Lecture 11

11.1 Problem Formulation

Our goal is to solve the following convex optimization problem,

$$\min_{x \in Q} f(x) \tag{11.1}$$

where $Q \subseteq \mathbb{R}^n$ is a convex set, and $f : Q \to \mathbb{R}$ is a convex function. We consider a general situation, when f might not be differentiable, and set Q can also be a general convex set with difficult geometry.

To quantify the complexity of this problem, we need some regularity assumptions:

- For objective function f, we denote by V its *variation* over the set Q:

$$V \;:=\; \max_{x \in Q} f(x) - \min_{x \in Q} \;=\; \max_{x \in Q} f(x) - f^\star,$$

  and we assume that V is bounded: $V < +\infty$.

- For set Q, we assume that it is *bounded* and has a *nonempty interior* $\operatorname{int} Q \neq \varnothing$. When solving unconstrained optimization problems, we can always introduce an auxiliary ball of sufficiently big radius to satisfy this assumption. Quantitatively, we assume that there exist $0 < r \leq R < +\infty$ and $\bar{x} \in \operatorname{int} Q$ such that:

$$B_r(\bar{x}) \;\subseteq\; Q \;\subseteq\; B_R(\bar{x}),$$

  where $B_\alpha(\bar{x}) \;:=\; \{x \in \mathbb{R}^n \;:\; \|x - \bar{x}\|_2 \leq \alpha\}$ is the Euclidean ball of radius $\alpha \geq 0$.

  The ratio $\frac{R}{r} \geq 1$ is sometimes called the *asphericity* of Q, and can be seen as the "condition number" of the set.

Thus, the parameters V, r, R will describe the complexity of solving (11.1), but as we will see, the dependence on them is rather weak. The main complexity parameter will be the dimension n.

As before, our goal is to find a point $\bar{x} \in Q$ such that

$$f(\bar{x}) - f^\star \;\leq\; \varepsilon.$$

11.1.1 Separation Oracle

We assume that we have an access to the following *separation oracle* to solve (11.1):

$$\mathcal{O}(x) \;=\; \begin{cases} f'(x) \;\in\; \partial f(x), & \text{if } x \in \operatorname{int} Q, \\ s_Q(x), & \text{otherwise,} \end{cases} \tag{11.2}$$

where $s_Q(x) \in \mathbb{R}^n$, $s_Q(x) \neq 0$ is a vector that separates point x from Q. Thus, by the definition of $s_Q(x)$, it holds:

$$\langle s_Q(x), x - y \rangle \quad \geq \quad 0, \qquad y \in Q. \tag{11.3}$$

We know from the separation theorem that such a vector always exists for $x \notin \operatorname{int} Q$, but it may not be unique. A subgradient vector $f'(x) \in \partial f(x)$ is not uniquely defined. Therefore, we might have many options to actually implement the oracle $\mathcal{O}(x)$, and any particular selection works for us. Note that by the definition of the subgradient, we have:

$$\langle f'(x), x - y \rangle \quad \geq \quad f(x) - f(y), \qquad \forall x, y \in \operatorname{dom} f. \tag{11.4}$$

Without loss of generality, we can always assume that $f'(x) \neq 0$. Otherwise, if $f'(x) = 0$, inequality (11.4) implies that $x = x^\star$ is the global optimum and we can return as the result of a method.

At the same time, when $f'(x) \neq 0$, the subgradient also provides us with a *separation* of our space into two halves, and we know which half contains the optimum:

$$x^\star \quad \in \quad \Big\{ \; y \quad : \quad \langle f'(x), x - y \rangle \geq 0 \; \Big\}.$$

Geometrically, inequality $\langle f'(x), x - y \rangle \geq 0$ separates the sublevel set of the objective f at point x.

Example 11.1.1. Consider the set

$$Q \quad = \quad \{y \in \mathbb{R}^n \; : \; \langle a, y \rangle \leq b\},$$

for given $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$, which is a half space. Assume $x \notin \operatorname{int} Q$. What will be a result of the separation oracle $s_Q(x)$? It is easy to see that

$$s_Q(x) \quad := \quad a,$$

will do the job. Indeed, for $x \notin \operatorname{int} Q$ we have $\langle a, x \rangle \geq b$ At the same time, for any $y \in Q$ we have

$$\langle a, y \rangle \quad \leq \quad b \quad \leq \quad \langle a, x \rangle,$$

and condition (11.3) is satisfied.

