A geometric alternative to Nesterov's accelerated gradient descent

Sébastien Bubeck
Microsoft Research
sebubeck@microsoft.com

Yin Tat Lee*
MIT
yintat@mit.edu

Mohit Singh
Microsoft Research
mohits@microsoft.com

June 29, 2015

Abstract

We propose a new method for unconstrained optimization of a smooth and strongly convex function, which attains the optimal rate of convergence of Nesterov's accelerated gradient descent. The new algorithm has a simple geometric interpretation, loosely inspired by the ellipsoid method. We provide some numerical evidence that the new method can be superior to Nesterov's accelerated gradient descent.

1 Introduction

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a β-smooth and α-strongly convex function. Thus, for any $x, y \in \mathbb{R}^n$, we have

$$f(x) + \nabla f(x)^\top (y - x) + \frac{\alpha}{2}|y - x|^2 \leq f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{\beta}{2}|y - x|^2.$$

Let $\kappa = \beta/\alpha$ be its condition number. It is a one line calculation to verify that a step of gradient descent on f will decrease (multiplicatively) the squared distance to the optimum by $1 - 1/\kappa$. In this paper we propose a new method, which can be viewed as some combination of gradient descent and the ellipsoid method, for which the squared distance to the optimum decreases at a rate of $(1 - 1/\sqrt{\kappa})$ (and each iteration requires one gradient evaluation and two line-searches). This matches the optimal rate of convergence among the class of first-order methods, [Nesterov(1983), Nesterov(2004)].

1.1 Related works

Nesterov's acceleration (i.e., replacing κ by $\sqrt{\kappa}$ in the convergence rate) has proven to be of fundamental importance both in theory and in practice, see e.g. [Bubeck(2014)] for references. However

*Most of this work were done while the author was at Microsoft Research, Redmond. The author was supported by NSF awards 0843915 and 1111109.

the intuition behind Nesterov's accelerated gradient descent is notoriously difficult to grasp, and this has led to a recent surge of interest in new interpretations of this algorithm, as well as the reasons behind the possibility of acceleration for smooth problems, see [Allen-Zhu and Orecchia(2014), Lessard et al.(2014)Lessard, Recht, and Packard, Su et al.(2014)Su, Boyd, and Candès, Flammarion and Bach(20

In this paper we propose a new method with a clear intuition and which achieves acceleration. Since the function is strongly convex, gradient at any point gives a ball, say A, containing the optimum solution. Using the fact that the function is smooth, one can get an improved bound on the radius of this ball. The algorithm also maintains a ball B containing the optimal solution obtained via the information from previous iterations. A simple calculation then shows that the smallest ball enclosing the intersection of A and B already has a radius shrinking at the rate of $1 - \frac{1}{\kappa}$. To achieve the accelerated rate, we make the observation that the gradient information in this iteration can also be used to shrink the ball B and therefore, the radius of the enclosing ball containing the intersection of A and B shrinks at a faster rate. We detail this intuition in Section 2. The new optimal method is described and analyzed in Section 3. We conclude with some experiments in Section 4.

1.2 Preliminaries

We write $|\cdot|$ for the Euclidean norm in $\mathbb{R}^n$, and $\mathrm{B}(x, r^2) := \{y \in \mathbb{R}^n : |y - x|^2 \leq r^2\}$ (note that the second argument is the radius squared). We define the map $\texttt{line\_search} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ by

$$\texttt{line\_search}(x, y) = \operatorname*{argmin}_{t \in \mathbb{R}} f(x + t(y - x)),$$

and we denote

$$x^+ = x - \frac{1}{\beta}\nabla f(x), \text{ and } x^{++} = x - \frac{1}{\alpha}\nabla f(x).$$

Recall that by strong convexity one has

$$\forall y \in \mathbb{R}^n, f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\alpha}{2}|y - x|^2,$$

which implies in particular:

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

Furthermore recall that by smoothness one has $f(x^+) \leq f(x) - \frac{1}{2\beta}|\nabla f(x)|^2$ which allows to *shrink* the above ball by a factor of $1 - \frac{1}{\kappa}$ and obtain the following:

$$x^* \in \mathrm{B}\left(x^{++}, \frac{|\nabla f(x)|^2}{\alpha^2}\left(1 - \frac{1}{\kappa}\right) - \frac{2}{\alpha}(f(x^+) - f(x^*))\right) \tag{1}$$

2 Intuition

In Section 2.1 we describe a geometric alternative to gradient descent (with the same convergence rate) which gives the core of our new optimal method. Then in Section 2.2 we explain why one can expect to accelerate this geometric algorithm.

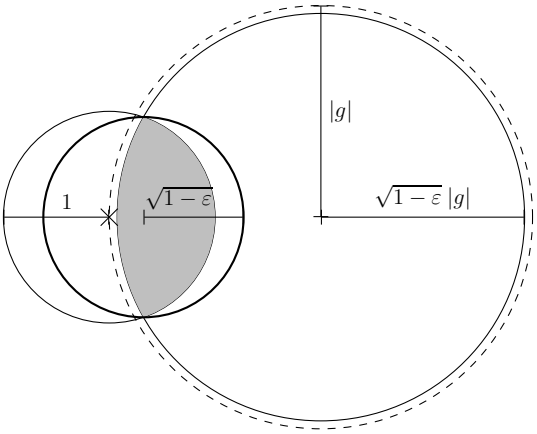

Figure 1: One ball shrinks.

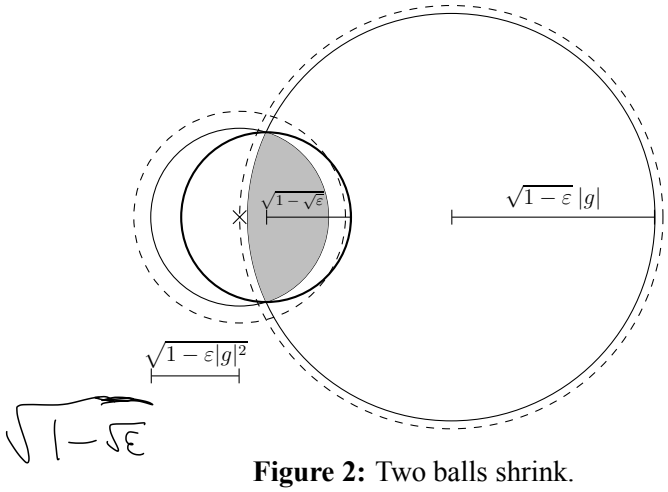

Figure 2: Two balls shrink.

The left diagram shows the intersection shrinks at the same rate if only one of the ball shrinks; the right diagram shows the intersection shrinks much faster if two balls shrinks at the same absolute amount.

2.1 A suboptimal algorithm

Assume that we are given a guarantee $R_0 > 0$ on the distance from some point x_0 to the optimum, that is $x^* \in \mathrm{B}(x_0, R_0^2)$. Combining this original enclosing ball for x^* and the one obtained by (1) (with $f(x^*) \leq f(x_0^+)$) one obtains

$$x^* \in \mathrm{B}(x_0, R_0^2) \cap \mathrm{B}\left(x_0 - \frac{1}{\alpha}\nabla f(x_0), \frac{|\nabla f(x_0)|^2}{\alpha^2}\left(1 - \frac{1}{\kappa}\right)\right).$$

If $\frac{|\nabla f(x_0)|^2}{\alpha^2} \leq R_0^2(1 - \frac{1}{\kappa})$ then the second ball already shrinks by a factor of $(1 - \frac{1}{\kappa})$. In the other case when $\frac{|\nabla f(x_0)|^2}{\alpha^2} > R_0^2(1 - \frac{1}{\kappa})$, the center of the two balls are far apart and therefore there is a much smaller ball containing the intersection of two balls. Formally, it is an easy calculation to see that for any $g \in \mathbb{R}^n$, $\varepsilon \in (0, 1)$, there exists $x \in \mathbb{R}^n$ such that

$$\mathrm{B}(0, 1) \cap \mathrm{B}(g, |g|^2(1 - \varepsilon)) \subset \mathrm{B}(x, 1 - \varepsilon). \qquad \text{(Figure 1)}$$

