681 | 14222 | 28118 | 1297 | 5.3 | 3718 | 3065 | 3097 | 3262 | 1.1e-11 |
| $10^{-8}$ | 47139 | 124.3 | 95560 | 100584 | 53646 | 106652 | 1289 | 5.3 | 3686 | 3043 | 3074 | 3245 | 2.1e-11 |
| $10^{-10}$ | 72186 | 194.3 | 145934 | 156713 | 82157 | 163534 | 1297 | 5.2 | 3717 | 3098 | 3132 | 3262 | 2.5e-11 |

| $\mu = 1.50e - 04$ |
| $10^{-2}$ | 239 | 0.6 | 488 | 460 | 270 | 536 | 225 | 0.9 | 648 | 510 | 514 | 584 | 3.3e-13 |
| $10^{-4}$ | 1985 | 5.2 | 4048 | 3860 | 2257 | 4476 | 1713 | 6.9 | 5041 | 4040 | 4058 | 4322 | 7.0e-11 |
| $10^{-6}$ | 13824 | 35.7 | 28534 | 25354 | 15726 | 31010 | 2527 | 10.2 | 7225 | 6019 | 6082 | 6434 | 2.5e-11 |
| $10^{-8}$ | 56339 | 146.2 | 116280 | 106460 | 64105 | 126410 | 2594 | 10.6 | 7288 | 6095 | 6182 | 6491 | 3.6e-11 |
| $10^{-10}$ | – | – | – | – | – | – | – | – | – | – | – | – |

Table 2: GeoPG-B and APG-B for solving linear regression with elastic net regularization. $p = 2000, n = 2000$

| $\alpha$ | iter | cpu | f-ev | g-ev | p-ev | MVM | iter | cpu | f-ev | g-ev | p-ev | MVM | f-diff |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mu = 1.82e - 02$ |
| $10^{-2}$ | 327 | 1.9 | 660 | 680 | 371 | 740 | 387 | 2.8 | 1117 | 936 | 946 | 980 | 2.0e-13 |
| $10^{-4}$ | 2263 | 12.8 | 4620 | 4445 | 2571 | 5096 | 2454 | 17.9 | 6858 | 6181 | 6225 | 6168 | 4.3e-11 |
| $10^{-6}$ | 12579 | 67.5 | 25566 | 26229 | 14312 | 28421 | 4478 | 32.7 | 12494 | 11180 | 11216 | 11300 | 1.8e-11 |
| $10^{-8}$ | 55577 | 299.3 | 121120 | 121939 | 63268 | 126044 | 4595 | 33.7 | 12814 | 11754 | 11795 | 11699 | 1.4e-10 |
| $10^{-10}$ | – | – | – | – | – | – | – | – | – | – | – | – |

| $\mu = 1.82e - 03$ |
| $10^{-2}$ | 306 | 1.7 | 622 | 621 | 346 | 688 | 279 | 2.1 | 813 | 677 | 684 | 713 | 6.4e-13 |
| $10^{-4}$ | 2355 | 12.7 | 4820 | 4534 | 2675 | 5296 | 2634 | 19.3 | 7482 | 6774 | 6846 | 6596 | 3.9e-13 |
| $10^{-6}$ | 14827 | 79.8 | 30328 | 28671 | 16862 | 33388 | 12756 | 94.1 | 36510 | 32580 | 32735 | 32121 | 2.2e-10 |
| $10^{-8}$ | 56286 | 305.7 | 114576 | 115199 | 64050 | 127099 | 11665 | 88.0 | 3397 | 32580 | 31987 | 29352 | 6.1e-11 |
| $10^{-10}$ | – | – | – | – | – | – | – | – | – | – | – | – |

| $\mu = 1.82e - 04$ |
| $10^{-2}$ | 283 | 1.5 | 576 | 560 | 320 | 636 | 219 | 1.6 | 643 | 523 | 528 | 561 | 4.7e-13 |
| $10^{-4}$ | 2420 | 13.2 | 4864 | 5242 | 2749 | 5487 | 2339 | 17.2 | 6818 | 6467 | 6509 | 5882 | 5.8e-11 |
| $10^{-6}$ | 16882 | 91.4 | 34412 | 31337 | 19186 | 38049 | 14803 | 109.3 | 41943 | 44052 | 44384 | 37152 | 4.9e-10 |
| $10^{-8}$ | 79903 | 430.5 | 163098 | 146951 | 90639 | 179423 | 41331 | 305.8 | 116983 | 113444 | 113952 | 104206 | 1.6e-10 |
| $10^{-10}$ | – | – | – | – | – | – | – | – | – | – | – | – |

