CS 435, 2018
Lecture 5, Date: 22 March 2018
Instructor: Nisheeth Vishnoi

Nesterov's Accelerated Gradient Descent

In this lecture, we derive the Accelerated Gradient Descent algorithm whose convergence rate is $O(\varepsilon^{-1/2})$ which improves upon $O(\varepsilon^{-1})$ – achieved by the standard gradient descent.

Contents

1 Nesterov's Accelerated Gradient Descent

1.1 Setting

We would like to solve an unconstrained optimization problem of the form

$$\min_{x \in \mathbb{R}^n} f(x),$$

where f is a convex, L-smooth[1] function with respect to a norm $\|\cdot\|$. In other words, it holds that

$$\forall x, y \in \mathbb{R}^n \quad f(y) \leq f(x) + \langle y - x, \nabla f(x) \rangle + \frac{L}{2} \|x - y\|^2 .$$

We also let $R : \mathbb{R}^n \to \mathbb{R}$ be a α-strongly convex regularizer with respect to a norm $\|\cdot\|$, i.e.,

$$D_R(x, y) := R(x) - R(y) - \langle \nabla R(y), x - y \rangle \geq \frac{\alpha}{2} \|x - y\|^2 .$$

Recall that the above defined (pseudo)-distance function $D_R(x, y)$ is called the Bregman divergence of R. In this lecture we consider only regularizers for which the map $\nabla R : \mathbb{R}^n \to \mathbb{R}^n$ is bijective.

When reading these notes it might be beneficial to keep in mind the special case of $R(x) := \frac{1}{2} \|x\|^2$. Then $D_R(x, y) = \frac{1}{2} \|x - y\|^2$ and ∇R is the identity map.

So far we have learned (Lecture 3), that there is an algorithm for optimizing L-smooth functions, performing roughly $O(\varepsilon^{-1})$ iterations. The goal of this lecture is to improve it to $O(\varepsilon^{-1/2})$ – an optimal algorithm in the black-box model.

1.2 Main Theorem

Theorem 1. *There is an algorithm, which given:*

- 1*st-order oracle access to a convex function* $f : \mathbb{R}^n \to \mathbb{R}$,
- *a number* L *such that* f *is* L*-smooth with respect to a norm* $\|\cdot\|$,
- *oracle access to the gradient map* ∇R *and its inverse map* $(\nabla R)^{-1}$ *for a regularizer* $R : \mathbb{R}^n \to \mathbb{R}$,
- *bound on strong convexity parameter* $\alpha > 0$ *of* R *with respect to* $\|\cdot\|$,
- *an initial point* $x_0 \in \mathbb{R}^n$ *such that* $D_R(x, x_0) \leq D^2$ *(where* $x^\star$ *is an optimal solution to* $\min_{x \in \mathbb{R}^n} f(x)$*)*,
- *an* $\varepsilon > 0$,

[1] The notions of L-smoothness and having L-Lipschitz gradient are equivalent for the ℓ_2 norm.

outputs a point $x \in \mathbb{R}^n$ *such that* $f(x) \leq f(x^\star) + \varepsilon$. *The algorithm makes* $T = O\left(\sqrt{\frac{LD^2}{\alpha\varepsilon}}\right)$ *queries to the respective oracles and performs* $O(nT)$ *arithmetic operations.*

Note that the Theorem we proved in Lecture 3 achieved $O\left(\frac{LD^2}{\alpha\varepsilon}\right)$ iterations – exactly the square of what the above theorem achieves! This speed-up over the gradient descent algorithm was derived by Nesterov in [3] (see also [4]). This idea of acceleration was then extended to numerous other variants of gradient descent and led to introduction of highly efficient algorithms such as FISTA [2].

1.3 Proof Strategy - Estimate Sequences

In our proof of Theorem 1 instead of first stating the algorithm and then proving its properties we instead proceed in the opposite order. We first formulate an important theorem asserting existence of a so-called estimate sequence. In the process of proving this theorem we derive – step by step – the Accelerated Gradient Descent algorithm, which then turns out to imply Theorem 1.

A crucial notion used in deriving the Accelerated Gradient Descent algorithm is an estimate sequence (see [4]).