In particular the two above display implies that there exists $x_1 \in \mathbb{R}^n$ such that

$$x^* \in \mathrm{B}\left(x_1, R_0^2\left(1 - \frac{1}{\kappa}\right)\right).$$

Denote by T the map from x_0 to x_1 defined implicitely above, and let (x_k) be defined by $x_{k+1} = T(x_k)$. Then we just proved

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

In other words, after $2\kappa \log(R_0/\varepsilon)$ iterations where each iteration cost one call to the gradient oracle) one obtains a point ε-close to the minimizer of f.

2.2 Why one can accelerate

Assume now that we are give a guarantee $R_0 > 0$ such that $x^* \in \mathrm{B}(x_0, R_0^2 - \frac{2}{\alpha}(f(y) - f(x^*)))$ where $f(x_0) \leq f(y)$ (say by choosing $y = x_0$). Using the fact that $f(x_0^+) \leq f(x_0) - \frac{1}{2\beta}|\nabla f(x_0)|^2 \leq f(y) - \frac{1}{2\alpha\kappa}|\nabla f(x_0)|^2$, we obtain that

$$x^* \in \mathrm{B}\left(x_0, R_0^2 - \frac{|\nabla f(x_0)|^2}{\alpha^2\kappa} - \frac{2}{\alpha}\left(f(x_0^+) - f(x^*)\right)\right)$$

which, intuitively, allows us the shrink the radius squared from R_0^2 to $R_0^2 - \frac{|\nabla f(x_0)|^2}{\alpha^2\kappa}$ using the local information at x_0. From (1), we have

$$x^* \in \mathrm{B}\left(x_0^{++}, \frac{|\nabla f(x_0)|^2}{\alpha^2}\left(1 - \frac{1}{\kappa}\right) - \frac{2}{\alpha}\left(f(x_0^+) - f(x^*)\right)\right).$$

Now, intersecting the above two shrunk balls and using Lemma 1 (see below and also see Figure 2), we obtain that there is an x_1' such that

$$x^* \in \mathrm{B}\left(x_1', R_0^2\left(1 - \frac{1}{\sqrt{\kappa}}\right) - \frac{2}{\alpha}\left(f(x_0^+) - f(x^*)\right)\right)$$

giving us an acceleration in shrinking of the radius. To carry the argument for the next iteration, we would have required that $f(x_1') \leq f(x_0^+)$ but it may not hold. Thus, we choose x_1 by a line search

$$x_1 = \texttt{line\_search}\left(x_1', x_0^+\right)$$

which ensures that $f(x_1) \leq f(x_0^+)$. To remedy the fact that the ball for the next iteration is centered at x_1' and not x_1, we observe that the line search also ensures that $\nabla f(x_1)$ is perpendicular to the line going through x_1 and x_1'. This geometric fact is enough for the algorithm to work at the next iteration as well. In the next section we describe precisely our proposed algorithm which is based on the above insights.

3 An optimal algorithm

Let $x_0 \in \mathbb{R}^n$, $c_0 = x_0^{++}$, and $R_0^2 = \left(1 - \frac{1}{\kappa}\right)\frac{|\nabla f(x_0)|^2}{\alpha^2}$. For any $k \geq 0$ let

$$x_{k+1} = \texttt{line\_search}\left(c_k, x_k^+\right),$$

and c_{k+1} (respectively R_{k+1}^2) be the center (respectively the squared radius) of the ball given by (the proof of) Lemma 1 which contains

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

The formulas for c_{k+1} and R_{k+1}^2 are given in Algorithm 1.