Table 3: GeoPG-B and APG-B for solving linear regression with elastic net regularization. $p = 2000, n = 4000$
| λ | APG-B | GeoPG-B | λ | APG-B | GeoPG-B |
|---|---|---|---|---|---|
| 10^-2 | 266 | 0.3 | 340 | 530 | 301 | 599 | 260 | 0.6 | 769 | 602 | 608 | 662 | 1.3E-14 | 10^-4 | 1758 | 1.7 | 3985 | 3683 | 1998 | 3974 | 463 | 1.1 | 1374 | 1138 | 1144 | 1196 | 1.2E-14 | 10^-6 | 10970 | 10.4 | 21654 | 23858 | 12277 | 24518 | 410 | 0.9 | 1216 | 964 | 970 | 1058 | 1.5E-13 | 10^-8 | 23279 | 22.2 | 46646 | 52163 | 26493 | 52943 | 412 | 0.9 | 1222 | 976 | 982 | 1060 | 1.9E-13 | 10^-10 | 26057 | 24.9 | 52236 | 58464 | 29660 | 59260 | 431 | 0.9 | 1279 | 1063 | 1069 | 1104 | 2.2E-13 |

Table 4: GeoPG-B and APG-B for solving linear regression with elastic net regularization on dataset a9a

| λ = 1e - 04 | APG-B | GeoPG-B | λ = 1e - 05 | APG-B | GeoPG-B |
|---|---|---|---|---|---|
| 10^-2 | 267 | 0.3 | 544 | 526 | 302 | 600 | 249 | 0.5 | 734 | 571 | 577 | 642 | 6.7E-16 | 10^-4 | 1948 | 1.9 | 4088 | 4319 | 2315 | 4622 | 1701 | 3.7 | 5909 | 4273 | 4312 | 3.7E-12 | 10^-6 | 14954 | 14.3 | 30012 | 33215 | 17018 | 33985 | 4801 | 10.4 | 14388 | 11381 | 11386 | 12223 | 1.4E-12 | 10^-8 | 63920 | 60.9 | 127954 | 144494 | 72741 | 145426 | 910 | 2.0 | 2715 | 2629 | 2634 | 2347 | 3.7E-12 | 10^-10 | 94861 | 90.6 | 189814 | 214931 | 107970 | 215895 | 910 | 2.0 | 2715 | 2441 | 2446 | 2333 | 7.0E-13 |

| λ = 1e - 05 | APG-B | GeoPG-B | λ = 1e - 06 | APG-B | GeoPG-B |
|---|---|---|---|---|---|
| 10^-2 | 258 | 0.3 | 518 | 507 | 292 | 600 | 235 | 0.5 | 692 | 596 | 602 | 604 | 1.2E-14 | 10^-4 | 2035 | 1.9 | 4088 | 4319 | 2315 | 4622 | 1701 | 3.7 | 5909 | 4273 | 4312 | 3.7E-12 | 10^-6 | 16353 | 15.6 | 32768 | 36396 | 18609 | 37188 | 5773 | 12.5 | 13706 | 14961 | 14967 | 14808 | 4.5E-13 | 10^-8 | 85246 | 81.4 | 170570 | 193007 | 97062 | 194086 | 910 | 2.0 | 2715 | 2629 | 2634 | 2347 | 3.7E-12 | 10^-10 | 26057 | 24.9 | 52236 | 58464 | 29660 | 59260 | 431 | 0.9 | 1279 | 1063 | 1069 | 1104 | 2.2E-13 |

