An optimal first order method based on optimal quadratic averaging

Dmitriy Drusvyatskiy∗ Maryam Fazel† Scott Roy‡

Abstract

In a recent paper, Bubeck, Lee, and Singh introduced a new first order method for minimizing smooth strongly convex functions. Their geometric descent algorithm, largely inspired by the ellipsoid method, enjoys the optimal linear rate of convergence. We show that the same iterate sequence is generated by a scheme that in each iteration computes an optimal average of quadratic lower-models of the function. Indeed, the minimum of the averaged quadratic approaches the true minimum at an optimal rate. This intuitive viewpoint reveals clear connections to the original fast-gradient methods and cutting plane ideas, and leads to limited-memory extensions with improved performance.

1 Introduction

Consider a function $f : \mathbb{R}^n \to \mathbb{R}$ that is $\beta$-smooth and $\alpha$-strongly convex. Thus each point $x$ yields a quadratic upper estimator and a quadratic lower estimator of the function. Namely, inequalities $q(y; x) \leq f(y) \leq Q(y; x)$ hold for all $x, y \in \mathbb{R}^n$, where we set

$$q(y; x) := f(x) + \langle \nabla f(x), y - x \rangle + \frac{\alpha}{2} \|y - x\|^2,$$

$$Q(y; x) := f(x) + \langle \nabla f(x), y - x \rangle + \frac{\beta}{2} \|y - x\|^2.$$

Classically, one step of the steepest descent algorithm decreases the squared distance of the iterate to the minimizer of $f$ by the fraction $1 - \alpha/\beta$. This linear convergence rate is suboptimal from a computational complexity viewpoint. Optimal first-order methods, originating in Nesterov’s work [11] achieve the superior (and the best possible) linear rate $1 - \sqrt{\alpha/\beta}$; see also the discussion in [10, Section 2.2]. Such accelerated schemes, on the other hand, are notoriously difficult to analyze. Numerous recent papers (e.g. [1, 2, 5, 9, 13]) have aimed to shed new light on optimal algorithms.

This manuscript is motivated by the novel geometric descent algorithm of Bubeck, Lee, and Singh [5]. Their scheme is highly geometric, sharing some aspects with the ellipsoid method, and it achieves the optimal linear rate of convergence. Moreover, the geometric approach of Bubeck, Lee, and Singh is inspired by the ellipsoid method, and it enjoys the optimal linear rate of convergence. We show that the same iterate sequence is generated by a scheme that in each iteration computes an optimal average of quadratic lower-models of the function. Indeed, the minimum of the averaged quadratic approaches the true minimum at an optimal rate. This intuitive viewpoint reveals clear connections to the original fast-gradient methods and cutting plane ideas, and leads to limited-memory extensions with improved performance.

∗University of Washington, Department of Mathematics, Seattle, WA 98195; ddrusv@uw.edu. Research of Drusvyatskiy was partially supported by the AFOSR YIP award FA9550-15-1-0237.
†University of Washington, Department of Electrical Engineering, Seattle, WA 98195; mfazel@uw.edu. Research partially supported by ONR award N00014-12-1-1002 and NSF award CIF-1409836.
‡University of Washington, Department of Mathematics, Seattle, WA 98195; scottroy@uw.edu

arXiv:1604.06543v3 [math.OC] 28 Feb 2017
descent algorithm often has much better practical performance than accelerated gradient methods; see the discussion in [5]. Motivated by their work, in this paper we propose an intuitive method that maintains a quadratic lower model of the objective function, whose minimal value converges to the true minimum at an optimal linear rate. We will show that the two methods are indeed equivalent in the sense that they produce the same iterate sequence. The quadratic averaging viewpoint, however, has important advantages. First, it immediately yields a comparison with the original accelerated gradient method [10, 11] and cutting plane techniques. Secondly, quadratic averaging motivates a simple strategy for significantly accelerating the method in practice by utilizing accumulated information — a limited memory version of the scheme.

The outline of the paper is as follows. In Section 2, we describe the optimal quadratic averaging framework (Algorithm 1) — the focal point of the manuscript. In Section 3, we propose a limited memory version of Algorithm 1 based on iteratively solving small dimensional quadratic programs. In Section 4, we show that our Algorithm 1 and the geometric descent method of [5] produce the same iterate sequence. Section 5 is devoted to numerical illustrations, in particular showing that the optimal quadratic averaging algorithm with memory can be competitive with L-BFGS. We finish the paper with Section 6, where we discuss the challenges that must be overcome in order to derive proximal extensions. In the final stages of revising this paper, a new manuscript [7] appeared explaining how to overcome exactly these challenges.

1.1 Notation
We follow the notation of [5]. Given a point \( x \in \mathbb{R}^n \), we define a short step
\[
x^+ := x - \frac{1}{\beta} \nabla f(x)
\]
and a long step
\[
x^{++} := x - \frac{1}{\alpha} \nabla f(x).
\]
Setting \( y = x^+ \) in the quadratic bound \( f(y) \leq Q(y; x) \) yields the standard inequality
\[
f(x^+) + \frac{1}{2\beta} \|
abla f(x)\|^2 \leq f(x). \tag{1}
\]
We denote the unique minimizer of \( f \) by \( x^* \), its minimal value by \( f^* \), and its condition number by \( \kappa := \beta/\alpha \). Throughout, the symbol \( B(x, R^2) \) stands for the Euclidean ball of radius \( R \) around \( x \). For any points \( x, y \in \mathbb{R}^n \), we let line_search \( (x, y) \) be the minimizer of \( f \) on the line between \( x \) and \( y \).

2 Optimal quadratic averaging
The starting point for our development is the elementary observation that every point \( \bar{x} \) provides a quadratic under-estimator of the objective function, having a canonical form. Indeed, completing the square in the strong convexity inequality \( f(x) \geq q(x; \bar{x}) \) yields
\[
f(x) \geq \left( f(\bar{x}) - \frac{\|
abla f(\bar{x})\|^2}{2\alpha} \right) + \frac{\alpha}{2} \| x - \bar{x}^{++} \|^2. \tag{2}
\]
Suppose we have now available two quadratic lower-estimators:
\[ f(x) \geq Q_A(x) := v_A + \frac{\alpha}{2} \| x - x_A \|^2 \]
\[ f(x) \geq Q_B(x) := v_B + \frac{\alpha}{2} \| x - x_B \|^2. \]

Clearly, the minimal values of \( Q_A \) and of \( Q_B \) lower-bound the minimal value of \( f \). For any \( \lambda \in [0, 1] \), the average \( Q_\lambda := \lambda Q_A + (1 - \lambda) Q_B \) is again a quadratic lower-estimator of \( f \). Thus we are led to the question:

What choice of \( \lambda \) yields the tightest lower-bound on the minimal value of \( f \)?

To answer this question, observe the equality
\[ Q_{\lambda}(x) := \lambda Q_A(x) + (1 - \lambda) Q_B(x) = v_\lambda + \frac{\alpha}{2} \| x - c_\lambda \|^2, \]
where
\[ c_{\lambda} = \lambda x_A + (1 - \lambda) x_B \]
and
\[ v_\lambda = v_B + \left( v_A - v_B + \frac{\alpha}{2} \| x_A - x_B \|^2 \right) \lambda - \left( \frac{\alpha}{2} \| x_A - x_B \|^2 \right) \lambda^2. \]

In particular, the average \( Q_\lambda \) has the same canonical form as \( Q_A \) and \( Q_B \). A quick computation now shows that \( v_\lambda \) (the minimum of \( Q_\lambda \)) is maximized by setting
\[ \bar{\lambda} := \text{proj}_{[0,1]} \left( \frac{1}{2} + \frac{v_A - v_B}{\alpha \| x_A - x_B \|^2} \right). \]

With this choice of \( \lambda \), we call the quadratic function \( \bar{Q} = \bar{v} + \frac{\alpha}{2} \| \cdot - \bar{c} \|^2 \) the optimal averaging of \( Q_A \) and \( Q_B \). See Figure 1 for an illustration.

