Optimization for Machine Learning
CS-439

Lecture 10: Lower Bounds and Accelerated Gradient Descent

Nicolas Flammarion

EPFL – `github.com/epfml/OptML_course`

May 9, 2025

Can We be Faster?

The rates of covergence of Gradient Descent for various function classes we have seen so far

Function Class	Rate for GD
Convex & L-Lipschitz	$\mathcal{O}(1/\sqrt{T})$
Convex & L-Smooth	$\mathcal{O}(1/T)$
μ-Strongly convex & L-Smooth	$\mathcal{O}\left(\exp\left\{-\mu T/L\right\}\right)$

Table: Rates of convergence of Gradient Descent after T iterations for various function classes.

Question: Is GD the optimal algorithm ? What optimal rates of convergence do we expect for these class of functions?

Lower Bounds

Why do we need computational lower bounds?

- Lower bounds establish the optimality of an algorithm
- Lower bounds can guide the design of efficient algorithms

The lower bounds we present first appeared in the works of Nemirovski and Yudin [NY83] and follows the revised presentation by Nesterov [Nes18].

The Class of Algorithms

Before we start, let us define the class of gradient-based algorithms we are interested in.

Definition: For a function f, a gradient-based algorithm initialized at $\mathbf{x}_0$ is defined as a sequence of points $\mathbf{x}_0, \mathbf{x}_1, \ldots$ generated by the following update rule:

$$\mathbf{x}_{t+1} \in x_0 + \mathrm{Span}\left\{\nabla f(\mathbf{x}_0), \nabla f(\mathbf{x}_1), \ldots, \nabla f(\mathbf{x}_t)\right\}$$

Note that GD with a constant stepsize η belongs to the class of gradient-based algorithm. By unraveling the updates of GD,

$$\mathbf{x}_t = \mathbf{x}_0 - \eta \sum_{i=0}^{t-1} \nabla f(\mathbf{x}_i),$$

belongs to the span of the past gradients.

The Strategy for Lower Bounds

- Construct a pathological family of functions, such that any gradient-based method cannot do better than a certain rate.
- For this lecture, we will focus on the class of convex and L-smooth functions over $\mathbb{R}^d$.

The Worst Function in the World

Let $f : \mathbb{R}^d \to \mathbb{R}$ be the function defined as

$$f(\mathbf{x}) = \frac{L}{4}\left[\frac{1}{2}\mathbf{x}[1]^2 + \frac{1}{2}\sum_{i=1}^{d-1}(\mathbf{x}[i] - \mathbf{x}[i{+}1])^2 + \frac{1}{2}\mathbf{x}[d]^2 - \mathbf{x}[1]\right]. \tag{1}$$

With the tri-diagonal matrix $A \in \mathbb{R}^{d\times d}$ defined by

$$A = \begin{bmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 \\ 0 & 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 2 \end{bmatrix},$$

the function f can be altenatively written as

$$f(\mathbf{x}) = \frac{L}{8}\mathbf{x}^T A\mathbf{x} - \frac{L}{4}\mathbf{x}[1].$$

Properties of the Worst Function

Convexity and Smoothness. Note that $\nabla^2 f(\mathbf{x}) = (L/4) \cdot A$. For any vector $h \in \mathbb{R}^d$, note that

$$h^\top A h = \sum_{i=1}^{d-1} (h[i] - h[i+1])^2 + h[1]^2 + h[d]^2.$$

As $h^\top A h \geq 0$, we have that A is positive semi-definite. Thus, f is convex. To prove smoothness,

$$\begin{aligned} h^\top A h &= \sum_{i=1}^{d-1} (h[i] - h[i+1])^2 + h[1]^2 + h[d]^2 \\ &\leq 2 \sum_{i=1}^{d-1} \left[h[i]^2 + h[i+1]^2 \right] + h[1]^2 + h[d]^2 \leq 4 \sum_{i=1}^{d} h[i]^2 = 4\|h\|^2. \end{aligned}$$

Therefore, $A \preceq 4 I_d$ and f is L-smooth.

