ACCELERATED GRADIENT DESCENT USING BALLS

JAY BENDRE, JOHN DINH, AND COLLIN KENNEDY

Abstract. We take a deep dive into an geometric view of Nesterov's accelerated gradient descent (NAGD), which obtains the optimal rate of convergence among the class of α-strongly convex and β smooth functions. Bubeck et al. develops two algorithms, one with the convergence rate of standard gradient descent, and one that matches the rate of NAGD. After introducing the algorithm, we prove and provide geometric intuition as to why the algorithm achieves an accelerated rate of convergence.

1. Introduction

For our project, we sought to take a deep dive into understanding "A Geometric Alternative to Nesterov's Accelerated Gradient Descent", published by Sébastian Bubeck, Yin Tat Lee, and Mohit Singh in 2015 [BLS15]. But before we introduce the main points and theorems underlying the algorithm(s) they present, it is important to be familiar with some background information (Who's Nesterov? What is acceleration within the context of gradient descent? Etc...) to provide some context and motivation in order to clearly illustrate what problems the authors intend to solve with their work.

The gradient descent algorithm was invented by none other than the French mathematician Augustin-Louis Cauchy in 1847, in order to solve the unconstrained optimization problem:

$$\arg\min_{x\in\mathbb{R}^p} f(x)$$

To be brief, for a differentiable function f, the gradient descent algorithm solves the aforementioned optimization problem by taking an initial guess, x_0, and iteratively applying:

$$x_{t+1} = x_t - \gamma\nabla f(x_t), \quad t \geq 0$$

This is of course based on the observation that $f(x)$ decreases the fastest when one moves in the direction of the negative gradient of f at some x_t. When other conditions are met, such as when f is strongly convex and Lipschitz continuous, this means gradient descent can result in convergence to the *global* minimum. Perhaps most importantly, it has a time complexity of $\mathcal{O}\left(\log(\frac{1}{\kappa})\right)$. Now of course, there are plenty of issues with gradient descent. Putting its inability to distinguish between global and local minima aside, the most notable issue with gradient descent is how quickly the algorithm converges. Depending on how small or how large γ (the learning rate) is, the algorithm may either take a very long time to converge, or not converge at all.

In 1983, Russian mathematician Yurii Nesterov addressed some of these issues with his publishing of "A Method of Solving a Complex Programming Problem with Convergence Rate $\mathcal{O}\left(\frac{1}{k^2}\right)$"[Nes83]. As the title suggests, Nesterov was able to demonstrate that an iterative algorithm he developed could achieve a rate of convergence much faster than the standard gradient descent algorithm. Since the purpose of this project is to discuss an *alternative* to Nesterov's Accelerated Gradient Descent (NAGD) algorithm, I am going to avoid

delving into more complex details about the algorithm. With that said, NAGD essentially performs gradient descent, but at each iteration a *momentum* term takes the search a little farther, and incorporates a little correction just in case the minimum is overshot, which can occur when incorporating momentum.

This is of course, an oversimplification, and NAGD (and the underlying math) is notoriously difficult to grasp intuitively. Funny enough, this is a major part of the motivation for the geometric alternative proposed by Bubeck et al.: it is easier to understand, and in some cases is even superior to Nesterov's AGD.

The authors actually present two algorithms, a sub-optimal and accelerated (optimal) version of two shrinking balls. In the following sections, we provide a description of the algorithms and of the intuition behind them. We then discuss the main theorem (Theorem 1) they present to prove the rate of convergence of their optimal, accelerated algorithm and provide important details that relate to the proof.

2. Algorithm Description

2.1. **The Sub-Optimal Algorithm.** At a high level, the sub-optimal algorithm notes that x^* exists at the intersection of the small ball and the large ball (Fig. 1). Conceptually, we can then think that as the larger ball shrinks, it can be shown that the optimal solution x^* exists in a smaller (and iteratively smaller) ball with updated center x_t that shrinks by a factor of $1 - \frac{1}{\epsilon}$ at each iteration. The algorithm ultimately converges when the absolute distance between x_t and x_{t-1} falls below some threshold τ.