Table 5: GeoPG-B and APG-B for solving linear regression with elastic net regularization on dataset RCV1
Table 6: GeoPG-B and APG-B for solving linear regression with elastic net regularization on data set Gisette. Note that neither APG-B nor GeoPG-B converges in $10^5$ iterates when $\mu = 1e-05$ and $\alpha = 10^{-6}, 10^{-8}, 10^{-10}$.

| $\alpha$ | iter | cpu | f-ev | g-ev | p-ev | MVM | iter | cpu | f-ev | g-ev | p-ev | MVM | f-diff |
|----------|------|-----|------|------|------|-----|------|-----|------|------|------|-----|-------|
| $10^{-2}$ | 4026 | 198.1 | 8144 | 7229 | 4583 | 9121 | 4253 | 239.3 | 12593 | 10474 | 10506 | 10758 | 4.8e-14 |
| $10^{-4}$ | 30537 | 1504.2 | 61478 | 61380 | 34786 | 69371 | 6030 | 342.4 | 17939 | 17977 | 18006 | 15411 | 1.6e-13 |
| $10^{-6}$ | – | – | – | – | – | – | 5197 | 294.0 | 15419 | 16126 | 16159 | 13241 | – |
| $10^{-8}$ | – | – | – | – | – | – | 5692 | 322.8 | 16050 | 18851 | 18881 | 14506 | – |
| $10^{-10}$ | – | – | – | – | – | – | 6150 | 353.5 | 18295 | 23420 | 23450 | 15714 | – |

$\mu = 1e-04$

| $\mu = 1e-05$ |
|----------|------|-----|------|------|------|-----|------|-----|------|------|------|-----|-------|

Table 7: GeoPG-B and APG-B for solving logistic regression with elastic net regularization. $p = 6000, n = 3000$

| $\alpha$ | iter | cpu | f-ev | g-ev | p-ev | MVM | iter | cpu | f-ev | g-ev | p-ev | MVM | f-diff |
|----------|------|-----|------|------|------|-----|------|-----|------|------|------|-----|-------|
| $10^{-2}$ | 55 | 0.9 | 112 | 96 | 60 | 158 | 46 | 1.3 | 125 | 145 | 146 | 207 | 1.1e-13 |
| $10^{-4}$ | 256 | 4.3 | 536 | 470 | 289 | 761 | 55 | 1.7 | 144 | 194 | 194 | 269 | 5.6e-13 |
| $10^{-6}$ | 509 | 8.7 | 1048 | 972 | 577 | 1551 | 61 | 2.0 | 164 | 218 | 220 | 300 | 1.3e-12 |
| $10^{-8}$ | 573 | 9.5 | 1188 | 1086 | 649 | 1737 | 60 | 1.9 | 161 | 223 | 225 | 305 | 1.4e-12 |
| $10^{-10}$ | 585 | 9.6 | 1208 | 1112 | 663 | 1777 | 59 | 2.1 | 158 | 231 | 233 | 313 | 1.4e-12 |

$\mu = 1.00e-04$

| $\mu = 1.00e-05$ |
|----------|------|-----|------|------|------|-----|------|-----|------|------|------|-----|-------|

| $\mu = 1.00e-06$ |
Table 8: GeoPG-B and APG-B for solving logistic regression with elastic net regularization. 
\( p = 3000, n = 6000 \)

| \( \alpha \) | iter | cpu | f-ev | g-ev | p-ev | MVM | iter | cpu | f-ev | g-ev | p-ev | MVM | f-diff |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10^{-5} | 58 | 0.9 | 114 | 107 | 63 | 172 | 60 | 1.6 | 169 | 200 | 196 | 279 | 5.1e-14 |
| 10^{-4} | 253 | 4.1 | 516 | 466 | 284 | 752 | 110 | 3.5 | 292 | 420 | 412 | 562 | 1.9e-12 |
| 10^{-3} | 893 | 15.1 | 1824 | 1757 | 1012 | 2771 | 115 | 4.8 | 305 | 464 | 463 | 615 | 4.1e-12 |
| 10^{-2} | 1265 | 21.9 | 2584 | 2543 | 1435 | 3980 | 114 | 4.8 | 302 | 504 | 501 | 649 | 4.9e-12 |
| 10^{-1} | 1333 | 22.6 | 2712 | 2691 | 1513 | 4206 | 114 | 4.8 | 302 | 543 | 540 | 688 | 5.0e-12 |

