Lecture 13

13.1 Lower Complexity Bound

We proved the following result for the subgradient method minimizing convex Lipschitz functions:

$$f(\bar{x}_K) - f^\star \quad \leq \quad \tfrac{MR}{\sqrt{K}}. \tag{13.1}$$

In the first part of the course, we saw that in the smooth convex case, it is possible to have better rates (for the gradient method and for the accelerated fast gradient method). However, in non-smooth convex optimization, it appears that the rate (13.1) is optimal.

13.1.1 Problem Class

To prove the lower bound, we restrict ourselves onto the following class of problems, which is obviously a particular case of our situation.

$$f^\star \quad = \quad \min_{x \in \mathbb{R}^n} \Big\{ \, f(x) \quad : \quad \|x\| \leq R \, \Big\}, \tag{13.2}$$

where f is convex and Lipschitz continuous:

$$f(y) - f(x) \quad \leq \quad M\|y - x\|, \qquad \forall x, y.$$

The norm is the standard Euclidean, and $M > 0$ and $R > 0$ our two key complexity parameters.

We consider the class of *all first-order optimization methods*, starting from an arbitrary initialization x_1[1]. We associate an optimization method with a sequence of mappings:

$$\mathcal{A} \quad = \quad (A_1, A_2, \ldots)$$

that define the iteration process:

$$x_{k+1} \quad = \quad A_k(\mathcal{O}_f(x_1), \ldots, \mathcal{O}_f(x_k)), \quad k \geq 1,$$

where $\mathcal{O}_f(x) := (f(x), f'(x))$ for some arbitrary subgradient $f'(x) \in \partial f(x)$. Therefore, each mapping A_k, $k \geq 1$ takes all first-order oracle information available up to the current moment and return the next iterate. Without loss of generality, we assume that the result of the algorithm is the last generated point: x_K, where $K \geq 1$ is a fixed number of iterations.

[1] We start iterations from $k \geq 1$ in this lecture to keep our notation simpler.

13.1.2 Lower Bound

We prove the following theorem.

Theorem 13.1.1. *Let $M > 0$ and $R > 0$ be fixed. Then, for any first-order algorithm running for $K \geq 1$ iterations, there exists a function $f : \mathbb{R}^n \to \mathbb{R}$ with $n \geq K$ such that*

1. *f is convex and Lipschitz with constant M;*
2. *For the output of the algorithm on this function, it holds:*

$$f(x_K) - f^\star \quad \geq \quad \frac{MR}{2\sqrt{K}}.$$

Proof. Let $\delta > 0$ be a fixed parameter (it can be arbitrarily small).

We use a *resisting oracle* that will choose a set of numbers (orientations):

$$\xi_1, \ldots, \xi_K \quad \in \quad \{-1, 1\},$$

and a permutation of coordinates:

$$t \quad \mapsto \quad \sigma(t) \quad \in \quad \{1, 2, \ldots, n\}.$$

These parameters will be built along the oracle calls from the given algorithm, which is an arbitrary first-order method. We also denote by $e_i \in \mathbb{R}^n$, $1 \leq i \leq n$ the standard basis vectors.

We consider the following family of convex function, $x \in \mathbb{R}^n$:

$$f_k(x) \quad := \quad M \cdot \max_{1 \leq i \leq k} \Big[\, \xi_i \langle e_{\sigma(i)}, x \rangle - (i-1)\delta \, \Big]. \tag{13.3}$$

Note that

$$\langle e_{\sigma(i)}, x \rangle \quad = \quad x^{(\sigma(i))} \quad \in \quad \mathbb{R}$$

is the ith coordinate of a vector after permuting the coordinates.

It is clear that every $f_k(\cdot)$ is Lipschitz continuous with constant M.