Example 11.1.2. Using a separation oracle for a half space, it is very easy to implement the separation oracle for the polyhedron, which is the intersection of half spaces:

$$Q \quad = \quad \Big\{ y \in \mathbb{R}^n \; : \; Ay \leq b \} \quad = \quad \Big\{ y \in \mathbb{R}^n \quad : \quad \langle a_1, y \rangle \leq b_1, \ldots, \langle a_m, y \rangle \leq b_m \},$$

where $a_1, \ldots, a_m \in \mathbb{R}^n$ are rows of the matrix $A \in \mathbb{R}^{m \times n}$. For a given x, we need to go and check whether any of the inequalities is violated. If so, we return the vector a_i that correspond to the violated inequality. Thus, separation oracle for the linear programming can be implemented in $O(nm)$ arithmetical operations.

In a similar vein separation oracles can be developed for other types of standard convex sets.

11.2 Cutting Plane Scheme

The idea of cutting plane schemes in optimization is to use separation oracle to "cut" our search space into two halves and continue the search in one of them. This is a generalization of the binary search from the univariate case.

However, in one-dimensional case ($n = 1$), we usually have more or less one natural possibility:

- **The search region**, that we also call the the *localizer* G_k at iteration $k \geq 0$, is a segment:

$$G_k \quad = \quad [\ell_k, r_k], \qquad \ell_k < r_k.$$

  We ensure the invariant that $x^\star \in G_k$ (that is why it is called the localizer), and that

$$\text{size}(G_k) \quad := \quad r_k - \ell_k \quad \to \quad 0.$$

- **The next point**, x_k which defines where we access the oracle is the midpoint of the segment:

$$x_k \quad = \quad \tfrac{\ell_k + r_k}{2}.$$

Now, we want to generalize this construction to the multivariate case. For $n \geq 2$, we obtain great flexibility in how to define G_k and x_k, and there are many options that lead to particular implementations of the so-called *cutting plane scheme*.

The main obstacle in the general case $n \geq 2$ is that the "shape" of G_k might become quite difficult. For example, for a set of previous points $\{x_0, \ldots, x_k\}$ with known oracle information $g_i = \mathcal{O}(x_i)$, for $0 \leq i \leq k$, it is natural to construct the following set, which is the intersection of all separating half spaces:

$$G_{k+1} \quad = \quad \Big\{ \, y \in \mathbb{R}^n \ : \ \langle g_0, x_0 - y \rangle \geq 0, \ \ldots, \ \langle g_k, x_k - y \rangle \geq 0 \, \Big\}. \tag{11.5}$$

Then, clearly $x^\star \in G_{k+1}$. However, defined this way, set (11.5) becomes a polyhedron and a question of just finding a point $x_{k+1} \in G_{k+1}$ (the feasibility problem) is equivalent to a linear programming problem. But we do not want "any" point: we rather want a point such that sizes would go to zero: $\text{size}(G_k) \to 0$, which becomes it even more difficult to decide on x_{k+1}.

The idea to overcome this is as follows: instead of adding unstructured cuts (11.5), at every iteration $k \geq 0$ we preserve a structure of the set to be "simple by rich enough", defined by $E_k \subseteq \mathbb{R}^n$. In our case E_k will be an Ellipsoid centered at x_k:

$$E_k \quad := \quad \Big\{ y \in \mathbb{R}^n \ : \ \langle A_k^{-1}(y - x_k), y - x_k \rangle \quad \leq \quad 1 \Big\},$$

which is given by a positive definite symmetric matrix $A_k = A_k^\top \succ 0$. Our invariant remains that a solution always belongs to our localizers:

$$x^\star \quad \in \quad E_k, \qquad k \geq 0.$$

Then, at each iterate we call the separation oracle at the center x_k, and perform a single cut of E_k:

$$G_{k+1} \quad := \quad \Big\{ \, y \in E_k \quad : \quad \langle g_k, x_k - y \rangle \geq 0 \, \Big\}.$$

Thus, $x^\star \in G_{k+1}$, but G_{k+1} is not an ellipsoid anymore. To keep a simple structure, we have to *find a new ellipsoid* E_{k+1} that contains the halved one:

$$E_{k+1} \quad \supseteq \quad G_{k+1},$$

at that constitute one step of the method.

To initialize the method we have to choose the initial ellipsoid E_0. Usually it is taken to be a large enough Euclidean ball, $A_0 := \frac{1}{R}I$ around some given point $x_0 \in Q$, such that the feasible set is entirely contained in it:

$$E_0 \;:=\; B_R(x_0) \;\supseteq\; Q \;=:\; G_0,$$

such R exists by our assumption.