![Figure 1: The optimal averaging of \( Q_A(x) = 1 + 0.5(x + 2)^2 \) and \( Q_B(x) = 3 + 0.5(x - 4)^2 \).](image)

An algorithmic idea emerges. Given a current iterate \( x_k \), form the quadratic lower-model \( Q(\cdot) \) in (2) with \( \bar{x} = x_k \). Then let \( Q_k \) be the optimal averaging of \( Q \) and the quadratic lower

3
model $Q_{k-1}$ from the previous step. Finally define $x_{k+1}$ to be the minimizer of $Q_k$, and repeat. Though attractive, the scheme does not converge at an optimal rate. Indeed, this algorithm is closely related to the suboptimal method in [5]; see Section 4.1 for a discussion. The main idea behind acceleration, natural in retrospect, is a separation of roles: one must maintain two sequences of points $x_k$ and $c_k$. The points $x_k$ will generate quadratic lower models as above, while $c_k$ will be the minimizers of the quadratics. We summarize the proposed method in Algorithm [1]. The rule for determining the iterate $x_k$ by a line search is entirely motivated by the geometric descent method in [5].

**Algorithm 1: Optimal Quadratic Averaging**

| Input: | Starting point $x_0$ and strong convexity constant $\alpha > 0$. |
| Output: | Final quadratic $Q_K(x) = v_K + \frac{\alpha}{2} \| x - c_K \|^2$ and $x_K^+$. |
| Set $Q_0(x) = v_0 + \frac{\alpha}{2} \| x - c_0 \|^2$, where $v_0 = f(x_0) - \frac{\| \nabla f(x_0) \|^2}{2\alpha}$ and $c_0 = x_0^+$; |
| for $k = 1, \ldots, K$ do |
| Set $x_k = \text{line_search}(c_{k-1}, x_{k-1}^+);$ |
| Set $Q(x) = \left( f(x_k) - \frac{\| \nabla f(x_k) \|^2}{2\alpha} \right) + \frac{\alpha}{2} \| x - x_k^+ \|^2$; |
| Let $Q_k(x) = v_k + \frac{\alpha}{2} \| x - c_k \|^2$ be the optimal averaging of $Q$ and $Q_{k-1}$; |
| end |

**Remark 2.1.** When implementing Algorithm [1], we set $x_k^+ = \text{line_search}(x_k, x_k - \nabla f(x_k))$. This does not impact the analysis as $x_k^+$ still satisfies the key inequality [1]. With this modifi-

To aid in the analysis of the scheme, we record the following easy observation.

**Lemma 2.2.** Suppose that $\overline{Q} = \bar{v} + \frac{\alpha}{2} \| x - \bar{c} \|^2$ is the optimal averaging of the quadratics $Q_A = v_A + \frac{\alpha}{2} \| x - A \|^2$ and $Q_B = v_B + \frac{\alpha}{2} \| x - B \|^2$. Then the quantity $\bar{v}$ is nondecreasing in both $v_A$ and $v_B$. Moreover, whenever the inequality $|v_A - v_B| \leq \frac{\alpha}{2} \| x_A - x_B \|^2$ holds, we have

$$\bar{v} = \frac{\alpha}{8} \| x_A - x_B \|^2 + \frac{1}{2} (v_A + v_B) + \frac{1}{2\alpha} \left( \frac{v_A - v_B}{\| x_A - x_B \|} \right)^2.$$

**Proof.** Define $\hat{\lambda} := \frac{1}{2} + \frac{v_A - v_B}{\alpha \| x_A - x_B \|^2}$. Notice that we have

$$\hat{\lambda} \in [0, 1] \text{ if and only if } |v_A - v_B| \leq \frac{\alpha}{2} \| x_A - x_B \|^2.$$

If $\hat{\lambda}$ lies in $[0, 1]$, equality $\hat{\lambda} = \hat{\lambda}$ holds, and then from [5] we deduce

$$\bar{v} = v_{\lambda} = \frac{\alpha}{8} \| x_A - x_B \|^2 + \frac{1}{2} (v_A + v_B) + \frac{1}{2\alpha} \left( \frac{v_A - v_B}{\| x_A - x_B \|} \right)^2.$$

If $\hat{\lambda}$ does not lie in $[0, 1]$, then an easy argument shows that $\bar{v}$ is linear in $v_A$ either with slope one or zero. If $\hat{\lambda}$ lies in $(0, 1)$, then we compute

$$\frac{\partial \bar{v}}{\partial v_A} = \frac{1}{2} + \frac{1}{\alpha \| x_A - x_B \|^2} (v_A - v_B),$$
which is nonnegative because \( \frac{|x_A - x_B|}{\alpha_2 \|x_A - x_B\|} \leq \frac{1}{2} \). Since \( \tilde{v} \) is clearly continuous, it follows that \( \tilde{v} \) is nondecreasing in \( v_A \), and by symmetry also in \( v_B \).

We now show that Algorithm 1 achieves the optimal linear rate of convergence.

**Theorem 2.3** (Convergence of optimal quadratic averaging). In Algorithm 1 for every index \( k \geq 0 \), the inequalities \( v_k \leq f^* \leq f(x_k^+) \) hold and we have

\[
f(x_k^+) - v_k \leq \left( 1 - \frac{1}{\sqrt{\kappa}} \right)^k (f(x_0^+) - v_0).
\]

**Proof.** Since in each iteration, the algorithm only averages quadratic minors of \( f \), the inequalities \( v_k \leq f^* \leq f(x_k^+) \) hold for every index \( k \). Set \( r_0 = \frac{\alpha_2}{\alpha}(f(x_0^+) - v_0) \) and define the quantities \( r_k := (1 - \frac{1}{\sqrt{\kappa}})^k r_0 \). We will show by induction that the inequality \( v_k \geq f(x_k^+) - \frac{\alpha_2}{\kappa} r_k \) holds for all \( k \geq 0 \). The base case \( k = 0 \) is immediate, and so assume we have

\[
v_{k-1} \geq f(x_{k-1}^+) - \frac{\alpha_2}{2} r_{k-1}
\]

for some index \( k - 1 \). Next set \( v_A := f(x_k) - \frac{\|\nabla f(x_k)\|^2}{2\alpha} \) and \( v_B := v_{k-1} \). Then the function

\[
Q_k(x) = v_k + \frac{\alpha}{2} \|x - c_k\|^2,
\]

is the optimal averaging of \( Q_A(x) = v_A + \frac{\alpha}{2} \|x - x_k^+\|^2 \) and \( Q_B(x) = v_B + \frac{\alpha}{2} \|x - c_k-1\|^2 \). An application of (10) yields the lower bound \( \hat{v}_A \) on \( v_A \):

\[
v_A = f(x_k) - \frac{\|\nabla f(x_k)\|^2}{2\alpha} \geq f(x_k^+) - \frac{\alpha}{2} \frac{\|\nabla f(x_k)\|^2}{\alpha^2} \left( 1 - \frac{1}{\kappa} \right) := \hat{v}_A.
\]

The induction hypothesis and the choice of \( x_k \) yield a lower bound \( \hat{v}_B \) on \( v_B \):

\[
v_B \geq f(x_{k-1}^+) - \frac{\alpha}{2} r_{k-1} \geq f(x_k) - \frac{\alpha}{2} r_{k-1}
\]

\[
\geq f(x_k^+) + \frac{1}{2}\|\nabla f(x_k)\|^2 - \frac{\alpha}{2} r_{k-1}
\]

\[
= f(x_k^+) - \frac{\alpha}{2} \left( r_{k-1} - \frac{1}{\alpha^2 \kappa} \|\nabla f(x_k)\|^2 \right) := \hat{v}_B.
\]

Define the quantities \( d := \|x_k^+- c_{k-1}\| \) and \( h := \|\nabla f(x_k)\| \). We now split the proof into two cases. First assume \( h^2 \leq \frac{\|\nabla f(x_k)\|}{\kappa} \). Then we deduce

\[
v_k \geq v_A \geq \hat{v}_A = f(x_k^+) - \frac{\alpha}{2} h^2 \left( 1 - \frac{1}{\kappa} \right)
\]

\[
\geq f(x_k^+) - \frac{\alpha}{2} \frac{1}{\kappa} \left( 1 - \frac{1}{\kappa} \right)
\]

\[
\geq f(x_k^+) - \frac{\alpha}{2} r_{k-1} \left( 1 - \frac{1}{\sqrt{\kappa}} \right)
\]

\[
= f(x_k^+) - \frac{\alpha}{2} r_k,
\]
where the third line follows since $2/\sqrt{\kappa} \leq 1 + 1/\kappa$ holds. Hence in this case, the proof is complete.