1. Let us analyze its minimum. We have

$$\begin{aligned} f_k^\star \quad &:= \quad \min_{\|x\| \leq R} f_k(x) \quad = \quad M \cdot \min_{\|x\| \leq R} \max_{1 \leq i \leq k} \Big[\, \xi_i \langle e_{\sigma(i)}, x \rangle - (i-1)\delta \, \Big] \\ &\leq \quad M \cdot \min_{\|x\| \leq R} \max_{1 \leq i \leq k} \xi_t \langle e_{\sigma(i)}, x \rangle \quad = \quad M \cdot \min_{\|x\| \leq R} \max_{1 \leq i \leq k} \langle e_{\sigma(i)}, x \rangle \quad = \quad -M\gamma, \end{aligned}$$

where we noticed that, due to symmetry in the problem, the minimum is achieved when all first k coordinates after the permutation are the same:

$$x^{(\sigma(1))} \quad = \quad x^{(\sigma(2))} \quad = \quad \ldots \quad = \quad x^{(\sigma(k))} \quad = \quad -\gamma,$$

for some $\gamma > 0$, while other coordinates are zero. The smallest value is achieved at the boundary of the ball, $\|x\|^2 = k\gamma^2 = R^2$, and we find that $\gamma := \frac{R}{\sqrt{k}}$. Hence,

$$f_k^\star \quad \leq \quad -\frac{MR}{\sqrt{k}}. \tag{13.4}$$

2. Now, we present a *resisting strategy* for choosing ξ_k and $\sigma(k)$. We pick them *adversarially* by following the following rules.

- At first step, the algorithm asks the oracle information at the initial point x_1. Let us pick

$$\sigma(1) \quad \in \quad \arg\max_{1\le i\le n} |\langle e_i, x_1\rangle|$$

In other words, $\sigma(1)$ is an index of a maximal entry (in absolute value) among coordinates of x_1. Then, we specify $\xi_1 \in \{-1, 1\}$ in a way that

$$\xi_1\langle e_{\sigma(1)}, x_1\rangle \quad = \quad |\langle e_{\sigma(1)}, x_1\rangle|,$$

hence $\xi_1 = \mathrm{sign}(\langle e_{\sigma(1)}, x_1\rangle) = \mathrm{sign}(x_1^{(\sigma(1))})$. So

$$f_1(x) \quad = \quad M \cdot \xi_1 \cdot \langle e_{\sigma(1)}, x\rangle$$

is fully defined.

- For $1 \le k \le K-1$, assume that we have built $f_k(x)$. Let x_{k+1} be the point of the trajectory of $\mathcal{A}$ at iteration k applied to $f_k(x)$:

$$x_{k+1} \quad = \quad \mathcal{A}(\mathcal{O}_{f_k}(x_1), \ldots, \mathcal{O}_{f_k}(x_k)).$$

Note that it can be arbitrary as we cannot control what the method returns.

Let us choose as $\sigma(k+1)$ the index of a maximal element of x_{k+1} (in absolute value), except $\sigma(1), \ldots, \sigma(k)$. Thus,

$$\sigma(k+1) \quad \in \quad \arg\max_{1\le i\le n \text{ s.t. } i\notin\{\sigma(1),\ldots,\sigma(k)\}} |\langle e_i, x_{k+1}\rangle|,$$

and specify $\xi_{k+1} \in \{-1, 1\}$ such that

$$\xi_{k+1}\langle e_{\sigma(k+1)}, x_{k+1}\rangle \quad = \quad |\langle e_{\sigma(k+1)}, x_{k+1}\rangle|,$$

i.e. $\xi_{k+1} = \mathrm{sign}(\langle e_{\sigma(k+1)}, x_{k+1}\rangle) = \mathrm{sign}(x_{k+1}^{(\sigma(k+1))})$. Thus we obtained the next $f_{k+1}(x)$.