Theorem 1 *For any $k \geq 0$, one has $x^* \in \mathrm{B}(c_k, R_k^2)$, $R_{k+1}^2 \leq \left(1 - \frac{1}{\sqrt{\kappa}}\right) R_k^2$, and thus*

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

Proof We will prove a stronger claim by induction that for each $k \geq 0$, one has

$$x^* \in \mathrm{B}\left(c_k, R_k^2 - \frac{2}{\alpha}\left(f(x_k^+) - f(x^*)\right)\right).$$

The case $k = 0$ follows immediately by (1). Let us assume that the above display is true for some $k \geq 0$. Then using $f(x^*) \leq f(x_{k+1}^+) \leq f(x_{k+1}) - \frac{1}{2\beta}|\nabla f(x_{k+1})|^2 \leq f(x_k^+) - \frac{1}{2\beta}|\nabla f(x_{k+1})|^2$, one gets

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

Furthermore by (1) one also has

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

Thus it only remains to observe that the squared radius of the ball given by Lemma 1 which encloses the intersection of the two above balls is smaller than $\left(1 - \frac{1}{\sqrt{\kappa}}\right) R_k^2 - \frac{2}{\alpha}(f(x_{k+1}^+) - f(x^*))$. We apply Lemma 1 after moving c_k to the origin and scaling distances by R_k. We set $\varepsilon = \frac{1}{\kappa}$, $g = \frac{|\nabla f(x_{k+1})|}{\alpha}$, $\delta = \frac{2}{\alpha}\left(f(x_{k+1}^+) - f(x^*)\right)$ and $a = x_{k+1}^{++} - c_k$. The line search step of the algorithm implies that $\nabla f(x_{k+1})^\top (x_{k+1} - c_k) = 0$ and therefore, $|a| = |x_{k+1}^{++} - c_k| \geq |\nabla f(x_{k+1})|/\alpha = g$ and Lemma 1 applies to give the result. ■

Lemma 1 *Let $a \in \mathbb{R}^n$ and $\varepsilon \in (0,1), g \in \mathbb{R}_+$. Assume that $|a| \geq g$. Then there exists $c \in \mathbb{R}^n$ such that for any $\delta > 0$,*

$$\mathrm{B}(0, 1 - \varepsilon g^2 - \delta) \cap \mathrm{B}(a, g^2(1-\varepsilon) - \delta) \subset \mathrm{B}\left(c, 1 - \sqrt{\varepsilon} - \delta\right).$$

Proof First observe that if $g^2 \leq 1/2$ then one can take $c = a$ since $\frac{1}{2}(1-\varepsilon) \leq 1 - \sqrt{\varepsilon}$. Thus we assume now that $g^2 > 1/2$, and note that we can also clearly assume that $n = 2$. Consider the segment joining the two points at the intersection of the two balls under consideration. We denote c for the point at the intersection of this segment and $[0, a]$, and $x = |c|$ (that is $c = x\frac{a}{|a|}$). A simple picture reveals that x satisfies

$$1 - \varepsilon g^2 - \delta - x^2 = g^2(1-\varepsilon) - \delta - (|a| - x)^2 \quad \Leftrightarrow \quad x = \frac{1 + |a|^2 - g^2}{2|a|}.$$

When $x \leq |a|$, neither of the balls covers more than half of the other ball and hence the intersection of the two balls is contained in the ball $\mathrm{B}\left(x\frac{a}{|a|}, 1 - \varepsilon g^2 - \delta - x^2\right)$ (See figure 2). Thus it only remains to show that $x \leq |a|$ and that $1 - \varepsilon g^2 - \delta - x^2 \leq 1 - \sqrt{\varepsilon} - \delta$. The first

Algorithm 1: Minimum Enclosing Ball of the Intersection to Two Balls

Input: a ball centered at x_A with radius R_A and a ball centered at x_B with radius R_B.
if $|x_A - x_B|^2 \geq |R_A^2 - R_B^2|$ **then**
  $c = \frac{1}{2}(x_A + x_B) - \frac{R_A^2 - R_B^2}{2|x_A - x_B|^2}(x_A - X_B)$. $R^2 = R_B^2 - \frac{\left(|x_A - x_B|^2 + R_B^2 - R_A^2\right)^2}{4|x_A - x_B|^2}$.
else if $|x_A - x_B|^2 < R_A^2 - R_B^2$ **then**
  $c = x_B$. $R = R_B$.
else
  $c = x_A$. $R = R_A$.[a]
end
Output: a ball centered at c with radius R.