Optimum of the Worst Function

Gradient. The gradient of f is given by

$$\nabla f(\mathbf{x}) = \frac{L}{4}\left(A\mathbf{x} - e_1\right),$$

where $e_1 = (1, 0, \cdots)$ is the first elementary basis vector of $\mathbb{R}^d$. The optimum $\mathbf{x}^*$ is obtained by solving the linear system

$$A\mathbf{x}^* = e_1.$$

The solution is given by

$$\mathbf{x}^*[i] = 1 - \frac{i}{d+1} \quad \forall\, 1 \leq i \leq d.$$

The Trajectory of The Gradient-Based Algorithm

Lemma

Without loss of generality, we can assume that $\mathbf{x}_0 = 0$*. Then, for* $t \leq d$ *the trajectory of the gradient-based algorithm for the function* f *defined in Eq.* (1) *satisfies*

$$\mathbf{x}_t \in Span\left\{e_1, e_2, \ldots, e_t\right\},$$

where e_i *is the* i*-th elementary basis vector of* $\mathbb{R}^d$*.*

Proof. The proof follows from induction. Let, $\mathbf{x}_s \in \mathrm{Span}\left\{e_1, e_2, \ldots, e_s\right\}$ for all $0 \leq s < t$. Note that

- $\nabla f(\mathbf{x}_s) = \frac{L}{4}\left(A\mathbf{x}_s - e_1\right) \in \mathrm{Span}\left\{e_1, e_2, \ldots, e_{s+1}\right\}$, i.e, the gradient update is restricted to the first $s+1$ coordinates.

Therefore, $\mathbf{x}_{s+1} \in \mathrm{Span}\left\{\nabla f(\mathbf{x}_0), \nabla f(\mathbf{x}_1), \ldots, \nabla f(\mathbf{x}_s)\right\}$ is also restricted to the first $s+1$ coordinates. By induction, we have that $\mathbf{x}_t \in \mathrm{Span}\left\{e_1, e_2, \ldots, e_t\right\}$.

Lower Bound

Lemma

For any gradient-based algorithm and for $t = {}^{(d-1)}/_{2}$, there exists a function f defined in Eq. (1) for which the following lower bound holds:

$$\min_{s \leq t} f(\mathbf{x}_s) - f(\mathbf{x}^*) \geq \frac{3L\|\mathbf{x}_0 - \mathbf{x}^*\|^2}{32(t+1)^2}$$

Comment. The lower bound is valid only when the iterations are comparable with the dimension of the problem, i.e, $t < d/2$.

Proof. From the previous lemma, we have that $\mathbf{x}_t \in \mathrm{Span}\{e_1, e_2, \ldots, e_t\}$. Using this fact,

$$f(\mathbf{x}_t) \geq \min_{\mathbf{x} \in \mathrm{Span}\{e_1, e_2, \ldots, e_t\}} f(\mathbf{x}).$$

Denote $\mathbf{x}_t^\star = \operatorname*{argmin}_{\mathbf{x} \in \mathrm{Span}\{e_1, e_2, \ldots, e_t\}} f(\mathbf{x})$, be the minimizer of f over the span of the first t coordinates.

Lower Bound Continued ...

Note that the funtion f over the set $\mathrm{Span}\{e_1, e_2, \ldots, e_t\}$ is

$$f_t(\mathbf{x}) = \frac{L}{4}\left[\frac{1}{2}\mathbf{x}[1]^2 + \frac{1}{2}\sum_{i=1}^{t-1}(\mathbf{x}[i] - \mathbf{x}[i{+}1])^2 + \frac{1}{2}\mathbf{x}[t]^2 - \mathbf{x}[1]\right].$$

Using the same computation as the minimizer of f, we have that the minimizer of f_t is given by

$$\mathbf{x}_t^\star[i] = \begin{cases} 1 - \frac{i}{t+1} & \text{for } 1 \leq i \leq t, \\ 0 & \text{for } t+1 \leq i \leq d \end{cases}$$

Combinig the previous results,

$$f(\mathbf{x}_t) - f(\mathbf{x}^*) \geq f(\mathbf{x}_t^\star) - f(\mathbf{x}^*). \tag{2}$$

Lower Bound Continued ...