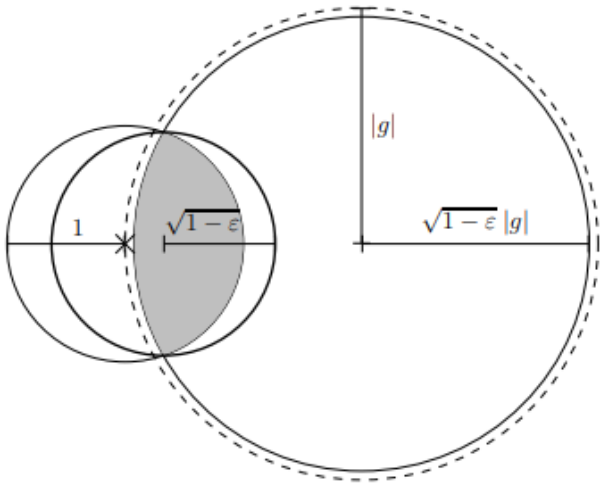

FIGURE 1

Algorithm 1: Sub-optimal algorithm describing shrinkage and convergence of the solution

function **Sub-Optimal Algorithm**;
Input : ($\mathrm{B}(x_0, R_0^2)$:= Initializer where we assume that our solution $x^* \in B(x_0, R_0^2)$ for given $R_0 > 0$, τ := Stopping criteria)
Output: x_t := Solution such that after t iterations $x_t \to x^*$
for $t \to iterations$ **do**
  if $\frac{||\nabla f(x)||^2}{\alpha^2} \leq R_0^2\left(1 - \frac{1}{\kappa}\right)$ **then**
    Shrink $\mathrm{B}(x_0, R_0^2)$ by $\left(1 - \frac{1}{\kappa}\right)$
  else
    $x_t = T(x_{t-1})$ where T is a map bsetween x_t & x_{t-1} such that
    $x^* \in \mathrm{B}\left(x_t, R_0^2\left(1 - \frac{1}{k}\right)\right)$;
    Update $R_0^2 = R_0^2\left(1 - \frac{1}{k}\right)$;
    if $||x_t - x_{t-1}||^2 \leq \tau$ **then**
      return (x_t)
    end
  end
end

2.2. **Optimal Algorithm.** We obtain the shrunken ball from the sub-optimal algorithm, and now introduce another ball that incorporates the information from x_0, in order to obtain another shrunken ball. That is, using the information from x_0, we obtain a new ball in the next iteration which contains the minimizer located in the intersection of these two balls(from the current iteration):

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

where the 1^{st} ball incorporates smoothness of f and the information from x_0 and the right ball is given in Eq. 3. The intersection of these two balls contains a radius given by the intersection will give us a convergence rate of $1 - \frac{1}{\sqrt{\kappa}}$, which is proven by Theorem 1.