Definition 2. *A sequence* $(\phi_t, \lambda_t, x_t)_{t\in\mathbb{N}}$, *where* $\phi_t : \mathbb{R}^n \to \mathbb{R}$, $\lambda_t \in [0,1]$ *and* $x_t \in \mathbb{R}^n$ *(for all* $t \in \mathbb{N}$*) is said to be an estimate sequence for a function* $f : R^n \to \mathbb{R}$ *if it satisfies the following properties.*

(1) **(Lower bound)** *For all* $t \in \mathbb{N}$ *and for all* $x \in \mathbb{R}^n$, $\phi_t(x) \leq (1-\lambda_t)f(x) + \lambda_t\phi_0(x)$.

(2) **(Upper bound)** *For all* $x \in \mathbb{R}^n$, $f(x_t) \leq \phi_t(x)$.

Intuitively we should think of the sequence $(x_t)_{t\in\mathbb{N}}$ as converging to a minimizer of f. The functions $(\phi_t)_{t\in\mathbb{N}}$ serve as approximations to f, which provide tighter and tighter (as t increases) bounds on the gap $f(x_t) - f(x^\star)$. More precisely, condition (1) says that $\phi_t(x)$ is an approximate lower bound to $f(x)$ and condition (2) says that the minimum value of ϕ_t is above $f(x_t)$.

To illustrate this definition, suppose for a moment that $\lambda_t = 0$ for some $t \in \mathbb{N}$ in the estimate sequence. Then, from conditions (2) and (1) we obtain

$$f(x_t) \leq \phi_t(x^\star) \leq f(x^\star).$$

This implies that x_t is an optimal solution. Thus, while hoping that $\lambda_t = 0$ may be too ambitious, $\lambda_t \to 0$ is what we will achieve. In fact, Nesterov's method constructs a sequence λ_t which goes to zero as $\frac{1}{t^2}$, a quadratic speed-up over the standard gradient descent method. Formally, we prove the following theorem.

Theorem 3. *For every convex, L-smooth (with respect to* $\|\cdot\|$*) function* $f : \mathbb{R}^n \to \mathbb{R}$, *for every* α*-strongly convex regularizer R (with respect to the same norm* $\|\cdot\|$*), and for every* $x_0 \in \mathbb{R}^n$, *there*

exists an estimate sequence $(\phi_t, \lambda_t, x_t)_{t\in\mathbb{N}}$ *with* $\phi_0(x) := f(x_0) + \frac{L}{2\alpha} D_R(x, x_0)$ *and* $\lambda_t \leq \frac{c}{t^2}$ *for some absolute constant* $c > 0$.

Suppose now that $D_R(x^\star, x_0) \leq D^2$. Then, what we obtain for such a sequence, using conditions (2) and (1) with $x = x^\star$, is

$$f(x_t) \overset{\text{(Lower Bound)}}{\leq} \phi_t(x^\star) \overset{\text{(Upper Bound)}}{\leq} (1-\lambda_t) f(x^\star) + \lambda_t \frac{L}{2\alpha} D_R(x^\star, x) \leq f(x^\star) + \frac{cLD^2}{\alpha t^2}. \quad (1)$$

Thus, it is enough to take $t \approx \sqrt{\frac{LD^2}{\varepsilon}}$ to make sure that $f(x_t) - f(x^\star) \leq \varepsilon$. We cannot yet deduce Theorem 1 from Theorem 3, as in the form as stated it is not algorithmic – we need to know that such a sequence can be efficiently computed using a 1st order oracle to f and R only, while Theorem 3 only claims existence. However, as we will see, the proof of Theorem 3 provides an efficient algorithm to compute estimate sequences.

1.4 Construction of an Estimate Sequence

This section is devoted to prove Theorem 3. To start, we make a simplifying assumption that f has 1-Lipschitz gradient. This can be obtained by considering $\frac{f}{L}$ instead of f, where L is the Lipschitz constant of the gradient of f. Similarly we might assume that R is 1-strongly convex, scaling R by α if necessary.