Next suppose $h^2 > \frac{r_{k-1}}{2}$ and let $v + \frac{\alpha}{2} \|\cdot - c\|^2$ be the optimal average of the two quadratics \(\hat{v}_A + \frac{\alpha}{2} \|\cdot - x_k^+\|^2\) and \(\hat{v}_B + \frac{\alpha}{2} \|\cdot - c_{k-1}\|^2\). By Lemma 2.2, the inequality $v_k \geq v$ holds. We claim that equality

\[
v = \hat{v}_B + \frac{\alpha}{8} \left( \frac{d^2 + \frac{2}{\alpha}(\hat{v}_A - \hat{v}_B))^2}{d^2} \right) \quad \text{holds. (4)}
\]

This follows immediately from Lemma 2.2 once we show \(\frac{|\hat{v}_A - \hat{v}_B|}{\alpha d^2} \leq \frac{r_{k-1} - h^2}{2d^2}\). To this end, note first the equality

\[
d^2 - h^2 = \|x_k - c_{k-1}\|^2 - \frac{2}{\alpha} \langle \nabla f(x_k), x_k - c_{k-1} \rangle = \|x_k - c_{k-1}\|^2 \geq 0.
\]

Thus we have $h^2 - r_{k-1} < h^2 \leq d^2$. Finally, the assumption $h^2 > \frac{r_{k-1}}{2}$ implies

\[
r_{k-1} - h^2 < \frac{r_{k-1}}{2} < h^2 \leq d^2. \quad \text{(5)}
\]

Hence we can be sure that (4) holds. Plugging in $\hat{v}_A$ and $\hat{v}_B$ yields

\[
v = f(x_k^+) - \frac{\alpha}{2} \left( r_{k-1} - \frac{1}{\kappa} h^2 - \frac{(d^2 + r_{k-1} - h^2)^2}{4d^2} \right).
\]

Hence the proof is complete once we show the inequality

\[
r_{k-1} - \frac{1}{\kappa} h^2 - \frac{(d^2 + r_{k-1} - h^2)^2}{4d^2} \leq \left( 1 - \frac{1}{\sqrt{\kappa}} \right) r_{k-1}.
\]

After rearranging, our task simplifies to showing the inequality

\[
r_{k-1} \leq \frac{h^2}{\kappa} + \frac{(d^2 + r_{k-1} - h^2)^2}{4d^2}.
\]

Taking derivatives and using inequality (5), one can readily verify that the right-hand-side is nondecreasing in $d^2$ on the interval $d^2 \in [h^2, +\infty)$. Thus plugging in the endpoint $d^2 = h^2$ we deduce

\[
\frac{h^2}{\kappa} + \frac{(d^2 + r_{k-1} - h^2)^2}{4d^2} \geq \frac{h^2}{\kappa} + \frac{h^2}{4h^2}.
\]

Minimizing the right-hand-side over all $h$ satisfying $h^2 \geq \frac{r_{k-1}}{2}$ yields the inequality

\[
\frac{h^2}{\kappa} + \frac{r_{k-1}^2}{4h^2} \geq \frac{r_{k-1}}{\sqrt{\kappa}}.
\]

The proof is complete. \(\square\)

It is instructive to compare optimal averaging (Algorithm 1) with Nesterov’s optimal methods in [10,11]. For convenience, we record the optimal gradient method following [10], in Algorithm 2.

Comparing Algorithms 1 and 2 we see that


| Algorithm 2: General scheme of an optimal method [Nesterov] |
|------------------------------------------------------------|
| **Input:** Starting points $x_0$ and $c_0$, strong convexity constant $\alpha > 0$, smoothness parameter $\beta > 0$, and initial quadratic curvature $\gamma_0 \geq \alpha$. |
| **Output:** Final quadratic $Q_K(x) = v_K + \frac{\gamma_K}{2} \|x - c_K\|^2$. |
| Set $Q_0(x) = v_0 + \frac{\gamma_0}{2} \|x - c_0\|^2$, where $v_0 = f(x_0) - \frac{1}{2\alpha} \|\nabla f(x_0)\|^2$; |
| for $k = 1, \ldots, K$ do |
| Compute averaging parameter $\lambda_k \in (0, 1)$ from $\beta \lambda_k^2 = (1 - \lambda_k) \gamma_{k-1} + \lambda_k \alpha$; |
| Set $\gamma_k = (1 - \lambda_k) \gamma_{k-1} + \lambda_k \alpha$; |
| Set $x_k = (1 - \theta_k)c_{k-1} + \theta_k x_{k-1}^+$ where $\theta_k = \frac{\gamma_k}{\gamma_{k-1} + \lambda_k \alpha}$; |
| Set $Q(x) = \left(f(x_k) - \frac{\|\nabla f(x_k)\|^2}{2\alpha}\right) + \frac{\alpha}{2} \|x - x_{k-1}^+\|^2$; |
| Let $c_k$ be the minimizer of the quadratic $Q_k(x) = (1 - \lambda_k)Q_{k-1}(x) + \lambda_k Q(x)$; |
| end |
| /* If we set $\gamma_0 = \alpha$, then we have $\gamma_k = \alpha$, $\lambda_k = \frac{1}{\sqrt{\kappa}}$, and $\theta_k = \frac{\sqrt{\kappa}}{1 + \sqrt{\kappa}}$. */ |

- $x_k$ is some point on the line between $c_{k-1}$ and $x_{k-1}^+$, and
- $Q_k$ is an average of the previous quadratic $Q_{k-1}$ and the strong convexity quadratic lower bound $Q$ based at $x_k$.

As we discuss in Appendix [A], we can modify Nesterov’s method so that like in optimal quadratic averaging, we set $x_k = \text{line_search}(c_{k-1}, x_{k-1}^+)$ in each iteration. After this change, only two differences remain between the schemes:

- the initial quadratic $Q_0$ is different, and
- the averaging parameter is computed differently.

These differences, however, are fundamental. In Algorithm [1], the quadratic $Q_0$ lower bounds $f$ and therefore optimal averaging makes sense; in the accelerated gradient method, $Q_0$ does not lower bound $f$, and the idea of optimal averaging does not apply.

### 3 Optimal quadratic averaging with memory

Each iteration of Algorithm [1] forms an optimal average of the current lower quadratic model with the one from the previous iteration; that is, as stated the scheme has a memory size of one. We next show how the scheme easily adapts to maintaining limited memory, i.e. by averaging multiple quadratics in each iteration. We mention in passing that the authors of [5] left open the question of efficiently speeding up their geometric descent algorithm in practice. One approach of this flavor has recently appeared in [4, Section 4]. The optimal averaging viewpoint, developed here, provides a direct and satisfying alternative. Indeed, computing the optimal average of several quadratics is easy, and amounts to solving a small dimensional quadratic optimization problem.

To see this, fix $t$ quadratics $Q_t(x) := v_i + \frac{\alpha}{2} \|x - c_i\|^2$, with $i \in \{1, \ldots, t\}$, and a weight vector $\lambda$ in the $t$-dimensional simplex $\Delta_t := \{ x \in \mathbb{R}_0^t : \sum_{i=1}^t x_i = 1, \ x \geq 0 \}$. The average
quadratic

\[ Q_\lambda(x) := \sum_{i=1}^{t} \lambda_i Q_i(x) \]

maintains the same canonical form as each \( Q_i \).

**Proposition 3.1.** Define the matrix \( C = [c_1 \ c_2 \ \ldots \ c_t] \) and vector \( v = [v_1 \ v_2 \ \ldots \ v_t]^T \).