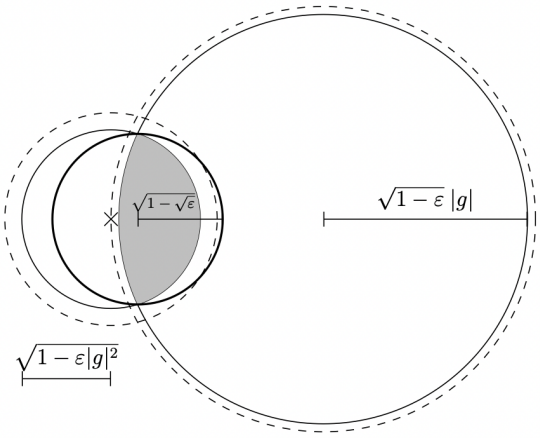

FIGURE 2. Two balls shrink

Similarly to the sub-optimal algorithm, the optimal algorithm notes that x^* exists at the intersection of the small ball and the large ball. The algorithm takes a starting value x_0, and performs a step of gradient descent to find the center of the small ball. To ensure this center is optimal (local minimizer), the algorithm finds the intersection between the small ball and the large ball, and finds a point within the region that then acts as the *new* center of the small circle at the next iteration. Because the intersecting balls both depend on the values being updated by the line search (specifically their radii), ,the region of intersection *shrinks* at a faster rate compared to the sub-optimal counterpart (by $1 - \frac{1}{\sqrt{\kappa}}$ as opposed to $1 - \frac{1}{\kappa}$ (See Fig. 2)

3. **Main Results**

When describing their optimal accelerated algorithm (See Figure 4 in Appendix A), the authors also present the following theorem:

Theorem 1. *For any $k \geq 0$, $\exists\ x^* \in B(c_k, R_k^2)$ and $R_{k+1}^2 \leq (1 - \frac{1}{\sqrt{\kappa}})R_k^2$ and thus:*

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

Theorem 1 is basically stating that for any given iteration k, we have that the optimal solution x^* is contained in a ball at center c_k with squared radius R_k^2, and that the squared radius at the next iteration (R_{k+1}^2) is less than or equal to the current radius R_k^2 by a factor of $1 - \frac{1}{\sqrt{\kappa}}$. The important takeaway here is that this factor $1 - \frac{1}{\sqrt{\kappa}}$ is *greater* than the factor $1 - \frac{1}{\kappa}$ from the sub-optimal algorithm, which of course results in the optimal algorithm having a faster rate of convergence than its sub-optimal counterpart.

Following the proof of Theorem 1, the authors then present some numerical evidence of a slightly modified version of the optimal algorithm reviewed in this summary, the Geometric Descent (GeoD) algorithm, and its performance and how it compares to a handful of other optimization algorithms, including steepest descent (SD), accelerated full gradient method (AFG), accelerated full gradient with adaptive restart (AFGwR), and quasi-newton with limited memory BFGS (L-BFGS) in two different scenarios. In the first scenario, the authors evaluate the performance of the algorithms on a standard binary classification problem using 40 different datasets from LIBSVM, an open-source machine learning library. They consider the minimization of the following empirical risk function:

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

where a_i, b_i depend on the dataset, λ is the regularization coefficient (the authors consider 5 different coefficients), and ϕ is the smooth hinge loss function. They find that GEoD performs better than SD, AFG, and AFGwR, but worse than L-BFGS. The authors note that it would be interesting to see if GeoD performas comparably (or even better than) L-BFGS if the GeoD algorithm considered an intersection of multiple balls as opposed to just two.

In the second scenario, the authors consider a "Worst Case Experiment." Here, they consider minimizing

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

where β is the smoothness parameter. They find that all the algorithms perform consistently with what theory predicts up until the first n iterations. Beyond $\Theta(n)$ iterations, Geometric Descent and L-BFGS converge at a much faster rate compared to the other algorithms. Another interesting thing to note is that although L-BFGS is the quicker algorithm, GeoD uses less memory.

4. **Proof Ideas**

Prior to introducing the main theorem the paper presents, the introduction of some preliminary notation is needed. Given that we are trying to minimize a function $f \in R^n$, denote x^* as the minimizer of f. Also, assume that f is α-strongly convex and β smooth. Denote $\kappa = \frac{\beta}{\alpha}$ as the condition number of f and $\mathrm{B}(x, r^2)$ as a Euclidean ball centered at x with radius, r^2. In addition, we define the following long and short gradient descent steps, respectively:

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

Under the assumption that f is strongly convex, we can rewrite the definition of a strongly convex function as the following:

$$\frac{\alpha}{2}\|y - x + \frac{1}{\alpha}\nabla f(x)\|^2 \leq \frac{\|\nabla f(x)\|^2}{2\alpha} - (f(x) - f(y)) \tag{1}$$