| \( \mu = 1.00e - 04 \) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10^{-5} | 56 | 0.8 | 112 | 89 | 60 | 151 | 42 | 1.1 | 116 | 133 | 132 | 188 | 1.4e-13 |
| 10^{-4} | 159 | 2.2 | 328 | 237 | 174 | 413 | 128 | 3.7 | 340 | 455 | 447 | 616 | 1.7e-11 |
| 10^{-3} | 750 | 11.3 | 1560 | 1238 | 845 | 2085 | 157 | 5.2 | 392 | 621 | 614 | 817 | 5.3e-11 |
| 10^{-2} | 1927 | 30.3 | 4012 | 3447 | 2182 | 5631 | 158 | 5.8 | 410 | 679 | 674 | 877 | 8.6e-11 |
| 10^{-1} | 2364 | 37.5 | 4934 | 4290 | 2677 | 6969 | 164 | 6.6 | 427 | 760 | 753 | 965 | 1.5e-10 |

| \( \mu = 1.00e - 05 \) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10^{-5} | 54 | 0.8 | 108 | 85 | 58 | 145 | 42 | 1.1 | 110 | 136 | 134 | 191 | 2.9e-13 |
| 10^{-4} | 118 | 1.6 | 236 | 177 | 126 | 305 | 81 | 2.1 | 207 | 266 | 263 | 365 | 1.4e-11 |
| 10^{-3} | 493 | 6.4 | 1062 | 636 | 551 | 1189 | 153 | 4.9 | 365 | 588 | 580 | 776 | 2.9e-10 |
| 10^{-2} | 3492 | 45.0 | 7742 | 4365 | 3949 | 8316 | 163 | 5.8 | 379 | 686 | 677 | 886 | 8.3e-10 |
| 10^{-1} | 7655 | 98.4 | 17058 | 9498 | 8666 | 18166 | 169 | 6.8 | 403 | 782 | 775 | 990 | 1.7e-09 |

Table 9: GeoPG-B and APG-B for solving logistic regression with elastic net regularization. 
\( p = 3000, n = 3000 \)

| \( \alpha \) | iter | cpu | f-ev | g-ev | p-ev | MVM | iter | cpu | f-ev | g-ev | p-ev | MVM | f-diff |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10^{-5} | 55 | 0.5 | 110 | 99 | 60 | 161 | 53 | 0.8 | 144 | 172 | 171 | 243 | 2.7e-13 |
| 10^{-4} | 278 | 2.4 | 566 | 512 | 312 | 826 | 90 | 1.4 | 237 | 325 | 322 | 442 | 2.7e-12 |
| 10^{-3} | 845 | 7.1 | 1732 | 1637 | 957 | 2396 | 89 | 1.5 | 234 | 336 | 334 | 452 | 2.6e-12 |
| 10^{-2} | 1158 | 9.7 | 2378 | 2283 | 1314 | 3599 | 89 | 1.6 | 234 | 361 | 359 | 477 | 2.6e-12 |
| 10^{-1} | 1186 | 9.9 | 2444 | 2340 | 1345 | 3687 | 88 | 1.7 | 231 | 377 | 375 | 492 | 2.8e-12 |

| \( \mu = 1.00e - 04 \) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10^{-5} | 55 | 0.4 | 108 | 89 | 60 | 151 | 53 | 0.7 | 144 | 172 | 169 | 242 | 3.5e-13 |
| 10^{-4} | 172 | 1.3 | 352 | 273 | 191 | 466 | 122 | 1.8 | 327 | 424 | 415 | 579 | 3.2e-11 |
| 10^{-3} | 868 | 6.6 | 1834 | 1435 | 980 | 2437 | 145 | 2.3 | 374 | 529 | 523 | 714 | 6.8e-11 |
| 10^{-2} | 1985 | 16.0 | 4168 | 3527 | 2248 | 5777 | 144 | 2.5 | 372 | 565 | 563 | 747 | 5.9e-11 |
| 10^{-1} | 2475 | 19.9 | 5160 | 4545 | 2807 | 7354 | 143 | 2.7 | 365 | 607 | 605 | 787 | 7.4e-11 |