1.4.1 Step 1. Plan – Iterative Construction

The construction of the estimate sequence is iterative. Let $x_0 \in \mathbb{R}^n$ be an arbitrary point. We set

$$\phi_0(x) := D_R(x, x_0) + f(x_0), \qquad \lambda_0 = 1.$$

Thus, the first condition in the definition is trivially satisfied. For the second, note that

$$\min_x \phi_0(x) = f(x_0).$$

The construction of subsequent elements of the estimate sequence is inductive. Suppose we are given $(\phi_{t-1}, x_{t-1}, \lambda_{t-1})$. Then ϕ_t will be a convex combination of ϕ_{t-1} and the linear lower bound L_{t-1} to f at a carefully chosen point $y_{t-1} \in \mathbb{R}^n$ (to be defined later). More precisely, we set

$$L_{t-1}(x) := f(y_{t-1}) + \langle x - y_{t-1}, \nabla f(y_{t-1}) \rangle .$$

Note that by the 1st order convexity criterion, $L_{t-1}(x) \leq f(x)$ for all $x \in \mathbb{R}^n$. We set the new estimate to be

$$\phi_t(x) := (1-\gamma_t)\phi_{t-1}(x) + \gamma_t L_{t-1}(x), \quad (2)$$

where $\gamma_t \in [0, 1]$ will be determined later.

In the next steps of the proof we use the above scheme to prove conditions (1) and (2) of the estimate sequence. This is proved by induction, i.e., when proving the claim for $t \in \mathbb{N}$ we assume that it holds for $(t-1)$. On the way we state several constraints on x_t,

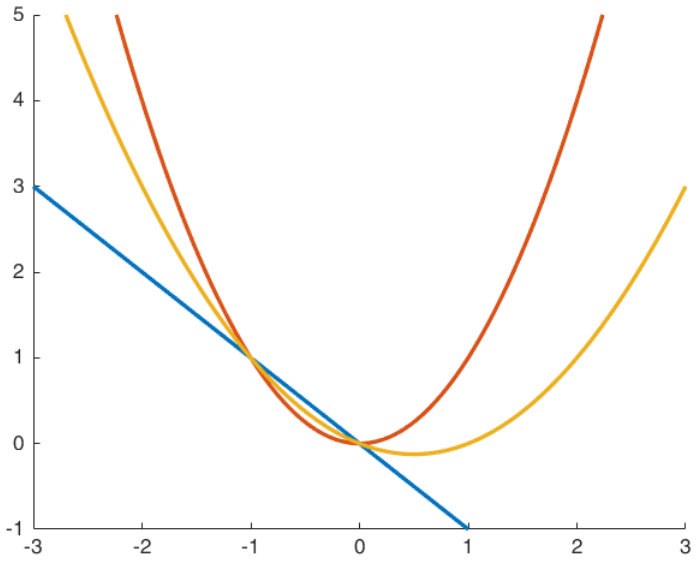

Figure 1: An illustration of combining a quadratic $g_1(x) = x^2$ (in red) with a linear function $l_1(x) = -x$ as a convex combination $g_2(x) = \frac{1}{2}g_1(x) + \frac{1}{2}l_1(x)$ (in yellow). The new parabola (for g_2) shifts slightly to the right and gets "wider" as compared to g_1.

y_t, γ_t and λ_t which are necessary for our proofs to work. At the final stage of the proof we collect all these **constraints** and show that they are **not contradictory** and thus there is a way to set these parameters in order to obtain a valid estimate sequence.

1.4.2 Discussion of Estimate Sequences

To gain some intuitions about estimate sequences and the sequence of ϕ_t's in particular, let us now consider the simplest of cases: when $n = 1$ and $R(x) := \frac{1}{2}x^2$.

Observe first that according to the update rule (2), the estimate $\phi_t(x)$ is always of the form

$$\phi_t(x) = \phi_t^\star + \lambda_t(x - z_t)^2,$$

where $\lambda_t = \prod_{i=1}^{t}(1 - \gamma_i)$, $\phi_t^\star$ is the minimum value of ϕ_t and z_t is the minimizer of ϕ_t. For an illustration of what happens when one makes a step from ϕ_t to ϕ_{t+1} consider Figure 1. The way the linear function L_{t-1} is picked is to shift the minimum of ϕ_{t-1} towards the minimum of f. Moreover, as $\lambda_t \to 0$, the parabola ϕ_t becomes wider and wider – this is somewhat necessary as we explain below.

Note that by the fact that f is 1-Lipschitz it follows that $\phi_0(x) = \frac{1}{2}(x - x_0)^2$ is a global upper bound on f. Consequently, the function $f_t(x) := (1 - \lambda_t)f(x) + \lambda_t\phi_0(x)$ is always a

convex function "in-between" $f(x)$ and $\phi_0(x)$. As $\lambda_t \to 0$, f_t is a better and better estimate of f. The condition (1) states that $\phi_t(x)$ should be a global lower bound for $f_t(x)$.