Under this definition, we obtain the enclosure of the minimzer of f, x^* in a ball with the following radius at x^{++}

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

Finally, under the assumption that f is β smooth, we have $f(x^+) \leq f(x) - \frac{1}{2\beta}\|\nabla f(x)\|^2$, which allows us to shrink the ball from above by a factor of $(1 - \frac{1}{\kappa})$. More intuitively, we are making the radius of the minimizer enclosing ball at x^{++} smaller by this factor:

$$x^* \in \mathrm{B}\left(x^{++}, \frac{\|\nabla f(x)\|^2}{\alpha^2}\left(1 - \frac{1}{\kappa}\right) - \frac{2}{\alpha}(f(x) - f(x^*))\right) \tag{3}$$

Now that we have the notion of this minimizer enclosing ball, the foundations behind the two algorithms Bubeck et al. present rely on this notion of iteratively updating the radius of two intersecting balls. Suppose that we have some x^* enclosing ball $A := B(x, r^2)$ from a previous iteration. Then, we can enclose x^* in some ball B from the intersection of A and the shrunken ball $\mathrm{B}\left(x^{++}, \frac{\|\nabla f(x)\|^2}{\alpha^2}(1 - \frac{1}{\kappa})\right)$ from Eq. 3. Now that we have this ball B with a smaller radius than A, we iteratively repeat this process by redefining A as this shrunken, minimizer enclosing ball B, giving us a iteration convergence rate with rate $1 - \frac{1}{\kappa}$, matching the rate for gradient descent, assuming α-strongly, β smooth functions. This will be the suboptimal algorithm Bubeck et al. introduces. In order to accelerate and achieve a

convergence rate of $1 - \frac{1}{\sqrt{\kappa}}$ we need to introduce one more concept, the line search mapping. It is an iterative method that finds the local minimum in multidimensional settings. It computes the search direction and the appropriate step size required for the algorithm to converge to the local minima. It is defined as:

$$\texttt{line\_search}(x, y) = \arg\min_{t \in R} f(x + t(y - x)) \tag{4}$$

where t is the step size required by the line search to update to a new value. In order to prove Theorem 1, proof by induction [Bub15] is used to prove the following claim:

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

The base case, $t = 0$ follows from Eq. 3. Then after assuming the t^{th} step is true, then using Eq. 3, we have the general minimizing enclosing ball at some $t + 1$ step, which uses the information from x:

$$x^* \in \mathrm{B}\left(x_{t+1}^{++}, \frac{\|\nabla f(x_{t+1})\|^2}{\alpha^2}\left(1 - \frac{1}{\kappa}\right)\right) \tag{5}$$

This will be our first shruken ball. For the second ball, we assume that

$$x^* \in \mathrm{B}\left(x_t^{++}, R_t^2 - \frac{\|\nabla f(x_{t+1})\|^2}{\alpha^2 \kappa}\right) \tag{6}$$

Lemma 1 implies that $\exists\ x'$ s.t. the intersection of ball 5 and ball 6 is encapsulated in some smaller ball. Using Eq. 3, our ball 5 becomes:

$$\mathrm{B}\left(x_{t+1}^{++}, \frac{\|\nabla f(x_{t+1})\|^2}{\alpha^2}\left(1 - \frac{1}{\kappa}\right) - \frac{2}{\alpha}\left(f(x_{t+1}) - f(x^*)\right)\right) \tag{7}$$

and ball 6 becomes:

$$\mathrm{B}\left(x_t^{++}, R_t^2 - \frac{\|\nabla f(x_{t+1})\|^2}{\alpha^2 \kappa} - \frac{2}{\alpha}\left(f(x_{t+1}^+) - f(x^*)\right)\right) \tag{8}$$

The key part to achieving an accelerated rate of convergence per iteration is dependent on the ball provided by the proof of Lemma 1. What this lemma is saying is that given we have the two balls above, there exists some other ball, which contains the intersection of these two balls with a related center and smaller radius, in the direction which decreases the gradient (given that we use the line search algorithm). Mathematically, Lemma 1 asserts that the radius of the ball that encloses the intersection of the two balls above is smaller than $1 - \frac{1}{\sqrt{\kappa}} R_t^2 - \frac{2}{\alpha}\left(f(x_{t+1}^+) - f(x^*)\right)$. Then, in order to ensure that the next step of the algorithm is as good as a step of gradient descent as the first step, the line search algorithm is applied.