Computing $f(\mathbf{x}^*)$ and $f(\mathbf{x}_t^\star)$,

$$f(\mathbf{x}^*) = \frac{L}{4}\left[\frac{1}{2}\mathbf{x}^*[1]^2 + \frac{1}{2}\sum_{i=1}^{d-1}(\mathbf{x}^*[i] - \mathbf{x}^*[i{+}1])^2 + \frac{1}{2}\mathbf{x}^*[t]^2 - \mathbf{x}^*[1]\right]$$

$$= \frac{L}{4}\left[\frac{1}{2}\left(\frac{d}{d+1}\right)^2 + \frac{d-1}{2}\left(\frac{1}{d+1}\right)^2 + \frac{1}{2}\left(\frac{1}{d+1}\right)^2 - \frac{d}{d+1}\right] = -\frac{L}{8}\left[1 - \frac{1}{d+1}\right]$$

Similarly, $f(\mathbf{x}_t^\star) = -\frac{L}{8}\left[1 - \frac{1}{t+1}\right]$. Using Eq. (2), we have that,

$$f(\mathbf{x}_t) - f(\mathbf{x}^*) \geq \frac{L}{8}\left[\frac{1}{t+1} - \frac{1}{d+1}\right],$$

$$\min_{s\leq t} f(\mathbf{x}_s) - f(\mathbf{x}^*) \geq \frac{L}{8}\left[\min_{s\leq t}\frac{1}{s+1} - \frac{1}{d+1}\right] = \frac{L}{8}\left[\frac{1}{t+1} - \frac{1}{d+1}\right]$$

Using the assumption that $d+1 = 2(t+1)$, $f(\mathbf{x}_t) - f(\mathbf{x}^*) \geq \frac{L}{16}\frac{1}{t+1}$. We gave a lower bound for the residual for any gradient-based algorithm.

Lower Bound Continued ...

Note that the rates are usually also depend on $\|\mathbf{x}_0 - \mathbf{x}^*\|^2$, we compute that to give the final lower bound.

$$\begin{aligned}
\|\mathbf{x}_0 - \mathbf{x}^*\|^2 = \|\mathbf{x}^*\|^2 = \sum_{i=1}^{d} \left(1 - \frac{i}{d+1}\right)^2 \\
= \sum_{i=1}^{d} \left(1 - \frac{2i}{d+1} + \frac{i^2}{(d+1)^2}\right) = d - 2\frac{d(d+1)}{2(d+1)} + \frac{d(d+1)(2d+1)}{6(d+1)^2} \\
= \frac{d(2d+1)}{6(d+1)} \leq \frac{d+1}{3} = \frac{2(t+1)}{3}
\end{aligned}$$

The final bound using the above computation writes,

$$\frac{f(\mathbf{x}_t) - f(\mathbf{x}^*)}{\|\mathbf{x}_0 - \mathbf{x}^*\|^2} \geq \frac{L}{16}\frac{1}{t+1} \cdot \frac{3}{2(t+1)} = \frac{3L}{32(t+1)^2}$$

Matching the Lower Bound: Acceleration for Smooth Convex Functions

Nesterov 1983 [Nes83, Nes18]: There is a gradient-based algorithm that achieves the rate of $\mathcal{O}(1/T^2)$ for T steps on smooth convex functions, and by the lower bound of Nemirovski and Yudin, this is a best possible algorithm!

The algorithm is known as (Nesterov's) accelerated gradient descent.

A number of (similar) optimal algorithms with other proofs for the rate of $\mathcal{O}(1/T^2)$ are known, but there is no well-established "simplest proof".

Here: a recent proof based on potential functions [BG17]. Proof is simple but not very instructive (it works, but it's not clear why).