3. We need to verify the following important fact, which is the outcome of our resisting strategy: for any $s < k$, functions $f_s(\cdot)$ and $f_k(\cdot)$ are *informationally indistinguishable* for any local oracle at x_s:

$$f_s(x) \quad \equiv \quad f_k(x), \qquad \forall x \quad \text{s.t.} \quad \|x - x_s\| \le \delta. \tag{13.5}$$

This implies that the first-order oracle information at x_s is identical for both functions, as the subdifferential is fully determined by function values in a neighbourhood of that point (for example, via the directional derivatives). For our method, it means that all oracle information received in the past remains consistent for subsequent functions in the "future":

$$\boxed{\mathcal{O}_{f_s}(x_s) \quad = \quad \mathcal{O}_{f_{s+1}}(x_s) \quad = \quad \mathcal{O}_{f_{s+2}}(x_s) \quad = \quad \ldots \quad = \quad \mathcal{O}_{f_k}(x_s)}$$

so running the algorithm for s iterations on f_s is the same as performing these iterations on f_k.

To prove (13.5), we note that

$$f_k(x) \quad = \quad \max\Big\{ f_s(x), \;\; M \cdot \max_{s<i\le k} \big[\, \xi_i\langle e_{\sigma(i)}, x\rangle - (i-1)\delta \,\big] \Big\}. \tag{13.6}$$

By the definition of $f_s(\cdot)$, due to our choice of $\sigma(s)$ and ξ_s, we have

$$\xi_s \langle e_{\sigma(s)}, x_s \rangle \;\; \geq \;\; \xi_i \langle e_{\sigma(i)}, x_s \rangle, \qquad \forall i > s.$$

Hence,

$$f_s(x_s) \;\; \geq \;\; M \cdot [\; \xi_i \langle e_{\sigma(i)}, x_s \rangle - (i-1)\delta \;] \;\; + \;\; M \cdot \delta, \qquad \forall i > s. \tag{13.7}$$

Due to the Lipschitz continuity, from (13.7), it holds for all x such that $\|x - x_s\| \leq \delta$ that

$$f_s(x) \;\; \geq \;\; M \cdot [\; \xi_t \langle e_{\sigma(t)}, x \rangle - (t-1)\delta \;], \qquad \forall i > s. \tag{13.8}$$

Applying this bound to (13.6) we conclude that (13.5) is true.

4. For the final function, we take $\boxed{f(x) := f_K(x)}$. Then, for the output x_K of the algorithm as applied to f, we have

$$\begin{aligned} f(x_K) \;\; &= \;\; f_K(x_K) \;\; \geq \;\; M \cdot [\, \xi_K \langle e_{\sigma(K)}, x_K \rangle - (K-1)\delta \,] \\ &= \;\; M \cdot [\, |x_K^{(\sigma(K))}| - (K-1)\delta] \;\; \geq \;\; -(K-1)\delta. \end{aligned} \tag{13.9}$$

Note that since $\delta > 0$ can be arbitrarily small, we can have $f(x_K) \approx 0$. At the same time, bound (13.4) shows that the exact minimum is strictly below zero. We finally obtain the following bound for the functional residual:

$$f(x_K) - f^\star \;\; \overset{(13.9),(13.4)}{\geq} \;\; \tfrac{MR}{\sqrt{K}} - \delta(K-1) \;\; \geq \;\; \tfrac{MR}{2\sqrt{K}},$$

for a sufficiently small $\delta \leq \frac{MR}{2(K-1)\sqrt{K}}$, which completes the proof. □

Remark 13.1.2. Note that the functions $f_k(\cdot)$ from the construction of the lower bound are very simple. Consider for simplicity $M = 1$ in (13.3). Then,

- For $k = 1$, we have

$$f_1(x) \;\; = \;\; \Big[\; \xi_1 \langle e_{\sigma(1)}, x \rangle \;\Big] \;\; = \;\; \pm x^{(\sigma(1))}.$$

- For $k = 2$, we have

$$\begin{aligned} f_2(x) \;\; &= \;\; \max\Big[\; \xi_1 \langle e_{\sigma(1)}, x \rangle,\; \xi_2 \langle e_{\sigma(2)}, x \rangle - \delta \;\Big] \\ &= \;\; \max\Big[\; \pm x^{(\sigma(1))},\; \pm x^{(\sigma(2))} - \delta \;\Big]. \end{aligned}$$

The signs and permutations are chosen in an adversarial way.