5. Full Proof

Before we prove theorem 1, we need to show that the claim of the Lemma 1. is true.

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

$$B(0, 1 - \epsilon g^2 - \delta) \cap B(a, g^2(1 - \epsilon) - \delta) \subset B(c, 1 - \sqrt{\epsilon} - \delta)$$

Proof. To prove the above lemma, its important to note the difference when $g^2 \leq \frac{1}{2}$. On doing so, we can see that c = a. This can be easily understood by observing the radius at $g^2 = \frac{1}{2}$. On doing so, the radius of $B(0, 1 - \frac{1}{2}\epsilon - \delta)$ is always greater than $B(a, \frac{1}{2}(1-\epsilon) - \delta)$. For a ball to cover the intersection of above described balls, it needs to have the same center as the one with the smaller radius. Based on the lemma we can observe that the radius of $B(c, 1 - \sqrt{\epsilon} - \delta)$ is always larger than either of the radiuses mentioned before. Thus on setting $a = c$ with the given radius, we can always be certain that the intersection described in the lemma will always be a part of ball $B(a, 1 - \sqrt{\epsilon} - \delta)$.

Now, to prove for the case when $g^2 > \frac{1}{2}$, we would get a better understanding by looking at the Figure 3. Our main objective is to try and relate $x = |c|$ with all the given radii we know and try to establish a relationship between them.

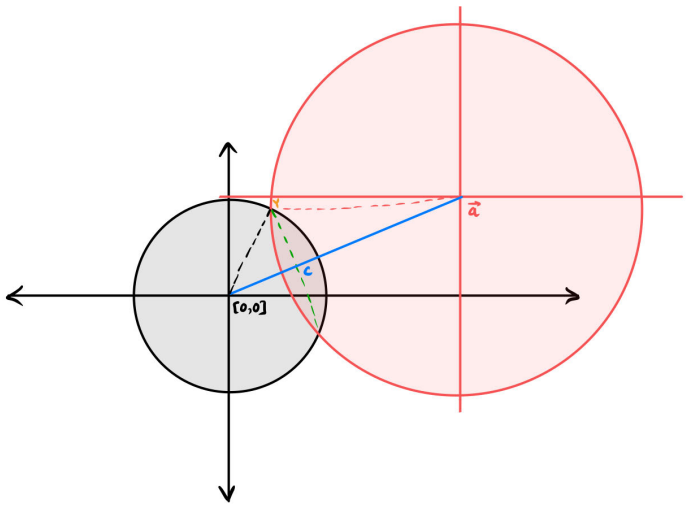

FIGURE 3. When $g^2 > \frac{1}{2}$

Consider the above setting, where c is the point of intersection between the vector $\overrightarrow{a}$ and the line segment joining the points of intersection between the two circles. From the diagram we can see that $\overrightarrow{c}$ is in the direction of ie. let $x = |c|$ then $\overrightarrow{c} = x\frac{\overrightarrow{a}}{|a|}$. Now consider ΔOCY and ΔYCA. In order to establish a relationship between these triangles, we use the common side between them CY. Using Pythagoras theorem, once we equate the sides for both triangles we get:

$$CY^2 = OY^2 - OC^2 = AY^2 - CA^2$$

We know that OY and AY are radii of the two circles and

$$OC = |C| \implies CA = OA - OC = |\overrightarrow{a}| - x$$

$$\implies 1 - \epsilon g^2 - \delta - x^2 = g^2(1-\epsilon) - \delta - (|\overrightarrow{a}| - x)^2$$