[a] If we assume $|x_A - x_B| \geq R_B$ as in Lemma 1, this extra case does not exist.

inequality is equivalent to $|a|^2 + g^2 \geq 1$ which follows from $|a|^2 \geq g^2 \geq 1/2$. The second inequality to prove can be written as

$$\varepsilon g^2 + \frac{(1 + |a|^2 - g^2)^2}{4|a|^2} \geq \sqrt{\varepsilon},$$

which is straightforward to verify (recall that $|a|^2 \geq g^2 \geq 1/2$). ■

Algorithm 2 we give is more agressive than Theorem 1, for instance, using line search instead of fixed step size. The correctness of this version follows from a similar proof as Theorem 1.

This algorithm does not require the smoothness parameter and the number of iterations; and it guarantees the function value is strictly decreasing. They are useful properties for machine learning applications because the only required parameter α is usually given. Furthermore, we believe that the integration of zeroth and first order information about the function makes our new method particularly well-suited in practice.

4 Experiments

In this section, we compare Geometric Descent method (GeoD) with a variety of full gradient methods. It includes steepest descent (SD), accelerated full gradient method (AFG), accelerated full gradient method with adaptive restart (AFGwR) and quasi-Newton with limited-memory BFGS updating (L-BFGS). For SD, we compute the gradient and perform an exact line search on the gradient direction. For AFG, we use the 'Constant Step Scheme II' in [Nesterov(2004)]. For AFGwR, [ODonoghue and Candes(2013)], we use the function restart scheme and replace the gradient step by an exact line search to improve its performance. For both AFG and AFGwR, the parameter is chosen among all powers of 2 for each dataset individually. For L-BFGS, we use the software developed by Mark Schmidt with default settings (see [Schmidt(2012)]).

In all experiments, the minimization problem is of the form $\sum_i \varphi(a_i^T x)$ where computing $a_i^T x$ is the computational bottleneck. Therefore, if we reuse the calculations carefully, each iteration of all mentioned methods requires only one calculation of $a_i^T x$ for some x. In particular, the cost

Algorithm 2: Geometric Descent Method (GeoD)

Input: parameters α and initial points x_0.

$x_0^+ = \texttt{line\_search}(x_0, x_0 - \nabla f(x_0))$.

$c_0 = x_0 - \alpha^{-1}\nabla f(x_0)$.

$R_0^2 = \frac{|\nabla f(x_0)|^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x_0) - f(x_0^+)\right)$.

for $i \leftarrow 1, 2, \cdots$ **do**

Combining Step:

$x_k = \texttt{line\_search}(x_{k-1}^+, c_{k-1})$.

Gradient Step:

$x_k^+ = \texttt{line\_search}(x_k, x_k - \nabla f(x_k))$.

Ellipsoid Step:

$x_A = x_k - \alpha^{-1}\nabla f(x_k)$. $R_A^2 = \frac{|\nabla f(x_k)|^2}{\alpha^2} - \frac{2}{\alpha}\left(f(x_k) - f(x_k^+)\right)$.

$x_B = c_{k-1}$. $R_B^2 = R_{k-1}^2 - \frac{2}{\alpha}\left(f(x_{k-1}^+) - f(x_k^+)\right)$.

Let $B(c_k, R_k^2)$ is the minimum enclosing ball of $B(x_A, R_A^2) \cap B(x_B, R_B^2)$.

end

Output: x_T.

of exact line searches is negligible compares with the cost of computing $a_i^T x$. Hence, we simply report the number of iterations in the following experiments.