Consider now a case when $f(x) \equiv c$ is a constant function (or alternatively the case when the slope of f changes very slowly – i.e. $|f''(x)|$ is very small). Then for large t, f_t approximates f rather well, so ϕ_t to be a lower bound on f_t, must be a very wide parabola.

Before we proceed to Step 2. of the proof, let us generalize what we have observed above for the simple case of quadratic function to more general regularizers R. We would like to show that if by shifting a Bregman divergence term of the form $x \mapsto D_R(x, z)$ by a linear function $\langle l, x\rangle$ we again obtain a Bregman divergence, from a possibly different point z.

Fact 1. *Let $z \in \mathbb{R}^n$ be an arbitrary point and $R : \mathbb{R}^n \to \mathbb{R}$ be a convex regularizer for which $\nabla R : \mathbb{R}^n \to \mathbb{R}^n$ is a bijection. Then, for every $l \in \mathbb{R}^n$ there exists $z' \in \mathbb{R}^n$ such that*

$$\forall x \in \mathbb{R}^n \quad D_R(x, z) + \langle l, x\rangle = D_R(x, z').$$

Morever, z' is uniquely determined by the following relation

$$\nabla R(z') = \nabla R(z) - l.$$

The proof of the above is a simple exercise. It follows now that by denoting the minimum value of ϕ_t by $\phi_t^\star$ and denoting by z_t the global minimum of ϕ_t (i.e. $\phi_t^\star = \phi_t(z_t)$) we have

$$\phi_t(x) = \phi_t^\star + \lambda_t D_R(x, z_t). \tag{3}$$

Moreover, z_t satisfies the following recurrence

$$\forall t \geq 1 \quad \nabla R(z_t) = \nabla R(z_{t-1}) - \frac{\gamma_t}{\lambda_t} \nabla f(y_{t-1}).$$

1.4.3 Step 2. Ensuring Condition (1)

We would like to ensure that

$$\phi_t(x) \leq (1 - \lambda_t) f(x) + \lambda_t \phi_0(x).$$

From our inductive construction we have

$$\begin{aligned}
\phi_t(x) &= (1 - \gamma_t)\phi_{t-1}(x) + \gamma_t L_{t-1}(x) \\
&\quad \text{(by def. of } \phi_t\text{)} \\
&\leq (1 - \gamma_t)[(1 - \lambda_{t-1}) f(x) + \lambda_{t-1}\phi_0(x)] + \gamma_t L_{t-1}(x) \\
&\quad \text{(by the induction hypothesis for } (t-1)\text{)} \\
&\leq (1 - \gamma_t)[(1 - \lambda_{t-1}) f(x) + \lambda_{t-1}\phi_0(x)] + \gamma_t f(x) \\
&\quad \text{(by } L_{t-1}(x) \leq f(x)\text{)} \\
&\leq ((1 - \gamma_t)(1 - \lambda_{t-1}) + \gamma_t) f(x) + (1 - \gamma_t)\lambda_{t-1}\phi_0(x). \\
&\quad \text{(rearranging)}
\end{aligned} \tag{4}$$

Now, by setting

$$\lambda_t := (1-\gamma_t)\lambda_{t-1}, \tag{5}$$

we obtain that

$$\phi_t(x) \leq (1-\lambda_t)f(x) + \lambda_t\phi_0(x).$$

Thus, as long as (5) holds, we obtain condition (1). Note also that a different way to state (5) is also that $\lambda_t = \prod_{1\leq i\leq t}(1-\gamma_t)$.

1.4.4 Step 3. Ensuring Condition (2): Dynamics of y_t

To satisfy Condition (2) our goal is to set x_t in such a way that

$$f(x_t) \leq \min_{x\in\mathbb{R}^n} \phi_t(x) = \phi_t^\star = \phi_t(z_t).$$

Note that this in particular requires us to specify y_{t-1}, as the right-hand side depends on y_{t-1}. Towards this, consider any $x \in \mathbb{R}^n$