$$\implies x = \frac{1 + |\overrightarrow{a}|^2 - g^2}{2|\overrightarrow{a}|}$$

When $x \leq |\overrightarrow{a}|$, we can be sure of the fact that neither of the balls cover more than half of the other and hence, the intersection would lie in the ball $B(x\frac{\overrightarrow{a}}{|a|}, 1 - \epsilon g^2 - \delta - x^2)$ which is a smaller ball than proposed in the lemma $B(c, 1 - \sqrt{\epsilon} - \delta)$, hence we can be sure that the intersection would certainly lie in the ball proposed in lemma. □

Now that we have established this important lemma, the proof of Theorem 1 begins with induction of the following claim:

$$x^* \in \mathrm{B}\left(c_k, R_k^2 - \frac{2}{\alpha}\left(f(x_k^+) - f(x^*)\right)\right) \tag{9}$$

Proof. To establish the base case, recall the smoothness condition of f, which allows us to establish:

$$f(x^*) \leq f(x_{k+1}^+) \leq f(x_{k+1}) - \frac{1}{2\beta}\|\nabla f(x_{k+1})\|^2 \leq f(x_k^+) - \frac{1}{2\beta}\|\nabla f(x_{k+1})\|^2 \tag{10}$$

For the base case, let $k = 0$, then it immediately follows from Eq. 3, that we have:

$$x^* \in \mathrm{B}\left(c_0, R_0^2 - \frac{2}{\alpha}\left(f(x^+) - f(x^*)\right)\right)$$

where $c_0 = x_0^{++}$. Now we assume that the k^{th} iteration is true, that is we have the two following balls, which come from the smoothness conditions from Eq. 10:

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

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

Now, under the $(k+1)^{th}$ case, we have the following new smoothness conditions that we will construct our new balls with centers c_{k+1} and x_{k+2}^{++}, respectively.

$$f(x^*) \leq f(x_{k+2}^+) \leq f(x_{k+2}) - \frac{1}{2\beta}\|\nabla f(x_{k+2})\|^2 \leq f(x_{k+2}^+) - \frac{1}{2\beta}\|\nabla f(x_{k+2})\|^2 \tag{11}$$

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

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

Thus by induction, we can confirm the claim made in Eq. 9. Now using the Lemma 1, we try to find the intersection between the above two balls mentioned for the k^{th} iteration. From the lemma, if we make the following changes

$$g = \frac{\|\nabla f(x_{k+1})\|}{\alpha}, \delta = \frac{2}{\alpha}(f(x_{k+1}^+) - f(x^*)), \epsilon = \frac{1}{\kappa} \text{ and } ,a = x_{k+1}^{++} - c_k$$

Using these parameters in the Lemma 1 and based on the fact that from line search we get $\nabla f(x_{k+1})^\top (x_{k+1} - c_k) = 0$ which in turn implies that the distance between the centers of both the balls would be always greater than the shift of x_{k+1} ie. $|x_{k+1}^{++} - c_k| \geq |\nabla f(x_{k+1})|/\alpha$. On applying the lemma, we'd get the rate of convergence as $\sqrt{\epsilon} = \frac{1}{\sqrt{\kappa}}$

□

References

[BLS15] Sébastien Bubeck, Yin Tat Lee, and Mohit Singh, *A geometric alternative to nesterov's accelerated gradient descent*, 2015.

[Bub15] Sébastien Bubeck, *Convex optimization: Algorithms and complexity*, Nov 2015.

[Nes83] Yurii Nesterov, *A method of solving a complex programming problem with convergence rate $o(1/k^2)$*, 1983.

Appendix A. Additional Details of the Proofs

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.

Figure 4

Email address: jdbendre@ucdavis.edu

Email address: jndinh@ucdavis.edu

Email address: cjkennedy@ucdavis.edu