Then we have

\[ Q_\lambda(x) = v_\lambda + \frac{\alpha}{2} \|x - c_\lambda\|^2, \]

where

\[ c_\lambda = C \lambda \quad \text{and} \quad v_\lambda = \left\langle \frac{\alpha}{2} \text{diag} (C^T C) + v, \lambda \right\rangle - \frac{\alpha}{2} \|C \lambda\|^2. \]

**Proof.** The Hessian of \( Q_\lambda \) is simply \( \frac{\alpha}{2} I \), and therefore the quadratic \( Q_\lambda(x) \) has the form

\[ v_\lambda + \frac{\alpha}{2} \|x - c_\lambda\|^2 \]

for some \( v_\lambda \) and \( c_\lambda \). Notice that \( c_\lambda \) is the minimizer of \( Q_\lambda \), and by differentiating, we determine that \( c_\lambda = \sum_{i=1}^{t} \lambda_i c_i = C \lambda \). We then compute

\[
\begin{align*}
v_\lambda &= Q_\lambda(c_\lambda) = \sum_{i=1}^{t} \left( \lambda_i v_i + \frac{\lambda_i \alpha}{2} \|C \lambda - c_i\|^2 \right) \\
&= \langle v, \lambda \rangle + \frac{\alpha}{2} \sum_{i=1}^{t} \lambda_i \left( \|C \lambda\|^2 - 2 \langle C \lambda, c_i \rangle + \|c_i\|^2 \right) \\
&= \langle v, \lambda \rangle + \frac{\alpha}{2} \|C \lambda\|^2 - \alpha \left\langle C \lambda, \sum_{i=1}^{t} \lambda_i c_i \right\rangle + \frac{\alpha}{2} \sum_{i=1}^{t} \lambda_i \|c_i\|^2 \\
&= \left\langle \frac{\alpha}{2} \text{diag} (C^T C) + v, \lambda \right\rangle - \frac{\alpha}{2} \|C \lambda\|^2.
\end{align*}
\]

The proof is complete. \( \square \)

Naturally, we define the optimal averaging of the quadratics \( Q_i \), with \( i \in \{1, 2, \ldots, t\} \), to be \( Q_\bar{\lambda} \), where \( \bar{\lambda} \) is the maximizer of the concave quadratic over the simplex:

\[
\min_{\lambda \in \Delta_t} v_\lambda = \left\langle \frac{\alpha}{2} \text{diag} (C^T C) + v, \lambda \right\rangle - \frac{\alpha}{2} \|C \lambda\|^2.
\]

There is no closed form expression for \( \bar{\lambda} \), but one can quickly find it by solving a quadratic program in \( t \) variables, for example by an active set method. Moreover, some thought shows that the matrix \( C^T C \) can be efficiently updated if one of the centers changes; we omit the details.

We propose an optimal averaging scheme with memory in Algorithm 3. As we see in Section 5, the method performs well numerically. Moreover, the scheme enjoys the same convergence guarantees as Algorithm 1; that is, Theorem 2.3 applies to Algorithm 3, with nearly the same proof (which we omit).
Algorithm 3: Optimal Quadratic Averaging with Memory

**Input:** Starting point $x_0$, strong convexity constant $\alpha > 0$, and memory size $t \geq 1$.

**Output:** Final quadratic $Q_K(x) = v_K + \frac{\alpha}{2} \|x - c_K\|^2$ and $x_K^+$.

Set $Q_0(x) = v_0 + \frac{\alpha}{2} \|x - c_0\|^2$, where $v_0 = f(x_0) - \frac{\|\nabla f(x_0)\|^2}{2\alpha}$ and $c_0 = x_0^+$.

for $k = 1, \ldots, K$

Set $x_{k+1} = \text{line_search}(c_{k-1}, x_{k-1})$;

Set $M_k(x) = f(x_k) - \frac{\|\nabla f(x_k)\|^2}{2\alpha} + \frac{\alpha}{2} \|x - x_k^+\|^2$;

Let $Q_k(x) := v_k + \frac{\alpha}{2} \|x - c_k\|^2$ be the optimal averaging of the

- $k + 1$ quadratics $Q_{k-1}, M_k, M_{k-1}, \ldots, M_1$ if $k \leq t$, or of the
- $t + 1$ quadratics $Q_{k-1}, M_k, M_{k-1}, \ldots, M_{k-t+1}$ if $k \geq t + 1$;

end

The reader may notice that Algorithm 3 shows some similarity to the classical Kelley’s method for minimizing nonsmooth convex functions [8]. In the simplest case of minimizing a smooth convex function $f$ on $\mathbb{R}^n$, Kelley’s method iterates the following steps

$$x_{k+1} = \arg\min_x f_k(x)$$

for the functions

$$f_k(x) := \max_{i=1,\ldots,k} \{f(x_i) + \langle \nabla f(x_i), x - x_i \rangle \}.$$ 

In other words, the scheme iteratively minimizes the (piecewise linear) lower-models $f_k$ of $f$. Coming back to the optimal averaging viewpoint, suppose that $Q_{\lambda}$ is an optimal average of the lower-bounding quadratics $Q_i$, for $i = 1, \ldots, k$. Then we may write

$$v_{\lambda} = \max_{\lambda \in \Delta_k} \min_x \sum_i \lambda_i Q_i(x) = \min_x \max_{\lambda \in \Delta_k} \sum_i \lambda_i Q_i(x) = \min_x \left( \max_{i=1,\ldots,k} Q_i(x) \right)$$

Thus $v_{\lambda}$ is the minimal value of the now different lower-model, $\max_{i=1,\ldots,k} Q_i$, of $f$. Kelley’s method is known to have poor numerical performance and convergence guarantees (e.g. [10, Section 3.3.2]), while Algorithm 3 achieves the optimal linear convergence rate. This disparity is of course based on the two key distinctions: (1) using quadratic lower-models coming from strong convexity instead of linear functions, and (2) maintaining two separate sequences $c_k$ (centers) and $x_k$ (sources of lower model updates).

4 Equivalence to geometric descent

Algorithm 4 is largely motivated by the geometric descent method introduced by Bubeck, Lee, and Singh [5]. In this section, we show the two methods (Algorithm 1 and Algorithm 4) indeed generate an identical iterate sequence.
4.1 Suboptimal geometric descent method

The basic idea of geometric descent [5] is that for each point \( x^* \in \mathbb{R}^n \), the strong convexity lower bound \( f^* \geq q(x^*; x) \) defines a ball containing \( x^* \):

\[
x^* \in B \left( x^+, \frac{\|\nabla f(x)\|^2}{\alpha^2} - \frac{2}{\alpha} (f(x) - f^*) \right).
\]

In turn, taking into account (1) yields the guarantee

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

A crude upper estimate of the radius above is obtained simply by ignoring the nonnegative term \( \frac{2}{\alpha} (f(x^+) - f^*) \). The suboptimal geometric descent method proceeds as follows. Suppose we have available some ball \( B(c_0, R_0^2) \) containing \( x^* \). As discussed, the quadratic lower bound at the center \( c_0 \), namely \( f^* \geq q(x^*, c_0) \), yields another ball \( B \left( c_0^+, \left( 1 - \frac{1}{\kappa} \right) \frac{\|\nabla f(c_0)\|^2}{\alpha^2} \right) \) containing \( x^* \). Geometrically it is clear that the intersection of these two balls must be significantly smaller than either of the individual balls. The following lemma from [5] makes this observation precise; see Figure 2 for an illustration.

**Lemma 4.1** (Minimal enclosing ball of the intersection). Fix a center \( x \in \mathbb{R}^n \), square radius \( R^2 > 0 \), step \( h \in \mathbb{R}^n \), and \( \epsilon \in (0, 1) \). Then there exists a new center \( c \in \mathbb{R}^n \) with

\[
B \left( x, R^2 \right) \cap B \left( x + h, (1 - \epsilon) \|h\|^2 \right) \subset B \left( c, (1 - \epsilon) R^2 \right).
\]

An application of Lemma 4.1 yields a new center \( c_1 \) with

\[
B \left( c_0, R_0^2 \right) \cap B \left( c_0^+, \left( 1 - \frac{1}{\kappa} \right) \frac{\|\nabla f(c_0)\|^2}{\alpha^2} \right) \subset B \left( c_1, \left( 1 - \frac{1}{\kappa} \right) R_0^2 \right).
\]

Repeating the procedure with the new ball \( B \left( c_1, \left( 1 - \frac{1}{\kappa} \right) R_0^2 \right) \) yields a sequence of centers \( c_k \) satisfying

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

We note that the centers \( c_k \) and \( R_0^2 \) of the minimal enclosing balls in Lemma 4.1 are easy to compute; see Algorithm 1 in [5].