11.2.1 Notion of Size

Now, to be able to show that the method converges sufficiently fast, we want to ensure that our sets $\{E_k\}_{k\geq 0}$ are getting smaller and smaller in "size" with a certain good rate:

$$\text{size}(E_k) \;\to\; 0 \qquad \text{with} \qquad k \to +\infty. \tag{11.6}$$

For the ellipsoid method, we use the volume as a measure of the size, for any compact convex set with non-empty interior $K \subset \mathbb{R}^n$:

$$\text{size}(K) \;:=\; (\text{Vol}(K))^{1/n}. \tag{11.7}$$

Another possible choice for a "size" is the diameter of the set in a given norm.

We require the following properties to be satisfied by a function size$(\cdot)$, which are obviously satisfied for the volume function (11.7).

1. *Monotonicity.* If $K_1 \subseteq K_2$ then: $\text{size}(K_1) \leq \text{size}(K_2)$.
2. *Homogeneity.* For any $\alpha \geq 0$ we have: $\text{size}(\alpha K) = \alpha\,\text{size}(K)$.
3. *Translation Invariance.* For any $x \in \mathbb{R}^n$ we have: $\text{size}(K + x) = \text{size}(K)$.

It appears that under these natural assumptions, we are able to relate the geometry of the localizer with the target objective function. We prove the following result.

Theorem 11.2.1. *Consider general cutting scheme, starting from some $E_0 \supseteq Q$, and preceding as follows, for $k \geq 0$:*

1. *Choose $x_k \in E_k$*
2. *Access the separation oracle: $g_k = \mathcal{O}(x_k)$*
3. *Find $E_{k+1} \supseteq \{\, y \in E_k \;:\; \langle g_k, x_k - y\rangle \geq 0 \,\}$*

Assume that for some $k \geq 0$ we have a small relative size of E_k, for some $0 < \delta < 1$:

$$\text{size}(E_k) \;\leq\; \delta\,\text{size}(Q). \tag{11.8}$$

Set $\bar{x}_k$ to be the point with the smallest function value among strictly feasible points:

$$\bar{x}_k \;:=\; \arg\min\Big\{ f(y) \;:\; y \in \{x_0, \ldots, x_k\} \;\; s.t. \;\; y \in \text{int}\, Q \Big\}. \tag{11.9}$$

Then

$$f(\bar{x}_k) - f^\star \;\leq\; \delta V. \tag{11.10}$$

Proof. Choose $\delta < \gamma \leq 1$ and denote the contracted set

$$Q_\gamma \;\; := \;\; \gamma Q + (1-\gamma)x^\star \;\; \subseteq \;\; Q.$$

We have

$$\operatorname{size}(Q_\gamma) \;\; = \;\; \gamma \operatorname{size}(Q) \;\; > \;\; \delta \operatorname{size}(Q) \;\; \overset{(11.8)}{\geq} \;\; \operatorname{size}(E_k).$$

Hence, $Q_\gamma \not\subseteq E_k$, and we conclude that there exists a point $y = \gamma z + (1-\gamma)x^\star \in Q_\gamma$ for some $z \in Q$ such that

$$y \;\; \notin \;\; E_k.$$

By our construction,

$$E_k \;\; \supseteq \;\; \Big\{ y \in Q \;\; : \;\; \langle g_0, x_0 - y \rangle \geq 0, \ldots, \langle g_k, x_k - y \rangle \Big\}.$$

Therefore, there exists an index $0 \leq i \leq k$ such that one of the separation conditions is violated:

$$\langle g_i, x_i - y \rangle \;\; < \;\; 0. \tag{11.11}$$

Note that due to $y \in Q$, it cannot be a separation oracle from Q. Therefore, $x_i \in \int Q$ and $g_i = f'(x_i) \in \partial f(x_i)$. This reasoning, in particular, ensures that among points $\{x_0, \ldots, x_k\}$ there is at least one from $\operatorname{int} Q$ and $\bar{x}_k$ is well defined in (11.9).