$$\begin{aligned}
\phi_t(x) &= (1-\gamma_t)\phi_{t-1}(x) + \gamma_t L_{t-1}(x) \\
&\quad \text{(by def. of } \phi_t\text{)} \\
&= (1-\gamma_t)(\phi_{t-1}(z_{t-1}) + \lambda_{t-1}D_R(x, z_{t-1})) + \gamma_t L_{t-1}(x) \\
&\quad \text{(by (3)} \\
&= (1-\gamma_t)(\phi_{t-1}(z_{t-1}) + \lambda_{t-1}D_R(x, z_{t-1})) + \gamma_t(f(y_{t-1}) + \langle x - y_{t-1}, \nabla f(y_{t-1})\rangle) \\
&\quad \text{(by def. of } L_{t-1}\text{)} \\
&\geq (1-\gamma_t)f(x_{t-1}) + \lambda_t D_R(x, z_{t-1}) + \gamma_t(f(y_{t-1}) + \langle x - y_{t-1}, \nabla f(y_{t-1})\rangle) \\
&\quad \text{(by condition (2) for } \phi_{t-1}\text{)} \\
&\geq (1-\gamma_t)(f(y_{t-1}) + \langle x_{t-1} - y_{t-1}, \nabla f(y_{t-1})\rangle + \gamma_t(f(y_{t-1}) + \langle x - y_{t-1}, \nabla f(y_{t-1})\rangle) + \lambda_t D_R(x, z_{t-1}) \\
&\quad \text{(by convexity of } f\text{)} \\
&= f(y_{t-1}) + \langle (1-\gamma_t)(x_{t-1} - y_{t-1}) + \gamma_t(x - y_{t-1}), \nabla f(y_{t-1})\rangle + \lambda_t D_R(x, z_{t-1}) \\
&\quad \text{(rearranging)} \\
&= f(y_{t-1}) + \langle (1-\gamma_t)x_{t-1} + \gamma_t z_{t-1} - y_{t-1}, \nabla f(y_{t-1})\rangle + \gamma_t \langle x - z_{t-1}, \nabla f(y_{t-1})\rangle + \lambda_t D_R(x, z_{t-1}) \\
&\quad \text{(by adding and subtracting } \gamma_t \langle x - z_{t-1}, \nabla f(y_{t-1})\rangle \text{ and rearranging)} \\
&= f(y_{t-1}) + \gamma_t \langle x - z_{t-1}, \nabla f(y_{t-1})\rangle + \lambda_t D_R(x, z_{t-1})
\end{aligned}$$

Where the last equality follows by setting

$$y_{t-1} := (1-\gamma_t)x_{t-1} + \gamma_t z_{t-1}.$$

To give some intuition why do we make such a choice for y_{t-1} consider for a second the case when $R(x) = \|x\|_2^2$ and $D_R(x, y)$ is the squared Euclidean distance. In the above, we

would like to obtain a term which looks like a 2nd order (quadratic) upper-bound on $f(\widetilde{x})$ around $f(y_{t-1})$, i.e.,

$$f(y_{t-1}) + \langle \widetilde{x} - y_{t-1}, \nabla f(y_{t-1}) \rangle + \frac{1}{2} \|\widetilde{x} - y_{t-1}\|^2 . \tag{6}$$

Such a choice of y_{t-1} allows us to cancel out an undesired linear term. We do not quite succeed in getting the form as in (6) – our expression has z_{t-1} instead of y_{t-1} in several places and has additional constants in front of the linear and quadratic term. We deal with these issues in the next step and make our choice of x_t accordingly.

1.4.5 Step 4. Ensuring Condition (2): Dynamics of x_t

We continue the derivation from Step 3. departing from where we have stopped:

$$\begin{aligned}
\phi_t(x) &\geq f(y_{t-1}) + \gamma_t \langle x - z_{t-1}, \nabla f(y_{t-1}) \rangle + \lambda_t D_R(x, z_{t-1}) \\
&\quad \text{(as established in Step 3.)} \\
&\geq f(y_{t-1}) + \gamma_t \langle x - z_{t-1}, \nabla f(y_{t-1}) \rangle + \frac{\lambda_t}{2} \|x - z_{t-1}\|^2 \\
&\quad \text{(by 1-strong convexity of } R\text{)} \\
&= f(y_{t-1}) + \langle \widetilde{x} - y_{t-1}, \nabla f(y_{t-1}) \rangle + \frac{\lambda_t}{2\gamma_t^2} \|\widetilde{x} - y_{t-1}\|^2 \\
&\quad \text{(by a change of variables: } \widetilde{x} - y_{t-1} := \gamma_t(x - z_{t-1})\text{)} \\
&\geq f(y_{t-1}) + \langle \widetilde{x} - y_{t-1}, \nabla f(y_{t-1}) \rangle + \frac{1}{2} \|\widetilde{x} - y_{t-1}\|^2 \\
&\quad \text{(assuming } \tfrac{\lambda_t}{\gamma_t^2} \geq 1\text{)} \\
&\geq f(x_t)
\end{aligned}$$