There is a very close connection between finding the minimal enclosing ball of the intersection of two balls and of optimally averaging quadratics. To see this, consider again two quadratics

\[
f(x) \geq Q_A(x) := v_A + \frac{\alpha}{2} \|x - x_A\|^2 \quad \text{and} \quad f(x) \geq Q_B(x) := v_B + \frac{\alpha}{2} \|x - x_B\|^2.
\]
Let $Q$ be the optimal average of $Q_A$ and $Q_B$. Notice that since $Q_A$, $Q_B$, and $Q$ lower bound $f$, the minimizer $x^*$ of $f$ is guaranteed to lie in the three balls:

- $B(x_A, R_A^2)$ where $R_A^2 = \frac{2}{\alpha} \left( \hat{f} - v_A \right)$,
- $B(x_B, R_B^2)$ where $R_B^2 = \frac{2}{\alpha} \left( \hat{f} - v_B \right)$,
- $B(\bar{c}, R^2)$ where $R^2 = \frac{2}{\alpha} \left( \hat{f} - \bar{v} \right)$,

where $\hat{f}$ is any upper bound on $f^*$. We observe the following elementary fact.

**Proposition 4.2** (Minimal enclosing ball and optimal averaging). The ball $B(\bar{c}, R^2)$ is precisely the minimal enclosing ball of the intersection $B(x_A, R_A^2) \cap B(x_B, R_B^2)$.

**Proof.** Define the quantity $\hat{\lambda} = \frac{1}{2} + \frac{v_A - v_B}{\alpha \|x_A - x_B\|^2}$. If $\hat{\lambda}$ lies in the unit interval $[0, 1]$, then a quick computation using Lemma 2.2 shows the expressions

$$R^2 = R_B^2 - \frac{\left( \|x_A - x_B\|^2 + R^2_B - R_A^2 \right)^2}{4 \|x_A - x_B\|^2}$$

and

$$\bar{c} = \hat{\lambda} x_A + (1 - \hat{\lambda}) x_B = \frac{1}{2} (x_A + x_B) - \frac{R_A^2 - R_B^2}{2 \|x_A - x_B\|^2} (x_A - x_B).$$

Now observe

- $\hat{\lambda} < 0$ if and only if $\|x_A - x_B\|^2 < R_A^2 - R_B^2$.
- $\hat{\lambda} \in [0, 1]$ if and only if $\|x_A - x_B\|^2 \geq |R_A^2 - R_B^2|$, and
- $\hat{\lambda} > 1$ if and only if $\|x_A - x_B\|^2 < R_B^2 - R_A^2$.

Comparing with the recipe [5, Algorithm 1] for computing the minimal enclosing ball, we see that $B(\bar{c}, R^2)$ is the minimal enclosing ball of the intersection $B(x_A, R_A^2) \cap B(x_B, R_B^2)$. □
4.2 Optimal geometric descent method

To obtain an optimal method, the authors of [5] observe that the term \( \frac{2}{\alpha} (f(x^+) - f^*) \) in the inclusion (3) cannot be ignored. Exploiting this term will require maintaining two sequences \( c_k \) (the centers of the balls) and \( x_k \) (points for generating new balls). Suppose in iteration \( k \), we know that \( x^* \) lies in the ball

\[
B \left( c_k, R_k^2 - \frac{2}{\alpha} (f(x^*_k) - f^*) \right).
\]

Consider now an arbitrary point, denoted suggestively by \( x_{k+1} \). Then (6) implies the inclusion

\[
x^* \in B \left( x_{k+1}^+, \left( 1 - \frac{1}{\kappa} \right) \frac{\|\nabla f(x_{k+1})\|^2}{\alpha^2} - \frac{2}{\alpha} (f(x_{k+1}) - f^*) \right).
\]

If we choose \( x_{k+1} \) to satisfy \( f(x_{k+1}) \leq f(x_k^+) \) and apply inequality (1) with \( x = x_{k+1} \), we can get a new upper estimate of the initial ball,

\[
x^* \in B \left( c_k, R_k^2 - \frac{1}{\kappa} \frac{\|\nabla f(x_{k+1})\|^2}{\alpha^2} - \frac{2}{\alpha} (f(x_{k+1}) - f^*) \right).
\]

It seems clear that if the centers \( c_k \) and \( x_{k+1}^+ \) of the two balls in (7) and (8) are “sufficiently far apart”, then their intersection is contained in an even smaller ball. This is the content of following lemma from [5].

**Lemma 4.3** (Two balls shrinking). Fix centers \( x_A, x_B \in \mathbb{R}^n \) and square radii \( r_A^2, r_B^2 > 0 \). Also fix \( \epsilon \in (0, 1) \) and suppose \( \|x_A - x_B\|^2 \geq r_B^2 \). Then there exists a new center \( c \in \mathbb{R}^n \) such that for any \( \delta > 0 \), we have

\[
B \left( x_A, r_A^2 - \epsilon r_B^2 - \delta \right) \cap B \left( x_B, (1 - \epsilon) r_B^2 - \delta \right) \subset B \left( c, (1 - \sqrt{\epsilon}) r_A^2 - \delta \right).
\]

A quick application of this result shows that provided

\[
\|x_{k+1}^+ - c_k\|^2 \geq \frac{\|\nabla f(x_{k+1})\|^2}{\alpha^2}
\]

holds, there exists a new center \( c_{k+1} \) with

\[
x^* \in B \left( c_{k+1}, \left( 1 - \frac{1}{\sqrt{\kappa}} \right) R_k^2 - \frac{2}{\alpha} (f(x_{k+1}) - f^*) \right).
\]

One way to ensure that \( x_{k+1} \) satisfies the two key conditions, \( f(x_{k+1}) \leq f(x_k^+) \) and inequality (9), is to simply let \( x_{k+1} \) be the minimizer of \( f \) along the line between \( c_k \) and \( x_k^+ \). Trivially this guarantees the inequality \( f(x_{k+1}) \leq f(x_k^+) \), while the univariate optimality condition \( \nabla f(x_{k+1}) \perp (c_k - x_{k+1}) \) means the triangle with vertices \( x_{k+1}, x_k^+, c_k \) is a right triangle and inequality (9) becomes “the hypotenuse is longer than a leg.” This is exactly the motivation for the line-search procedure in Algorithm [1]. Repeating the process yields iterates \( c_k \) that satisfy the optimal linear rate of convergence

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

The precise method is described in Algorithm [4].
Remark 4.4. When applying an iterative method to compute $x_{k+1} = \text{line\_search} \left( c_k, x_k^+ \right)$, one can use the following termination criterion. Check if $c_k$ satisfies $f(c_k) \leq f(x_k^+)$, then stop and set $x_{k+1} := c_k$. Notice (9) holds trivially with this choice of $x_{k+1}$. Else stop with a trial point $z$ on the line joining $c_k$ and $x_k^+$ satisfying $f(z) \leq f(x_k^+)$ and

$$\|z^+ - c_k\|^2 \geq \frac{\|\nabla f(z)\|^2}{\alpha^2}.$$

We claim that the line search will terminate in finite time, unless $\text{line\_search} \left( c_k, x_k^+ \right)$ is the true minimizer of $f$. Indeed, since $c_k \neq \text{line\_search} \left( c_k, x_k^+ \right)$ (otherwise we would have terminated in the if clause), one can easily check that $z = \text{line\_search} \left( c_k, x_k^+ \right)$ satisfies the above inequality strictly.