Employing convexity of f, we have

$$f(\bar{x}_k) \;\; \overset{(11.9)}{\leq} \;\; f(x_i) \;\; \overset{(11.11)}{<} \;\; f(x_i) + \langle f'(x_i), y - x_i \rangle \;\; \leq \;\; f(y) \;\; \leq \;\; \gamma f(z) + (1-\gamma) f^\star.$$

Rearranging the terms, we get:

$$f(\bar{x}_k) - f^\star \;\; \leq \;\; \gamma (f(z) - f^\star) \;\; \leq \;\; \gamma V.$$

Taking the limit $\gamma \to \delta$ completes the proof. □

11.3 Ellipsoid Method

We are ready to present the ellipsoid method.

As we see from the previous reasoning, all what we want to do is to construct a next set containing a part of the previous one:

$$E_{k+1} \;\; \supseteq \;\; \Big\{ \; y \in E_k \;\; : \;\; \langle g_k, x_k - y \rangle \geq 0 \; \Big\}, \tag{11.12}$$

and so that $\operatorname{size}(E_k) \to 0$.

For that purpose we use ellipsoids and size to be the volume function (11.7). To ensure (11.12), we use the following geometric lemma.

Lemma 11.3.1. *Let $E \subseteq \mathbb{R}^n$ to be an ellipsoid given by*

$$E \;\; = \;\; \Big\{ y \in \mathbb{R}^n \;\; : \;\; \langle A^{-1}(y-x), y - x \rangle \leq 1 \Big\},$$

where $x \in \mathbb{R}^n$ is its center and $A = A^\top \succ 0$. Consider an arbitrary cut through the center of E given by vector $g \in \mathbb{R}^n$. Then, for

- $x^+ := x - \frac{1}{(n+1)\langle Ag,g\rangle^{1/2}} Ag$
- $A^+ := \frac{n^2}{n^2-1}\Big(A - \frac{2}{(n+1)\langle Ag,g\rangle} Agg^\top A\Big)$

we have the new ellipsoid $E^+ := \{ y \in \mathbb{R}^n \ : \ \langle (A^+)^{-1}(y-x^+), y-x^+\rangle \le 1\}$ *such that*

1. $E^+ \supseteq \{y \in E \ : \ \langle g, x-y\rangle \ge 0\}$ *and*
2. $\mathrm{Vol}(E^+) \le \exp\Big(-\frac{1}{2n}\Big)\,\mathrm{Vol}(E)$.

See, e.g., Section 2.2 in [Bub15], or Section 3.2.8. in [Nes18], for the proof of this lemma.

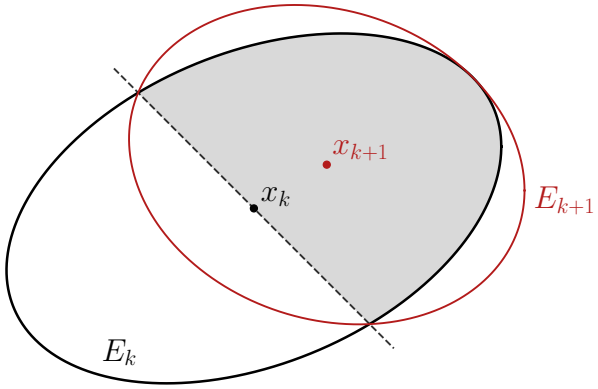

Figure 11.1: One iteration of the ellipsoid method.

11.3.1 Algorithm

Let us first write down our method in the algorithmic form.

Algorithm 11.1: *Ellipsoid Method.*

Initialization: $x_0 \in \mathbb{R}^n$ and $R > 0$ such that $B_R(x_0) \supseteq Q$. Fix $K \ge 1$. Set $A_0 := \frac{1}{R} I$.

For $k = 0 \dots K-1$ **iterate:**

1. Access the separation oracle: $g_k = \mathcal{O}(x_k)$
2. Compute new point: $x_{k+1} = x_k - \frac{1}{(n+1)\langle A_k g_k, g_k\rangle^{1/2}} A_k g_k$
3. Update the matrix: $A_{k+1} = \frac{n^2}{n^2-1}\Big(A_k - \frac{2}{(n+1)\langle A_k g_k, g_k\rangle} A_k g_k g_k^\top A_k\Big)$

Return a point $\bar{x}_K := \arg\min\{f(y) \ : \ y \in \{x_0, \dots, x_K\} \ \text{s.t.} \ y \in \operatorname{int} Q\}$.

Using our previous reasoning, we can prove the following complexity result for this algorithm.

Theorem 11.3.2. *Let* $0 < \varepsilon < V$ *be fixed. In order to achieve* $f(\bar{x}_K) - f^\star \le \varepsilon$ *it is enough to perform*

$$K \ = \ \Big\lfloor 2n^2 \ln \tfrac{RV}{r\varepsilon} \Big\rfloor + 1 \tag{11.13}$$

iterations of Algorithm 11.1 (separation oracle calls).