4.1 Binary Classification

We evaluate the algorithms via the binary classification problem on the 40 datasets[1] from LIBSVM data, [Chang and Lin(2011)]. The problem is to minimize the regularized empirical risk:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} \varphi(b_i\, a_i^T x) + \frac{\lambda}{2}|x|^2$$

where $a_i \in \mathbb{R}^d$, $b_i \in \mathbb{R}$ are given by the datasets, λ is the regularization coefficient and φ is the smoothed hinge loss function given by

$$\varphi(z) = \begin{cases} 0 & \text{if } z \geq 1 \\ \frac{1}{2} - z & \text{if } z \leq 0 \\ \frac{1}{2}(1-z)^2 & \text{otherwise.} \end{cases}$$

We solve this problem with different regularization coefficients $\lambda \in \{10^{-4}, 10^{-5}, 10^{-6}, 10^{-7}, 10^{-8}\}$ and report the median and 90th percentile of the number of steps required to achieve a certain accuracy. In figure 3, we see that GeoD is better than SD, AFG and AFGwR, but worse than L-BFGS. Since L-BFGS stores and uses the gradients of the previous iterations, it is interesting to see if GeoD will be competitive to L-BFGS if it computes the intersection of multiple balls instead of 2 balls.

[1] We omitted all datasets of size $\geq$ 100 MB for time consideration.

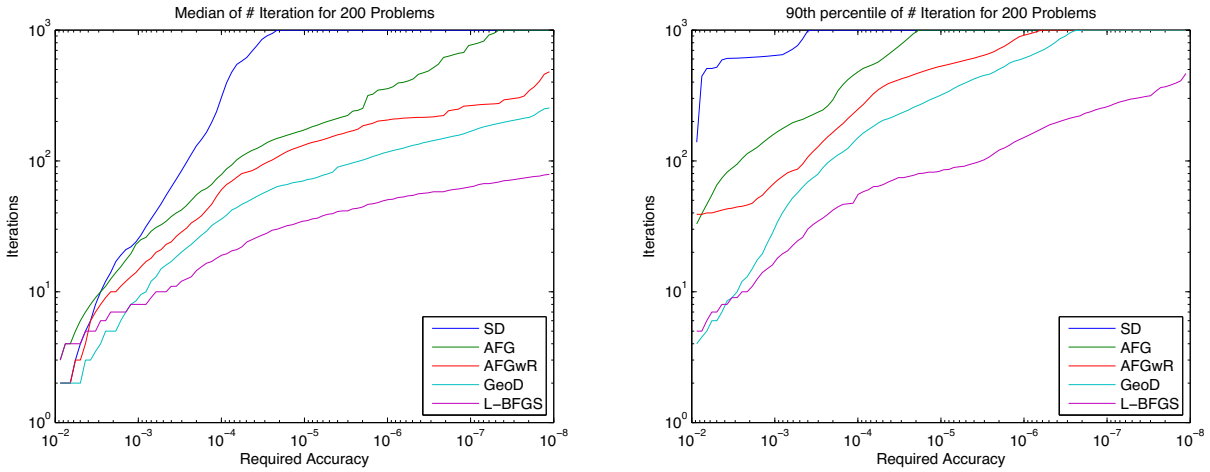

Figure 3: Comparison of full gradient methods on 40 datasets and 5 regularization coefficients with smoothed hinge loss function. The left diagram shows the median of the number of iterations required to achieve a certain accuracy and the right diagram shows the 90th percentile.

4.2 Worst Case Experiment

In this section, we consider the minimization problem

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

where β is the smothness parameter. Within the first n iterations, it is known that any iterative methods uses only the gradient information cannot minimize this function faster than the rate $1 - \Theta(\beta^{-1/2})$.