Let us visualize these functions. As $\delta > 0$ can be arbitrarily small, we can set it to zero for our visualization. Up to the permutation of coordinates, the first function is either

$$f_1(x) \;\; = \;\; x^{(1)} \qquad \text{or} \qquad f_1(x) \;\; = \;\; -x^{(1)},$$

where $x \in \mathbb{R}^n$ lives in high-dimensional space. Their graphs for $n = 2$ are shown in Fig. 13.1.

At the second iterate, up to the permutation of coordinates, we have four possible functions (see Fig. 13.2), depending on the selection of signs:

$$\begin{aligned} f_2(x) \;\; &= \;\; \max[x^{(1)}, x^{(2)}], & f_2(x) \;\; &= \;\; \max[-x^{(1)}, x^{(2)}] \\ f_2(x) \;\; &= \;\; \max[-x^{(1)}, -x^{(2)}], & f_2(x) \;\; &= \;\; \max[x^{(1)}, -x^{(2)}], \end{aligned}$$

and so on. Each time, we manage to place the minimum $x^\star$ at a significant distance from the query point.

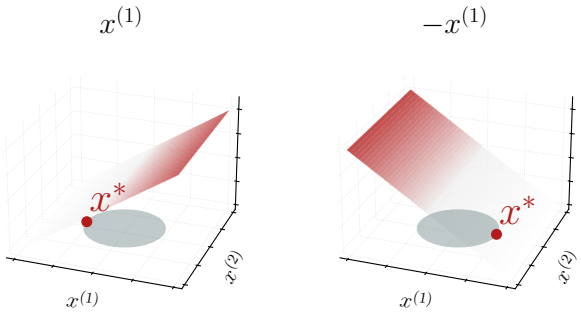

Figure 13.1: Two possible functions, $f_1(x) = x^{(1)}$ and $f_1(x) = -x^{(1)}$, to be selected at the first oracle call. The resisting strategy picks the function with the largest value at the requested point.

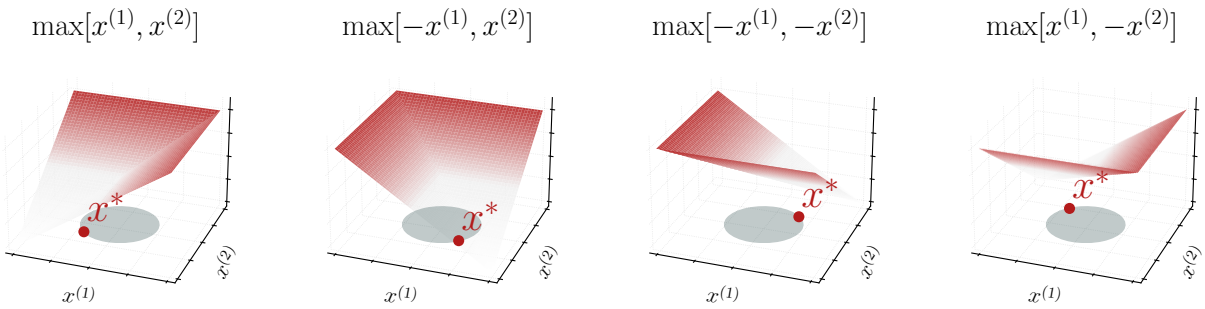

Figure 13.2: The four candidate functions after the resisting strategy at the second oracle call.

Remark 13.1.3. It is possible to set $\delta := 0$ in the worst-case function construction. However, in this case, the resisting oracle must be allowed to decide *which subgradient to return* when the query point is at a point where multiple components of the function are active.

By selecting $\delta > 0$, we ensure the same worst-case behavior even if the algorithm has access to the *entire subdifferential* $\partial f(x)$ at each query point.