Nesterov's accelerated gradient descent

Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex, differentiable, and smooth with parameter L. Choose $\mathbf{z}_0 = \mathbf{y}_0 = \mathbf{x}_0$ arbitrary. For $t \geq 0$, set

$$
\begin{aligned}
\mathbf{y}_{t+1} &:= \mathbf{x}_t - \frac{1}{L}\nabla f(\mathbf{x}_t) \\
\mathbf{z}_{t+1} &:= \mathbf{z}_t - \frac{t+1}{2L}\nabla f(\mathbf{x}_t) \\
\mathbf{x}_{t+1} &:= \frac{t+1}{t+3}\mathbf{y}_{t+1} + \frac{2}{t+3}\mathbf{z}_{t+1}.
\end{aligned}
$$

- Perform a "smooth step" from $\mathbf{x}_t$ to $\mathbf{y}_{t+1}$.
- Perform a more aggressive step from $\mathbf{z}_t$ to $\mathbf{z}_{t+1}$.
- Next iterate $\mathbf{x}_{t+1}$ is a weighted average of $\mathbf{y}_{t+1}$ and $\mathbf{z}_{t+1}$, where we compensate for the more aggressive step by giving $\mathbf{z}_{t+1}$ a relatively low weight.

Why should this work??

Nesterov's accelerated gradient descent: Error bound

Theorem

Let $f : \mathbb{R}^d \to \mathbb{R}$ be convex and differentiable with a global minimum $\mathbf{x}^\star$; furthermore, suppose that f is smooth with parameter L. Accelerated gradient descent yields

$$f(\mathbf{y}_T) - f(\mathbf{x}^\star) \leq \frac{2L \left\| \mathbf{z}_0 - \mathbf{x}^\star \right\|^2}{T(T+1)}, \quad T > 0.$$

To reach error at most ε, accelerated gradient descent therefore only needs $\mathcal{O}(1/\sqrt{\varepsilon})$ steps instead of $\mathcal{O}(1/\varepsilon)$.

Recall the bound for gradient descent:

$$f(\mathbf{x}_T) - f(\mathbf{x}^\star) \leq \frac{L}{2T} \|\mathbf{x}_0 - \mathbf{x}^\star\|^2, \quad T > 0.$$

Nesterov's accelerated gradient descent: The potential function

Idea: assign a potential $\Phi(t)$ to each time t and show that $\Phi(t+1) \leq \Phi(t)$.

Out of the blue: let's define the potential as

$$\Phi(t) := t(t+1)\left(f(\mathbf{y}_t) - f(\mathbf{x}^\star)\right) + 2L\left\|\mathbf{z}_t - \mathbf{x}^\star\right\|^2.$$

If we can show that the potential always decreases, we get

$$\underbrace{T(T+1)\left(f(\mathbf{y}_T) - f(\mathbf{x}^\star)\right) + 2L\left\|\mathbf{z}_T - \mathbf{x}^\star\right\|^2}_{\Phi(T)} \leq \underbrace{2L\left\|\mathbf{z}_0 - \mathbf{x}^\star\right\|^2}_{\Phi(0)}.$$

Rewriting this, we get the claimed error bound.

(optional material) Potential function decrease: Three Ingredients

Sufficient decrease for the smooth step from $\mathbf{x}_t$ to $\mathbf{y}_{t+1}$:

$$f(\mathbf{y}_{t+1}) \le f(\mathbf{x}_t) - \frac{1}{2L}\|\nabla f(\mathbf{x}_t)\|^2; \tag{3}$$

Vanilla analysis for the more aggressive step from $\mathbf{z}_t$ to $\mathbf{z}_{t+1}$: ($\gamma = \frac{t+1}{2L}$, $\mathbf{g}_t = \nabla f(\mathbf{x}_t)$):

$$\mathbf{g}_t^\top(\mathbf{z}_t - \mathbf{x}^\star) = \frac{t+1}{4L}\|\mathbf{g}_t\|^2 + \frac{L}{t+1}\left(\|\mathbf{z}_t - \mathbf{x}^\star\|^2 - \|\mathbf{z}_{t+1} - \mathbf{x}^\star\|^2\right); \tag{4}$$