Algorithm 4: Geometric Descent Method [Bubeck, Lee, Singh]

| Input: Starting point $x_0$, strong convexity constant $\alpha > 0$. |
| Output: $x_k^+$ |
| Set $c_0 = x_0^+$ and $R_0^2 = \|\nabla f(x_0)\|^2 - \frac{2}{\alpha} \left( f(x_0) - f(x_0^+) \right)$; |
| for $k = 1, \ldots, K$ do |
| Set $x_k = \text{line\_search} \left( x_{k-1}^+, c_{k-1} \right)$; |
| Set $x_A = x_k - \alpha^{-1} \nabla f(x_k)$ and $R_A^2 = \|\nabla f(x_k)\|^2 - \frac{2}{\alpha} \left( f(x_k) - f(x_k^+) \right)$; |
| Set $x_B = c_{k-1}$ and $R_B^2 = R_{k-1}^2 - \frac{2}{\alpha} \left( f(x_{k-1}^+) - f(x_k^+) \right)$; |
| Let $B(c_k, R_k^2)$ be the smallest enclosing ball of $B(x_A, R_A^2) \cap B(x_B, R_B^2)$; |
| end |

The following theorem shows that Algorithm 1 and Algorithm 4 indeed produce the same iterate sequence.

Theorem 4.5. Given the same initial point $x_0$, Algorithm 2 and Algorithm 4 produce the same iterates $x_k$ and $c_k$. Moreover, we have $v_k = f(x_k^+) - \frac{\alpha}{2} R_k^2$, where $v_k$ is the minimum value of the quadratic $Q_k$ in Algorithm 2 and $R_k$ is the radius of the ball in Algorithm 4.

Proof. Let $x_k$ and $c_k$ denote the iterates in Algorithm 1 and let $\hat{x}_k$ and $\hat{c}_k$ be the iterates in Algorithm 4. We proceed by induction on $k$. It follows immediately from the definition of the algorithms that $x_0 = \hat{x}_0, c_0 = \hat{c}_0$, and $v_0 = f(x_0^+) - \frac{\alpha}{2} R_0^2$. Now suppose, as an inductive assumption, $x_{k-1} = \hat{x}_{k-1}, c_{k-1} = \hat{c}_{k-1}$, and $v_{k-1} = f(x_{k-1}^+) - \frac{\alpha}{2} R_{k-1}^2$. To see the equality $x_k = \hat{x}_k$, observe

$$x_k = \text{line\_search} \left( x_{k-1}^+, c_{k-1} \right) = \text{line\_search} \left( \hat{x}_{k-1}^+, \hat{c}_{k-1} \right) = \hat{x}_k.$$

Let $x_A = x_k^+, x_B = c_{k-1}, d = \|x_A - x_B\|$, and define the quantities

$$v_A = f(x_k) - \frac{\|\nabla f(x_k)\|^2}{2\alpha}, \quad R_A^2 = \frac{\|\nabla f(x_k)\|^2}{\alpha^2} - \frac{2}{\alpha} \left( f(x_k) - f(x_k^+) \right),$$

$$v_B = v_{k-1}, \quad R_B^2 = R_{k-1}^2 - \frac{2}{\alpha} \left( f(x_{k-1}^+) - f(x_k^+) \right).$$
Notice that $Q_k(x) = v_k + \frac{\alpha}{2} \|x - c_k\|^2$ is the optimal averaging of $Q_A(x) := v_A + \frac{\alpha}{2} \|x - x_A\|^2$ and $Q_B(x) := v_B + \frac{\alpha}{2} \|x - x_B\|^2$, and that $B(\hat{c}_k, R_k^2)$ is the minimum enclosing ball of the intersection of $B(x_A, R_A^2)$ and $B(x_B, R_B^2)$. Simple algebra shows the relation
\[
R_A^2 = \frac{2}{\alpha} \left( f(x_k^+) - v_A \right),
\]
and from the inductive assumption $v_{k-1} = f(x_{k-1}^+) - \frac{\alpha}{2} R_{k-1}^2$, we also have
\[
R_B^2 = \frac{2}{\alpha} \left( f(x_k^+) - v_B \right).
\]
Thus, by Proposition 4.2 and the discussion preceding it, we have $c_k = \hat{c}_k$ and $v_k = f(x_k^+) - \frac{\alpha}{2} R_k^2$. This completes the induction. \qed

As we saw in Section 3, computing the optimal averaging of several quadratic functions is simple. On the other hand, it is far from clear how to find the minimum radius ball that encloses the intersection of more than two balls. Indeed, instead the authors of Algorithm 4 in the follow-up work [4] considered a “relaxation” that involves minimizing a self-concordant barrier for the intersection. While revising the current manuscript, we became aware that Beck in [3, Theorem 3.2] proved that the minimum enclosing ball of the intersection of finitely many balls can be computed by solving a convex quadratic program (QP). Namely, Beck showed that the squared radius of the minimal ball enclosing the intersection $\bigcap_{i=1}^t B(c_i, r_i^2)$ is exactly equal to
\[
\min_{\lambda \in \Delta_t} \left\{ \left\| \sum_{i=1}^t \lambda_i c_i \right\|^2 - \sum_{i=1}^t \lambda_i (\|a_i\|^2 - r_i^2) \right\},
\]
provided $t \leq n - 1$ and the intersection of the balls has nonempty interior. This QP is exactly the one we derived in Section 3 for the optimal quadratic averaging method with memory. Note that our derivation of the QP in Section 3 was completely elementary; the proof of [3, Theorem 3.2], on the other hand, is much more sophisticated relying on an S-lemma-type result.

**Proposition 4.6 (Optimal quadratic averaging & minimal enclosing ball).**

Let $Q(x) = v + \frac{\alpha}{2} \|x - c\|^2$ be the optimal averaging of quadratics $Q_i(x) = v_i + \frac{\alpha}{2} \|x - c_i\|^2$ for $i = 1, \ldots, t$ with $t < n$. Fix a real number $s \geq v_i$ for all $i = 1, \ldots, t$ and define the balls $B_i := \{Q_i \leq s\}$. Then provided that the intersection $\bigcap_{i=1}^t B_i$ has a nonempty interior, the ball $B := \{Q \leq s\}$ is the minimal enclosing ball of the intersection $\bigcap_{i=1}^t B_i$.

**Proof.** Let $R^2$ be the square radius of $B$ and let $R_i^2$ be the square radius of $B_i$, for $i = 1, \ldots, t$. Using Proposition 3.1, we deduce
\[
R^2 = \frac{2}{\alpha} (s - v) = \frac{2}{\alpha} \left( s - \max_{\lambda \in \Delta_t} \left\{ \frac{\alpha}{2} \sum_{i=1}^t \lambda_i \left( \frac{\alpha}{2} \|c_i\|^2 + v_i \right) - \frac{\alpha}{2} \left\| \sum_{i=1}^t \lambda_i c_i \right\|^2 \right\} \right)
\]
\[
= \min_{\lambda \in \Delta_t} \left\{ \sum_{i=1}^t \lambda_i \|c_i\|^2 - \sum_{i=1}^t \lambda_i \left( \|c_i\|^2 - R_i^2 \right) \right\} = \min_{\lambda \in \Delta_t} \left\{ \sum_{i=1}^t \lambda_i \|c_i\|^2 - \sum_{i=1}^t \lambda_i \left( \|c_i\|^2 - R_i^2 \right) \right\}.
\]
The center of $B$ is $c = \sum_{i=1}^{t} \lambda_i c_i$ where $\lambda$ is the minimizer of the expression above. Comparing with [3, Theorem 3.2], we see that $B$ is exactly the minimum radius ball enclosing the intersection $\bigcap_{i=1}^{t} B_i$.

5 Numerical examples

In this section, we numerically illustrate optimality gap convergence in Algorithm 1 and explore how Algorithm 3, the variant of Algorithm 1 with memory, aids performance. To this end, we focus on minimizing two functions: the regularized logistic loss function

$$L(w) := \frac{1}{N} \sum_{i=1}^{N} \log \left( 1 + e^{-y_i w^T x_i} \right) + \frac{\alpha}{2} \|w\|^2,$$

where $x_i \in \mathbb{R}^n$ and $y_i \in \{\pm 1\}$ are labeled training data, and the "world’s worst" function for first-order methods:

$$f(x) = \frac{B}{2} \left( (1-x_1)^2 + \sum_{i=1}^{n-1} (x_i - x_{i+1})^2 + x_n^2 \right) + \frac{1}{2} \sum_{i=1}^{n} x_i^2$$

(see [10, Section 2.1.2 and Section 2.1.4]). For the logistic regression examples, we use the LIBSVM [6] data sets a1a ($N = 1605$, $n = 123$) and colon-cancer ($N = 62$, $n = 2000$).