Where the last step follows by setting

$$x_t := y_{t-1} - \nabla f(y_{t-1}),$$

which is the minimizer of the RHS over all $\widetilde{x}$.

The reason for such a renaming of variables and introducing $\widetilde{x}$ follows the same intution as the choice of y_{t-1} in Step 4. We would like to arrive at an expression which is a quadratic upper bound on f around y_{t-1}, evaluated at a point $\widetilde{x}$. Such a choice of $\widetilde{x}$ allows us to cancel the γ_t in front of the linear part of this upper bound and hence to obtain the desired expression assuming that $\frac{\lambda_t}{\gamma_t^2} \geq 1$. As we will see later, this constraint really determines the convergence rate of the resulting method. Finally, the choice of x_t follows straightforwardly: we simply pick a point which minimizes this quadratic upper bound on f over $\widetilde{x}$.

1.4.6 Step 5. The Algorithm and Proof of the Main Theorem

Let us now collect all the constraints and summarize the updates rules according to which, new points are obtained. Initially, $x_0 \in \mathbb{R}^n$ is arbitrary, $z_0 = x_0$, $\gamma_0 = 0$ and $\lambda_0 = 1$. Further, for $t \geq 1$ we have $\lambda_t := \prod_{1 \leq i \leq t}(1 - \gamma_i)$ and

$$\begin{aligned} y_{t-1} &:= (1 - \gamma_t)x_{t-1} + \gamma_t z_{t-1}, \\ \nabla R(z_t) &:= \nabla R(z_{t-1}) - \frac{\gamma_t}{\lambda_t}\nabla f(y_{t-1}), \\ x_t &:= y_{t-1} - \nabla f(y_{t-1}). \end{aligned} \tag{7}$$

Note that y_t is just a "coupling" of the sequences (x_t) and (z_t), which follow two different optimization primitives:

1. The update rule to obtain x_t is simply to perform one step of gradient descent starting from y_{t-1}.
2. The sequnce z_t is just peforming mirror descent with respect to R, taking gradients at y_{t-1}.

Thus Nesterov's accelerated gradient descent is simply a result of coupling gradient descent with mirror descent. This viewpoint has been investigated in-depth in [1]. Note that, in particular, if we choose $\gamma_t = 0$ for all t then y_t is simply the gradient descent method from Lecture 3, and we choose $\gamma_t = 1$ for all t then y_t is mirror descent (introduced in Lecture 4).

Pictorially, in algorithm (7), the parameters are fixed in the following order.

$$\begin{array}{ccccccccc} x_0 & & x_1 & & & & & & x_t \\ & \to y_0 \nearrow\searrow & & \to y_1 \nearrow\searrow & \cdots\cdots \to & y_{t-1} & \nearrow\searrow & & \\ z_0 & & z_1 & & & & & & z_t \end{array}$$

The proof of Theorem 1 follows simply from the construction of the estimate sequence. The only remaining piece which we have not established yet is that one can take $\gamma_t \approx \frac{1}{t}$ and $\lambda_t \approx \frac{1}{t^2}$ – this follows from a straightfroward calculation and is proved in the next section.

From the update rules in (7) one can easily see that one can keep track of x_t, y_t, z_t by using a constant number of oracle queries to ∇f, ∇R and $(\nabla R)^{-1}$. Furthermore, as already demonstrated in (1), we have

$$f(x_t) \leq f(x^\star) + O\left(\frac{LD^2}{\alpha t^2}\right).$$

Thus to attain $f(x_t) \leq f(x^\star) + \varepsilon$, taking $t = O\left(\sqrt{\frac{LD^2}{\alpha\varepsilon}}\right)$ is sufficient.

1.4.7 Step 6. Choice of γ_t's

When deriving the estimate sequence we have made an assumption that

$$\lambda_t \geq \gamma_t^2,$$

which does not allow us to set γ_t's arbitrarily. The following lemma provides an example setting of γ_t's which satisfy this constraint.