13.2 Complexity of Non-Smooth Convex Optimization

Let us conclude by discussing the complexity landscape of black-box convex optimization. Depending on the problem dimension $n \geq 1$, there are different regimes with its own "optimal methods":

- $n = 1$: **univariate minimization** — *binary search*. While more practically efficient methods for univariate minimization exist, binary search remains optimal in terms of its oracle complexity and its practical simplicity.
- $n \geq 1$: **small dimension** — *cutting plane schemes*, which are generalizations of the binary search to higher dimensions. Allowing a method to perform $k \geq n$ iterations, the optimal complexity is
$$O(n \log \tfrac{MR}{\varepsilon}),$$
first-order oracle calls, which is achieved by the center of gravity method. However, this method is completely unpractical. The ellipsoid method, that we discussed previous lecture, has slightly worse oracle complexity:
$$O(n^2 \log \tfrac{MR}{\varepsilon}). \tag{13.10}$$

At the same time, the ellipsoid method is much more practical and it is possible to implement. However, due to the arithmetic cost of each iteration of order $O(n^2)$ and rather conservative steps, the ellipsoid method seems suitable for problems of range $n \approx 10 - 20$.

- $n \to \infty$: **large-scale optimization** — *cheap gradient / subgradient methods.* When $k \leq n$, the complexity of the subgradient method,

$$O\Big(\Big[\frac{MR}{\varepsilon}\Big]^2\Big), \tag{13.11}$$

is optimal, matching the lower bound that we have just proved. Note that in contrast to cutting-plane schemes, the complexity bound (13.11) *does not depend on the dimension*[2]. This makes it favorable for solving problems of huge dimensions, where no other methods can work.

Comparing the complexity bounds (13.11) and (13.10), we see that the subgradiend method is superior than the ellipsoid method (in terms of oracle complexity), at least when

$$n \quad \geq \quad \Big[\frac{MR}{\varepsilon}\Big].$$

Therefore,

1. The subgradient method is *superior* when the dimension is extremely large.
2. Conversely, when the target accuracy ε is very small ($\varepsilon \to 0$), the subgradient method cannot be expected to solve the problem to such precision.
3. The dependence of large-scale first-order methods on ε improves significantly when the problem is smooth (e.g., the gradient method: $O(1/\varepsilon)$, the fast gradient method: $O(1/\varepsilon^{1/2})$), or additionally strongly convex.

Another crucial aspect defining the success of subgradient-type methods is their *low per-iteration cost* and *resilience to noise*. As we will see, the same analysis that we applied to the subgradient method is directly applicable to stochastic methods, such as stochastic gradient descent (SGD) and its adaptive variants.

Finally, note an unsatisfying gap in this picture: problems of **moderate dimension** $10 \leq n \leq 10^4$, which routinely appear in computational practice. In the final part of the course, we will discuss *interior-point methods* (IPMs), that close this gap.

Interior-point methods require knowledge of the precise structure of the problem (such as linear or quadratic programming). Thus, they cannot formally be categorized as black-box methods. However, such structural knowledge is often available, and the IPMs remain extremely efficient for solving moderate-size problems with high accuracy, while possessing polynomial-time complexity.

[2] At least explicitly. As we will see, it may depend on the dimension n, through parameters M and R.

Literature

The lower complexity bounds for optimization methods were originally developed by Nemirovski and Yudin in their seminal book [NY83]. See also the lecture notes [Nem95].

A more advanced framework which incorporates different problem geometries (beyond the Euclidean) and various degrees of smoothness into the complexity bound is available in [GN15].

[GN15] Cristóbal Guzmán and Arkadi Nemirovski. On lower complexity bounds for large-scale smooth convex optimization. *Journal of Complexity*, 31(1):1–14, 2015.

[Nem95] Arkadi Nemirovski. *Information-based complexity of convex programming*. Lecture notes, 1995.

[NY83] Arkadi Nemirovski and David Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.