Convexity (graph of f is above the tangent hyperplane at $\mathbf{x}_t$):

$$f(\mathbf{x}_t) - f(\mathbf{w}) \le \mathbf{g}_t^\top(\mathbf{x}_t - \mathbf{w}), \quad \mathbf{w} \in \mathbb{R}^d. \tag{5}$$

(optional material) Potential function decrease: Proof

By definition of potential,

$$\begin{aligned}
\Phi(t+1) &= t(t+1)\left(f(\mathbf{y}_{t+1}) - f(\mathbf{x}^\star)\right) + 2(t+1)\left(f(\mathbf{y}_{t+1}) - f(\mathbf{x}^\star)\right) + 2L\left\|\mathbf{z}_{t+1} - \mathbf{x}^\star\right\|^2, \\
\Phi(t) &= t(t+1)\left(f(\mathbf{y}_{t}) - f(\mathbf{x}^\star)\right) + 2L\left\|\mathbf{z}_{t} - \mathbf{x}^\star\right\|^2.
\end{aligned}$$

Now, prove that $\Delta := (\Phi(t+1) - \Phi(t))/(t+1) \leq 0$:

$$\begin{aligned}
\Delta &= t\left(f(\mathbf{y}_{t+1}) - f(\mathbf{y}_t)\right) + 2\left(f(\mathbf{y}_{t+1}) - f(\mathbf{x}^\star)\right) + \frac{2L}{t+1}\left(\left\|\mathbf{z}_{t+1} - \mathbf{x}^\star\right\|^2 - \left\|\mathbf{z}_t - \mathbf{x}^\star\right\|^2\right) \\
&\overset{(4)}{=} t\left(f(\mathbf{y}_{t+1}) - f(\mathbf{y}_t)\right) + 2\left(f(\mathbf{y}_{t+1}) - f(\mathbf{x}^\star)\right) + \frac{t+1}{2L}\|\mathbf{g}_t\|^2 - 2\mathbf{g}_t^\top(\mathbf{z}_t - \mathbf{x}^\star) \\
&\overset{(3)}{\leq} t\left(f(\mathbf{x}_t) - f(\mathbf{y}_t)\right) + 2\left(f(\mathbf{x}_t) - f(\mathbf{x}^\star)\right) - \frac{1}{2L}\|\mathbf{g}_t\|^2 - 2\mathbf{g}_t^\top(\mathbf{z}_t - \mathbf{x}^\star) \\
&\leq t\left(f(\mathbf{x}_t) - f(\mathbf{y}_t)\right) + 2\left(f(\mathbf{x}_t) - f(\mathbf{x}^\star)\right) - 2\mathbf{g}_t^\top(\mathbf{z}_t - \mathbf{x}^\star) \\
&\overset{(5)}{\leq} t\mathbf{g}_t^\top(\mathbf{x}_t - \mathbf{y}_t) + 2\mathbf{g}_t^\top(\mathbf{x}_t - \mathbf{x}^\star) - 2\mathbf{g}_t^\top(\mathbf{z}_t - \mathbf{x}^\star) \\
&= \mathbf{g}_t^\top((t+2)\mathbf{x}_t - t\mathbf{y}_t - 2\mathbf{z}_t) \overset{\text{(algo)}}{=} \mathbf{g}_t^\top\mathbf{0} = 0. \quad \square
\end{aligned}$$

Bibliography

Nikhil Bansal and Anupam Gupta.
Potential-function proofs for first-order methods.
CoRR, abs/1712.04581, 2017.

Yurii Nesterov.
A method of solving a convex programming problem with convergence rate $o(1/k^2)$.
Soviet Math. Dokl., 27(2), 1983.

Yurii Nesterov.
Lectures on Convex Optimization, volume 137.
Springer, 2018.

Arkady. S. Nemirovsky and D. B. Yudin.
Problem complexity and method efficiency in optimization.
Wiley, 1983.