Lemma 4. *Let $\gamma_0 = \gamma_1 = \gamma_2 = \gamma_3 = 0$ and $\gamma_i = \frac{2}{i}$ for all $i \geq 4$, then*

$$\forall t \geq 0 \qquad \prod_{i=1}^{t}(1-\gamma_i) \geq \gamma_t^2.$$

Proof. For $t \leq 4$ one can verify the claim directly. Let $t > 4$. We have

$$\prod_{i=1}^{t}(1-\gamma_i) = \frac{2}{4}\cdot\frac{3}{5}\cdot\frac{4}{6}\cdot\ldots\cdot\frac{t-2}{t} = \frac{2\cdot 3}{(t-1)t},$$

by cancelling all but 2 terms in the numerator and denominator. It is now easy to verify that $\frac{6}{t(t-1)} \geq \frac{4}{t^2} = \gamma_t^2$. □

1.5 An algorithm for strongly convex and smooth functions

From Theorem 1 one can derive numerous other algorithms. Here, for instance we deduce a method for minimizing functions which are both L-smooth and α-strongly convex (with respect to the Euclidean norm) at the same time, i.e., it satisfies

$$\forall x, y \in \mathbb{R}^n \qquad \frac{\alpha}{2}\|x-y\|_2^2 \leq f(x) - f(y) - \langle x-y, \nabla f(y)\rangle \leq \frac{L}{2}\|x-y\|_2^2.$$

Theorem 5. *There is an algorithm that for any function $f : \mathbb{R}^n \to \mathbb{R}$, which is both α-strongly convex and L-smooth with respect to the Euclidean norm, given*

- *1st-order oracle access to a convex function f,*
- *the numbers $L, \alpha \in R$,*
- *an initial point $x_0 \in \mathbb{R}^n$ such that $\|x^\star - x_0\|_2^2 \leq D^2$ (where $x^\star$ is an optimal solution to $\min_{x\in\mathbb{R}^n} f(x)$),*
- *an $\varepsilon > 0$,*

outputs a point $x \in \mathbb{R}^n$ such that $f(x) \leq f(x^\star) + \varepsilon$. The algorithm makes $T = O\left(\sqrt{\frac{L}{\alpha}}\log\frac{LD^2}{\varepsilon}\right)$ queries to the respective oracles and performs $O(nT)$ arithmetic operations.

Note that, importantly, the above method has **logarithmic** dependency on $\frac{1}{\varepsilon}$ which we have not seen so far in this course! Indeed, having both L-smoothness and α-strong convexity is enough to construct such efficient algorithms, however for such an algorithm to be efficient one still has to make sure that the "condition number" of the function: $\frac{L}{\alpha}$, is small.

Proof. Consider $R(x) = \|x\|_2^2$ as our choice of a regularizer in Theorem 1, note that R is 1-strongly convex. Thus, the algorithm in Theorem 1 constructs a sequence of points $x_0, x_1, x_2, \ldots$ such that

$$f(x_t) - f(x^\star) \leq O\left(\frac{LD^2}{t^2}\right).$$

Denote $R_t = f(x_t) - f(x^\star)$. Initially we have

$$R_0 \geq \frac{\alpha}{2}\|x_0 - x^\star\|_2^2 = \frac{\alpha}{2}D^2,$$

becuase of strong convexity of f. Thus, the convergence guarantee of the algorithm can be rewritten as

$$R_t \leq O\left(\frac{LD^2}{t^2}\right) \leq O\left(\frac{LR_0}{\alpha t^2}\right).$$

In particular, the number of steps required to bring the gap R_t from R_0 to $\frac{R_0}{2}$ is $O\left(\sqrt{\frac{L}{\alpha}}\right)$. Thus, to go from R_0 to ε we need

$$O\left(\sqrt{\frac{L}{\alpha}} \cdot \log\frac{R_0}{\varepsilon}\right) = O\left(\sqrt{\frac{L}{\alpha}} \cdot \log\frac{LD^2}{\varepsilon}\right)$$

steps. The map ∇R is identity in this case, hence no additional computational cost is incurred because of R. □

2 Application to solving linear systems

2.1 Problem Statement

Consider the problem of solving linear systems of equations.

Solving Systems of Linear Equations.