In figure 4, we see that every method except SD converge in the same rate with different constants for the first n iterations. However, after $\Theta(n)$ iterations, both SD and AFG continue to converge in the rate the theory predicted while other methods converge much faster. We remark that the memory size of L-BFGS we are using is 100 and if in the right example we choose $n = 100$ instead of 200, L-BFGS will converge at $n = 100$ immediately. It is surprising that the AFGwR and GeoD can achieve a comparable result by using "memory size" being 1.

References

[Allen-Zhu and Orecchia(2014)] Z. Allen-Zhu and L. Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *Arxiv preprint arXiv:1407.1537*, 2014.

[Bubeck(2014)] S. Bubeck. Theory of convex optimization for machine learning. *Arxiv preprint arXiv:1405.4980*, 2014.

[Chang and Lin(2011)] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST).*, 2(3):27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

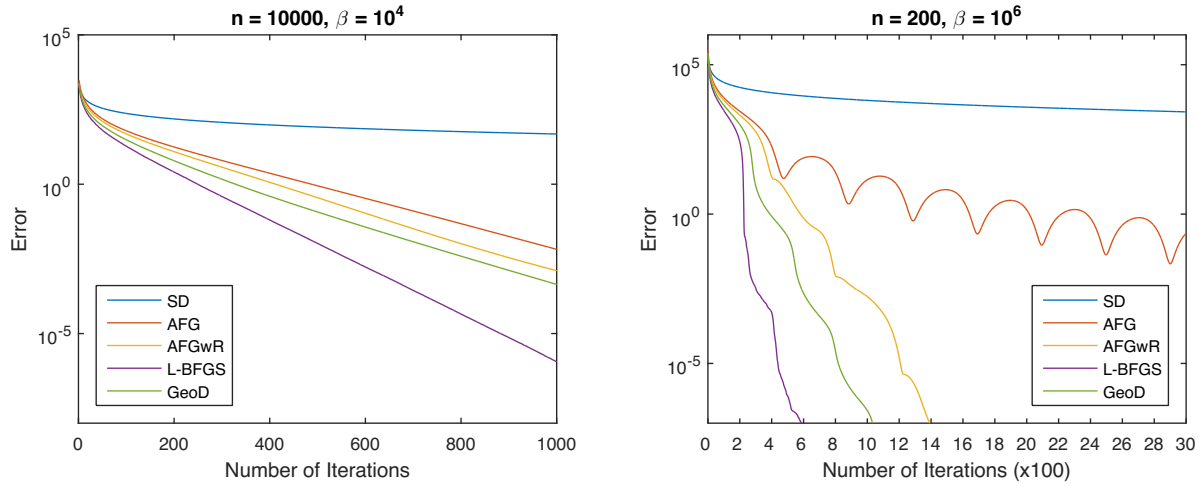

Figure 4: Comparision of full gradient methods for the function (2)

[Flammarion and Bach(2015)] N. Flammarion and F. Bach. From averaging to acceleration, there is only a step-size. In *Proceedings of the 28th Annual Conference on Learning Theory (COLT)*, 2015.

[Lessard et al.(2014)Lessard, Recht, and Packard] L. Lessard, B. Recht, and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *Arxiv preprint arXiv:1408.3595*, 2014.

[Nesterov(1983)] Y. Nesterov. A method of solving a convex programming problem with convergence rate o($1/k^2$). *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[Nesterov(2004)] Y. Nesterov. *Introductory lectures on convex optimization: A basic course*. Kluwer Academic Publishers, 2004.

[ODonoghue and Candes(2013)] Brendan ODonoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes. *Foundations of computational mathematics*, 15(3):715–732, 2013.

[Schmidt(2012)] M Schmidt. minfunc: unconstrained differentiable multivariate optimization in matlab. *URL http://www. di. ens. fr/mschmidt/Software/minFunc. html*, 2012.

[Su et al.(2014)Su, Boyd, and Candès] W. Su, S. Boyd, and E. Candès. A differential equation for modeling nesterovs accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.