| \( \mu = 1.00e - 05 \) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 10^{-5} | 55 | 0.4 | 108 | 91 | 59 | 152 | 48 | 0.7 | 129 | 158 | 155 | 224 | 6.7e-13 |
| 10^{-4} | 126 | 0.9 | 256 | 185 | 137 | 324 | 126 | 0.9 | 256 | 185 | 137 | 324 | 2.0e-12 |
| 10^{-3} | 515 | 3.4 | 1108 | 680 | 576 | 1258 | 146 | 2.2 | 344 | 524 | 517 | 705 | 4.6e-10 |
| 10^{-2} | 3196 | 21.0 | 7054 | 4118 | 3615 | 7735 | 154 | 2.5 | 372 | 587 | 586 | 778 | 8.9e-10 |
| 10^{-1} | 6434 | 42.7 | 14228 | 8384 | 7284 | 15670 | 152 | 2.8 | 370 | 630 | 629 | 820 | 5.4e-10 |
### Table 10: GeoPG-B and APG-B for solving logistic regression with elastic net on dataset a9a

| α   | iter | cpu | f-ev | g-ev | p-ev | MVM       | iter | cpu | f-ev | g-ev | p-ev | MVM       | f-diff |
|-----|------|-----|------|------|------|-----------|------|-----|------|------|------|-----------|--------|
| 10^{-2} | 96   | 0.2 | 194  | 174  | 106  | 282       | 96   | 0.2 | 194  | 174  | 106  | 282       | 1.1e-14 |
| 10^{-4} | 709  | 1.7 | 1440 | 1388 | 805  | 2195      | 756  | 3.8 | 2254 | 2669 | 2577 | 3615      | 8.2e-13 |
| 10^{-6} | 5195 | 13.6| 10488| 10973| 5912 | 16887     | 2581 | 13.4| 7725 | 8995 | 8770 | 12112     | 4.4e-11 |
| 10^{-8} | 25300| 64.8| 50772| 56141| 28793| 84936     | 716  | 3.7 | 2130 | 2529 | 2583 | 3427      | 9.9e-10 |
| 10^{-10}| 42633|109.4|85446 |95447 |48519 |143968    |723   |3.8  |2151  |2584  |2640  |3497       |7.9e-11  |

### Table 11: GeoPG-B and APG-B for solving logistic regression with elastic net on dataset Rcv1

| α   | iter | cpu | f-ev | g-ev | p-ev | MVM       | iter | cpu | f-ev | g-ev | p-ev | MVM       | f-diff |
|-----|------|-----|------|------|------|-----------|------|-----|------|------|------|-----------|--------|
| 10^{-2} | 106  | 0.3 | 210  | 199  | 119  | 320       | 72   | 0.4 | 207  | 258  | 255  | 347       | 1.4e-14 |
| 10^{-4} | 770  | 1.9 | 1550 | 1526 | 874  | 2402      | 685  | 3.5 | 2038 | 2448 | 2367 | 3301      | 3.7e-12 |
| 10^{-6} | 5842 | 14.7| 11762|12453 |6648 |19084     |3026 |15.5|9061 |11099 |10715|14750      |2.1e-11  |
| 10^{-8} | 46819| 119.9|93782 |104946|5331 |158259    |7784 |38.8|23335 |26969 |26326 |36558     |1.9e-12  |
| 10^{-10} | –    | –   | –    | –    | –    | –         |1488 |8.2 |4447 |5674 |5721 |5767       |–       |
| \( \alpha \) | \( m = 0 \) | \( m = 5 \) | \( m = 10 \) | \( m = 20 \) | \( m = 50 \) | \( m = 100 \) |
|---|---|---|---|---|---|---|
| \( \mu = 10^{-2} \) | 819 | 82.5 | 1310 | 164.1 | 1015 | 125.8 |
| \( \mu = 10^{-4} \) | 2177 | 217.5 | 3656 | 470.8 | 3430 | 417.9 |
| \( \mu = 10^{-6} \) | 2013 | 230.9 | 1606 | 235.9 | 1589 | 221.5 |
| \( \mu = 10^{-8} \) | 1793 | 214.9 | 1622 | 252.7 | 1530 | 224.4 |
| \( \mu = 10^{-10} \) | 1808 | 227.1 | 1599 | 260.8 | 1549 | 245.3 |