Input: A square matrix $A \in \mathbb{R}^{n\times n}$ and a vector $b \in \mathbb{R}^n$.

Goal: Find a vector $x \in \mathbb{R}^n$ such that $Ax = b$.

In the above for simplicity we assume that A is a square matrix. We will also assume that A is non-singular and hence the system $Ax = b$ has a unique solution.

2.2 Previous Work

There has been a lot of work to understand the complexity of solving linear systems. Gaussian eliminitation gives an algorithm with worst case time complexity $O(n^3)$. This can be improved upon by using fast matrix multiplication in $O(n^\omega) = O(n^{2.373})$.

Several different approaches have been proposed for solving this problem under various assumptions on the matrix A. The running time of these methods then depends not only on the dimension n but also quantities such as the *condition number* of A (defined later). Perhaps the most well-known example of such a method is the Conjugate Gradient method, see [5].

2.3 The Algorithm

Before we state the algorithm let us first set up some notation. We assume that the matrix A is non-singular and hence $A^\top A$ is positive definite. Let $\lambda_{\min}(A^\top A)$ and $\lambda_{\max}(A^\top A)$ be the smallest and largest eigenvalue of $A^\top A$ respectively. We define the condition number of $A^\top A$ to be

$$\kappa(A^\top A) := \frac{\lambda_{\max}(A^\top A)}{\lambda_{\min}(A^\top A)}.$$

Note that for arbitrary matrices A, $\kappa(A^\top A)$ can be quite large, but for special classes of A, such as matrices coming from graphs: random walk matrices, Laplacian matrices, etc. the condition number is often bounded as a function of n, such as $O(n)$. The algorithm we present below works efficiently if the condition number is small.

Theorem 6. *There is an algorithm, which given a square matrix $A \in \mathbb{R}^{n\times n}$, a vector b and a precision parameter $\varepsilon > 0$, outputs a point $y \in \mathbb{R}^n$ – an approximate solution to the linear system $Ax = b$, which satisfies*

$$\|Ay - b\|^2 \leq \varepsilon.$$

The algorithm performs $T = O\left(\sqrt{\kappa(A^\top A)} \log\left(\frac{\lambda_{\max}(A^\top A)\|x^\star\|}{\varepsilon}\right)\right)$ iterations (where $x^\star \in \mathbb{R}^n$ satisfies $Ax^\star = b$), each iteration requires computing a constant number of matrix-vector multiplications and inner products.

Proof. We apply Theorem 5 to solve the optimization problem

$$\min_{x\in\mathbb{R}^n} \|Ax - b\|_2^2.$$

Let us denote $f(x) := \|Ax - b\|_2^2$. Note that the optimal value of the above is 0, and is achieved for $x = x^\star$, where $x^\star$ is the solution to the considered linear systems.

Let us derive all the relevant parameters of f which are required to apply Theorem 5. By computing the Hessian of f we have

$$\nabla^2 f(x) = A^\top A.$$

Since $\lambda_{\min}(A^\top A) \cdot I \preceq A^\top A$ and $A^\top A \preceq \lambda_{\max}(A^\top A)$ we have that f is L-smooth for $L = \lambda_{\max}(A^\top A)$ and α-strongly convex for $\alpha = \lambda_{\min}(A^\top A)$.

As a starting point x_0 we can choose $x_0 := 0$ which is of distance $D := \|x_0 - x^\star\| = \|x^\star\|$ from the optimal solution. Thus, from Theorem 5 we obtain the running time

$$O\left(\sqrt{\kappa(A^\top A)} \cdot \log\left(\frac{\lambda_{\max}(A^\top A)\,\|x^\star\|}{\varepsilon}\right)\right).$$

Note that computing the gradient of $f(x)$ is

$$\nabla f(x) = A^\top(Ax - b),$$

hence it boils down to performing two matrix-vector multiplications. □

References

[1] Zeyuan Allen Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 3:1–3:22, 2017.

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

[3] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2). In *Doklady AN USSR*, volume 269, pages 543–547, 1983.

[4] Yurii Nesterov. *Introductory lectures on convex optimization*, volume 87. Springer Science & Business Media, 2004.

[5] Sushant Sachdeva and Nisheeth K. Vishnoi. Faster algorithms via approximation theory. *Foundations and Trends in Theoretical Computer Science*, 9(2):125–210, 2014.