5.1 Optimality gap convergence

From inequality (2), we get the well-known optimality gap estimate for strongly convex functions

$$f(x) - f^* \leq \frac{\|\nabla f(x)\|^2}{2\alpha}.$$ (10)

How does this estimate compare with the gaps $g_k := f(x_k^+) - v_k$ generated by Algorithm 1? Obviously the answer depends on the point where we evaluate the gap estimate in (10). Nonetheless, we can say that the gaps $g_k$ are tighter than the gaps $G_k := \frac{\|\nabla f(x_k)\|^2}{2\alpha}$. Indeed, by the definition of $v_k$, we trivially have $v_k \geq f(x_k) - G_k$ and thus

$$g_k = f(x_k^+) - v_k \leq f(x_k) - v_k \leq G_k.$$

On a relative scale, the difference between $g_k$ and $G_k$ is striking; see Figure 3. Notice that $G_k$ is an optimality gap estimate before averaging, and $g_k$ is an optimality gap estimate after averaging; the plots in Figure 3 show that optimal quadratic averaging makes great relative progress per iteration.

In Figure 4, we plot $g_k$, the true gaps $f(x_k^+) - f^*$, and the gap estimate in (10) at $x_k$, $x_k^+$, and $c_k$ for the "world’s worst" function and the logistic loss function. The true gaps are the tightest, albeit unknown at runtime. Surprisingly, the gaps $\frac{\|\nabla f(x_k)\|^2}{2\alpha}$ are quite bad: several orders of magnitude larger than $g_k$. So even though the centers $c_k$ may appear to be the focal points of the algorithm, the points $x_k^+$ are the ones to monitor in practice. Finally we note that the gaps $g_k$ and $\frac{\|\nabla f(x_k^+)\|^2}{2\alpha}$ are comparable, even though $g_k$ does not rely on gradient information at $x_k^+$.
Figure 3: Relative differences in gaps $\frac{G_k - g_k}{\epsilon_k}$ on the “world’s worst” function ($B = 10^6$, $n = 200$), and on the logistic loss on the colon-cancer data set with regularization $\alpha = 0.0001$.

Figure 4: Comparison of various optimality gaps on the “world’s worst” function ($B = 10^6$, $n = 200$), and on the logistic loss on the a1a data set with regularization $\alpha = 0.0001$.

5.2 Optimal quadratic averaging with memory

To demonstrate the effectiveness of optimal quadratic averaging with memory, we use it to minimize the logistic loss (see Figure 5). The speedup over the memoryless method is significant, even when taking into account the extra work per iteration needed to solve the small dimensional quadratic subproblems. In Figure 6, we compare Algorithm 3 with L-BFGS. The two schemes are on par with each other, and neither is better than the other in all cases.

It is perhaps fairer to compare L-BFGS with memory size $m$ to Algorithm 3 with memory
Figure 5: Algorithm 3 with various memory sizes $t$. The case $t = 1$ corresponds to the memoryless optimal averaging method in Algorithm 1. The task is logistic regression, with regularization $\alpha = 0.0001$, on data sets a1a and colon-cancer.

Figure 6: Algorithm 3 with memory size $t$ versus L-BFGS with memory size $m$. The task is logistic regression, with regularization $\alpha = 0.0001$, on data sets a1a and colon-cancer.

size $t = 2m$ (see Figure 7). Indeed, L-BFGS with memory size $m$ actually stores $m$ pairs of vectors, whereas Algorithm 3 with memory size $t$ only stores $t$ vectors. Moreover, the most expensive operation per iteration in L-BFGS requires $4mn$ multiplications (see [12, Algorithm 7.4]); in contrast, computing a new center in Algorithm 3 requires $2n(t + 1)$ multiplications plus the cost of solving a small quadratic program. (Updating the matrix $C^TC$ takes $t + 1$ inner products in $\mathbb{R}^n$, finding $\lambda$ amounts to solving a small quadratic program, and computing $CA$ takes $n$ inner products in $\mathbb{R}^{t+1}$.) In Figure 8, we again compare L-BFGS and Algorithm 3 on logistic regression, but with less regularization.
Figure 7: A fairer (equal memory) comparison of Algorithm 3 and L-BFGS. The task is still logistic regression, with regularization $\alpha = 0.0001$, on data sets a1a and colon-cancer. We focus on lower accuracy than we did in Figure 6.

We noticed that the small dimensional quadratic program in Algorithm 3 must be solved to high accuracy, especially on poorly conditioned problems; an active-set method works well. Accuracy in the line search is less important. Minimizing the one-dimensional function $r \mapsto f(x + rd)$, with $\|d\| = 1$, to within $10^{-4}$ accuracy in $r$ works well in general. In Figure 9, we show how line search accuracy affects Algorithm 1.
Figure 8: Algorithm 3 with memory size $t$ versus L-BFGS with memory size $m$. The task is logistic regression on data sets a1a and colon-cancer, with $\alpha = 10^{-6}$ (top row) and $\alpha = 10^{-8}$ (bottom row).

6 Comments on proximal extensions

It is natural to try to extend geometric descent and optimal quadratic averaging to a proximal setting. For the sake of concreteness, let us focus on geometric descent. We can easily extend the suboptimal version of the algorithm to the proximal setting, but some difficulties arise when accelerating the method. Suppose we are interested in solving the problem

$$\min_x f(x) := g(x) + h(x),$$

where $g: \mathbb{R}^n \to \mathbb{R}$ is $\beta$-smooth and $\alpha$-strongly convex, and $h: \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is closed, convex, and is such that the proximal mapping

$$\text{prox}_t h(x) := \arg\min_z \{h(z) + \frac{1}{2t} \|z - x\|^2\}$$
Figure 9: A comparison of how the line search tolerance in Algorithm 1 affects convergence. In the top row, we do the comparison with logistic regression on the a1a and colon-cancer data sets with regularization $\alpha = 10^{-4}$. In the bottom row, we use regularization $10^{-8}$.

is easily computable. In the analysis of first-order methods for such problems, the gradient mapping $G_t(x) := \frac{1}{t} (x - \text{prox}_t h (x - t \nabla g(x)))$ plays the role of the usual gradient. The following is a standard estimate; see for example [10, Section 2.2.3]. We provide a proof for completeness.

**Lemma 6.1.** Fix a step length $t > 0$ and define a proximal gradient step $x^+ := x - tG_t(x)$. Then for every $y \in \mathbb{R}^n$ the inequality holds:

$$f(y) \geq f(x^+) + \langle G_t(x), y - x \rangle + t \left( 1 - \frac{\beta t}{2} \right) \|G_t(x)\|^2 + \frac{\alpha}{2} \|y - x\|^2.$$

**Proof.** Appealing to $\beta$-smoothness of $g$, we deduce

$$f(x^+) \leq g(x) - t \langle \nabla g(x), G_t(x) \rangle + \frac{\beta t^2}{2} \|G_t(x)\|^2 + h(x^+).$$
Furthermore, strong convexity of $g$ implies
\[ f(x^+) \leq g(y) + \langle \nabla g(x), x^+ - y \rangle - \frac{\alpha}{2} \| y - x \|^2 + \frac{\beta t^2}{2} \| G_t(x) \|^2 + h(x^+). \]

Finally, using the observation that $G_t(x) - \nabla g(x)$ belongs to $\partial h(x^+)$, we have
\[ f(x^+) \leq f(y) + \langle G_t(x), x^+ - y \rangle - \frac{\alpha}{2} \| y - x \|^2 + \frac{\beta t^2}{2} \| G_t(x) \|^2. \]

Rearrangement completes the proof. \[ \square \]

If we let $y = x^*$ in Lemma 6.1 and rearrange we get
\[ x^* \in B \left( x - \frac{1}{\alpha} G_t(x), \left( \frac{1}{\alpha^2} - \frac{2}{\alpha} t + \frac{\beta}{\alpha} t^2 \right) \| G_t(x) \|^2 - \frac{2}{\alpha} (f(x^*) - f^*) \right). \]

How should we choose the step length $t$? A simple approach is to choose $t$ to minimize the quantity $\frac{1}{\alpha^2} - \frac{2}{\alpha} t + \frac{\beta}{\alpha} t^2$, i.e., set $t = \frac{1}{\beta}$. With this choice of $t$, we deduce the inclusion
\[ x^* \in B \left( x^{++}, \left( 1 - \frac{1}{\kappa} \right) \| G_{1/\beta}(x) \|^2 - \frac{2}{\alpha} (f(x^*) - f^*) \right), \]
where $x^{++} = x - \frac{1}{\alpha} G_{1/\beta}(x)$ is a long step and $x^+ = x - \frac{1}{\beta} G_{1/\beta}(x)$ is a short step. A proximal version of the suboptimal geometric descent follows easily from Lemma 4.1.