Table 12: GeoPG-B and APG-B for solving logistic regression with elastic net on dataset Gisette

| \( \alpha \) | \( \mu = 1e-03 \) | \( \mu = 1e-04 \) | \( \mu = 1e-05 \) |
|---|---|---|---|
| \( \mu = 10^{-2} \) | 961 | 93.7 | 2573 | 312.6 |
| \( \mu = 10^{-4} \) | 2146 | 217.2 | 2237 | 312.3 |
| \( \mu = 10^{-6} \) | 2226 | 276.7 | 2057 | 329.4 |
| \( \mu = 10^{-8} \) | 2283 | 296.2 | 2046 | 361.7 |
| \( \mu = 10^{-10} \) | 2776 | 436.7 | 2503 | 432.4 |

Table 13: L-GeoPG-B for solving logistic regression with elastic net regularization on data set Gisette
References

[1] H. Attouch, Z. Chbani, J. Peypouquet, and P. Redont. Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity. *Mathematical Programming*, 2016.

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

[3] R. P. Brent. An algorithm with guaranteed convergence for finding a zero of a function. In *Algorithms for Minimization without Derivatives*. Englewood Cliffs, NJ: Prentice-Hall, 1973.

[4] S. Bubeck and Y.-T. Lee. Black-box optimization with a politician. *ICML*, 2016.

[5] S. Bubeck, Y.-T. Lee, and M. Singh. A geometric alternative to Nesterov’s accelerated gradient descent. *arXiv preprint arXiv:1506.08187*, 2015.

[6] T. J. Dekker. Finding a zero by means of successive linear interpolation. In *Constructive Aspects of the Fundamental Theorem of Algebra*. London: Wiley-Interscience, 1969.

[7] D. Drusvyatksiy, M. Fazel, and S. Roy. An optimal first order method based on optimal quadratic averaging. *SIAM Journal on Optimization*, 2016.

[8] Y. C. Eldar, A. Beck, and M. Teboulle. A minimax Chebyshev estimator for bounded error estimation. *IEEE Transactions on Signal Processing*, 56(4):1388–1397, 2008.

[9] M. Gerdts, S. Horn, and S. Kimmerle. Line search globalization of a semismooth Newton method for operator equations in Hilbert spaces with applications in optimal control. *Journal of Industrial And Management Optimization*, 13(1):47–62, 2017.

[10] E. Hans and T. Raasch. Global convergence of damped semismooth Newton methods for L1 Tikhonov regularization. *Inverse Problems*, 31(2):025005, 2015.

[11] J. D. Lee, Y. Sun, and M. A. Saunders. Proximal Newton-type methods for minimizing composite functions. *SIAM Journal on Optimization*, 24(3):1420–1443, 2014.

[12] L. Lessard, B. Recht, and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 26(1):57–95, 2016.

[13] Y. E. Nesterov. A method for unconstrained convex minimization problem with the rate of convergence $O(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547, 1983.

[14] Y. E. Nesterov. *Introductory lectures on convex optimization: A basic course*. Applied Optimization. Kluwer Academic Publishers, Boston, MA, 2004.

[15] Y. E. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[16] K. Scheinberg, D. Goldfarb, and X. Bai. Fast first-order methods for composite convex optimization with backtracking. *Foundations of Computational Mathematics*, 14(3):389–417, 2014.

[17] W. Su, S. Boyd, and E. J. Candès. A differential equation for modeling Nesterov’s accelerated gradient method: Theory and insights. In *NIPS*, 2014.

[18] P. M. Vaidya. A new algorithm for minimizing convex functions over convex sets. *Mathematical Programming*, 73(3):291–341, 1996.
[19] A. Wibisono, A. Wilson, and M. I. Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 133:E7351–E7358, 2016.

[20] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67(2):301–320, 2005.