Proof. From Lemma 11.3.1 we know that

$$\mathrm{Vol}(E_k) \;\leq\; \exp\Bigl(-\tfrac{k}{2n}\Bigr)\,\mathrm{Vol}(E_0). \tag{11.14}$$

Therefore, for our size$(\cdot)$ we have that

$$\mathrm{size}(E_k) \;=\; \mathrm{Vol}(E_k)^{1/n} \overset{(11.14)}{\leq} \exp\Bigl(-\tfrac{k}{2n^2}\Bigr)\,\mathrm{size}(E_0). \tag{11.15}$$

Let us choose $\delta := \frac{\varepsilon}{V} < 1$. Then, by Theorem 11.2.1, we have $f(\bar{x}_K) - f^\star \leq \varepsilon$ as soon as $\mathrm{size}(E_k) \leq \delta\,\mathrm{size}(Q)$. According to (11.15), to achieve this goal it is enough to have

$$\exp\Bigl(-\tfrac{k}{2n^2}\Bigr)\,\mathrm{size}(E_0) \;\leq\; \delta\,\mathrm{size}(Q) \quad\Leftrightarrow\quad k \;\geq\; 2n^2 \ln \tfrac{\mathrm{size}(E_0)}{\delta\,\mathrm{size}(Q)}.$$

It remains to use the upper bound: $\frac{\mathrm{size}(E_0)}{\mathrm{size}(Q)} \leq \frac{R}{r}$ to complete the proof. □

11.3.2 Discussion

We see that the oracle complexity of the ellipsoid method is $O(n^2 \ln \frac{RV}{r\varepsilon})$. Each iteration of the method can be implemented in $O(n^2)$ arithmetic operations (for matrix-vector operations) plus additional cost of perfoming the separation oracle. For linear programming, separation oracle can be easily implemented in $O(nm)$ operations (for dense data) which leads to the total complexity of

$$O\Bigl(n^3(n+m)\ln \tfrac{RV}{r\varepsilon}\Bigr). \tag{11.16}$$

This bound can be used to show the famous result of polynomial solvability of linear programming, as parameters of the problem (such as V, r, R) comes under logarithm, which leads to the polynomial-like dependence on the size of data input.

The complexity result of the ellipsoid method shows the very important fact that

convex optimization is generally solvable.

However, in practice the ellipsoid method is usually less efficient for linear or semidefinite programming than the methods that take into account the structure of the problem, such as *interior-point methods*, which we study in the last part of the course, unless the dimension n is small ($n \approx 10-20$). At the same time, the dependence on m in (11.16) is linear, so we can use the ellipsoid method for solving low-dimensional problems with huge numbers of constraints.

Compared with cheap gradient methods, we see that despite its excellent logarithmic dependence on the target accuracy, the ellipsoid method depends *explicitly on the dimension* n. Consequently, the method does not work as $n \to \infty$ (even in theory), as the formulas in Algorithm 11.1 prevent the method from performing steps in this limit. A modern modification of the ellipsoid method that remains valid as $n \to \infty$ was developed in [Rod22].

Another, more theoretical version of the cutting plane scheme called the *center of gravity method*. At each iteration, we work with the cutting polytope (11.5) directly and define the next query point as the center of gravity of the set G_k:

$$x_{k+1} \;=\; \tfrac{1}{\mathrm{Vol}\,G_k} \int\limits_{G_k} y dy. \tag{11.17}$$

The oracle complexity of such method is

$$O(n \ln \tfrac{1}{\varepsilon})$$

separation oracle calls, which is the *optimal* dependence on n (matching the corresponding lower bound). However, this approach is completely unpractical, as each iteration requires computing an n-dimensional integral (11.17).

See [Nem95] for additional reading on cutting plane methods and their complexity . The general ellipsoid method was developed in 1976 by Arkadi Nemirovski and David Yudin, while a related space dilation subgradient method was introduced by Naum Shor in 1970.

Literature

[Bub15] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and trends in Machine Learning*, 8(3-4):231–357, 2015.

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

[Nes18] Yurii Nesterov. *Lectures on convex optimization.* Springer, 2018.

[Rod22] Anton Rodomanov. *Quasi-Newton methods with provable efficiency guarantees.* PhD thesis, Catholic University of Louvain, Louvain-la-Neuve, Belgium, 2022.