To accelerate the proximal geometric descent algorithm we assume in iteration $k$ that $x^*$ lies in some ball
\[ B \left( c_k, R_k^2 - \frac{2}{\alpha} (f(y_k) - f^*) \right). \]

We then consider a second minimizer enclosing ball derived from information at some point $x_{k+1}$:
\[ x^* \in B \left( x_{k+1}^{++}, \left( 1 - \frac{1}{\kappa} \right) \| G_{1/\beta}(x_{k+1}) \|^2 - \frac{2}{\alpha} (f(x_{k+1}^*) - f^*) \right). \]

Following the same pattern as in Section 4.2, if we choose $x_{k+1}$ to satisfy $f(x_{k+1}) \leq f(y_k)$ and appeal to the smoothness inequality $f(x_{k+1}^{++}) \leq f(x_{k+1}) - \frac{1}{2\beta} \| G_{1/\beta}(x_{k+1}) \|^2$, we deduce the inclusion
\[ x^* \in B \left( c_k, R_k^2 - \frac{1}{\kappa} \| G_{1/\beta}(x_{k+1}) \|^2 - \frac{2}{\alpha} (f(x_{k+1}^*) - f^*) \right). \]

By Lemma 4.3 there is a new center $c_{k+1}$ with
\[ x^* \in B \left( c_k, \left( 1 - \frac{1}{\sqrt{\kappa}} \right) R_k^2 - \frac{2}{\alpha} (f(x_{k+1}^*) - f^*) \right), \]
provided the old centers $x_{k+1}^{++}$ and $c_k$ are far apart; specifically, we must be sure that the inequality
\[ \| x_{k+1}^{++} - c_k \|^2 \geq \| G_{1/\beta}(x_{k+1}) \|^2 \]
holds.
How do we choose $x_{k+1}$ to satisfy both $f(x_{k+1}) \leq f(y_k)$ and $\|x_{k+1}^+ - c_k\|^2 \geq \|G_{1/a}(x_{k+1})\|^2_{\alpha^2}$? The desired $x_{k+1}$ does exist; for example, $x_{k+1} = x^*$ is such a point. In the proximal setting, it is not clear how to choose $x_{k+1}$ to ensure these two inequalities (even for specific problem classes). This is an interesting topic for future research.

Acknowledgments

We thank the anonymous referee for useful suggestions, which undoubtedly improved the quality of the paper. We also thank Stephen J. Wright for pointing out an important typo in the proof of Theorem 2.3 in an early version of the manuscript.

References

[1] Z. Allen-Zhu and L. Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. Preprint, arXiv:1407.1537, 2016.

[2] H. Attouch, J. Peypouquet, and P. Redont. Fast convergence of inertial dynamics and algorithms with asymptotic vanishing viscosity. Math. Program., doi:10.1007/s10107-016-0992-8, pages 1–53, 2016.

[3] A. Beck. On the convexity of a class of quadratic mappings and its application to the problem of finding the smallest ball enclosing a given intersection of balls. J. Global Optim., 39(1):113–126, 2007.

[4] S. Bubeck and Y.T. Lee. Black-box optimization with a politician. Preprint, arXiv:1602.04847, 2016.

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

[6] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011.

[7] S. Chen and S. Ma. Geometric descent method for convex composite minimization. arXiv:1612.09094, 2017.

[8] J. E. Kelley, Jr. The cutting-plane method for solving convex programs. J. Soc. Indust. Appl. Math., 8:703–712, 1960.

[9] L. Lessard, B. Recht, and A. Packard. Analysis and Design of Optimization Algorithms via Integral Quadratic Constraints. SIAM J. Optim., 26(1):57–95, 2016.

[10] Y. Nesterov. Introductory lectures on convex optimization, volume 87 of Applied Optimization. Kluwer Academic Publishers, Boston, MA, 2004. A basic course.

[11] Yu. E. Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. Dokl. Akad. Nauk SSSR, 269(3):543–547, 1983.

[12] J. Nocedal and S.J. Wright. Numerical optimization. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006.
Nesterov’s method is based on an estimate sequence; that is, a sequence of functions \( Q_k \) and nonnegative numbers \( \Lambda_k \) with

\[
\Lambda_k \to 0 \quad \text{and} \quad Q_k(x) \leq (1 - \Lambda_k) f(x) + \Lambda_k Q_0(x).
\]

Estimate sequences are useful because if \( y_k \) satisfies \( f(y_k) \leq v_k := \min_{x \in \mathbb{R}^n} Q_k(x) \), then

\[
f(y_k) - f^* \leq \Lambda_k (Q_0(x^*) - f^*) ;
\]

that is, \( f(y_k) \) approaches \( f^* \) with error proportional to \( \Lambda_k \), see [10].

The quadratics in Algorithm 2 (with appropriately chosen \( \Lambda_k \)) form an estimate sequence. To explain, for \( k \geq 1 \), pick vectors \( x_k \) and numbers \( \lambda_k \in (\delta, 1) \) with \( \delta > 0 \). Next, recursively define

\[
Q_0(x) = v_0 + \frac{\gamma_0}{2} \|x - c_0\|^2 \quad \text{and} \quad Q_k(x) = (1 - \lambda_k) Q_{k-1}(x) + \lambda_k \left( f(x_k) - \frac{\|\nabla f(x_k)\|^2}{2\alpha} + \frac{\alpha}{2} \|x - x_{k+1}^+\|^2 \right).
\]

Then the quadratics \( Q_k \) and numbers \( \Lambda_k = \prod_{j=1}^k (1 - \lambda_j) \) are an estimate sequence for \( f \). Nesterov’s method is designed to ensure the inequality \( f(x_k^+) \leq v_k \) with the added optimal rate condition \( \lambda_k \geq \sqrt{\frac{\alpha}{\beta}} \).

The scheme in Algorithm 2 with \( x_k = \text{line_search}(c_{k-1}, x_{k-1}^+) \) also guarantees these conditions. Trivially we have \( f(x_0^+) \leq v_0 \). Assume, for induction, that we have \( f(x_{k-1}^+) \leq v_{k-1} \). From [10] Lemma 2.2.3, we know

\[
v_k = (1 - \lambda_k)v_{k-1} + \lambda_k f(x_k) - \frac{\lambda_k^2}{2\gamma_k} \|\nabla f(x_k)\|^2 + \lambda_k(1 - \lambda_k) \left( \frac{\alpha}{2} \|x_k - c_{k-1}\|^2 + \langle \nabla f(x_k), c_{k-1} - x_k \rangle \right).
\]

Since \( x_k = \text{line_search}(c_{k-1}, x_{k-1}^+) \), we have \( f(x_k) \leq f(x_{k-1}^+) \leq v_{k-1} \) and \( \langle \nabla f(x_k), c_{k-1} - x_k \rangle = 0 \), and therefore

\[
v_k \geq f(x_k) - \frac{\lambda_k^2}{2\gamma_k} \|\nabla f(x_k)\|^2 = f(x_k) - \frac{1}{2\beta} \|\nabla f(x_k)\|^2 \geq f(x_k^+).
\]

Provided we set \( \gamma_0 \geq \alpha \), we get the optimal rate condition \( \lambda_k = \sqrt{\frac{\alpha}{\beta}} \geq \sqrt{\frac{\alpha}{\beta}} \).