1 Outline

In this lecture, we study

- optimization with functional constraints,
- Nesterov's bisection method,
- Polyak's switching algorithm.

2 Optimization with Functional Constraints

In this lecture, we consider problems of the structure

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & g(x) \leq 0 \\ & x \in \mathcal{X} \end{aligned} \tag{18.1}$$

where $f, g : \mathbb{R}^d \to \mathbb{R}$ are convex functions and $\mathcal{X} \subseteq \mathbb{R}^d$ is a convex set. For simplicity, we assume that both f and g are differentiable. Let x^* denote an optimal solution to (18.1), i.e.,

$$x^* \in \operatorname{argmin}\{f(x) : \ g(x) \leq 0, \ x \in \mathcal{X}\}.$$

The goal is to find an ϵ-approximate solution $\hat{x}$, i.e., a feasible point $\hat{x} \in \mathcal{X}$ such that

$$g(\hat{x}) \leq \epsilon \quad \text{and} \quad f(\hat{x}) - f(x^*) \leq \epsilon.$$

We may attempt to solve (18.1) using projected gradient descent, but this may be inefficient if the feasible set $\{x \in \mathcal{X} : g(x) \leq 0\}$ is complicated. Instead, we will study two alternative methods: Nesterov's bisection method [Nes18] and Polyak's switching algorithm [Pol67].

2.1 Nesterov's Bisection Method

Suppose that we know the optimal value $f^* = f(x^*)$. Then computing an ϵ-approximate solution to (18.1) is equivalent to finding a solution $\hat{x}$ with $h(\hat{x}) \leq \epsilon$ where h is defined as

$$h(x) := \max\{g(x), \ f(x) - f^*\}.$$

Lemma 18.1. $\min_{x \in \mathcal{X}} h(x) = h(x^*) = 0$.

Proof. As $g(x^*) \leq 0$ and $f(x^*) - f^* = 0$, we have $h(x^*) = 0$. For any $x \in \mathcal{X}$, if $g(x) \leq 0$, then by the definition of f^*, we have $f(x) - f^* \geq 0$ and thus $h(x) \geq 0$. Therefore, $\min_{x \in \mathcal{X}} h(x) = h(x^*) = 0$. □

By this lemma, problem (18.1) is equivalent to the problem of minimizing h over $\mathcal{X}$. For minimizing h, we may apply the projected subgradient method to find an ϵ-approximate solution in $O(1/\epsilon^2)$ iterations. Hence, if we know f^*, then we can solve (18.1) in $O(1/\epsilon^2)$ iterations. This in turn

implies that once we obtain a good estimate of f^*, we can solve (18.1) efficiently. Here, the idea behind Nesterov's algorithm is to esimate f^* using a bisection method.

Let us describe the algorithm as follows. For any $v \in \mathbb{R}$, we consider

$$h_v(x) := \max\{g(x),\ f(x) - v\}.$$

Here, we will iteratively update v to approximate f^*. Again, for any $v \in \mathbb{R}$, h_v is a convex function. To run the bisection method, we need lower and upper bounds on f^*. Let ℓ and u be such that

$$l \le f^* \le u.$$

For example, we can set $\ell = \min_{x\in\mathcal{X}} f(x)$ and $u = f(x_0)$ for some feasible solution $x_0 \in \mathcal{X}$ with $g(x_0) \le 0$. The algorithm is as follows:

Algorithm 1 Bisection-based method

Initial lower and upper bounds ℓ, u on f^*, desired accuracy level $\epsilon > 0$.
Set $N = \lceil \log_2 \frac{u-\ell}{\epsilon/2} \rceil$.
for $t = 1, \dots, N$ **do**
  Set $v = (\ell + u)/2$.
  Use the projected subgradient method to find an $\frac{\epsilon}{2}$-approximate solution x_v to $\min_{x\in\mathcal{X}} h_v(x)$.
  if $h_v(x_v) > \epsilon/2$ **then**
    Set $\ell = v$.
  else
    Set $u = v$.
  end if
end for
Return x_u

Lemma 18.2. *For any $v \in \mathbb{R}$, the subdifferential of h_v at x is given by*

$$\partial h_v(x) = \begin{cases} \{\nabla g(x)\} & \text{if } g(x) > f(x) - v \\ \{\nabla f(x)\} & \text{if } g(x) < f(x) - v \\ \operatorname{conv}\{\nabla g(x), \nabla f(x)\} & \text{if } g(x) = f(x) - v \end{cases}.$$

Hence, we can compute a subgradient of h_v at x by evaluating $\nabla f(x)$ and $\nabla g(x)$, which is necessary for implementing the projected subgradient subroutine in Algorithm 1. Assuming Lipschitz continuity of f and g, we know that the projected subgradient method returns an $\epsilon/2$-approximate solution to $\min_{x\in\mathcal{X}} h_v(x)$ in $O(1/\epsilon^2)$ iterations. The following theorem shows the convergence of Algorithm 1.

Theorem 18.3. *After $N = O(\log(1/\epsilon))$ bisection iterations, Algorithm 1 returns an ϵ-approximate solution to (18.1). If each of the N subproblems is solved using the projected subgradient method in $O(1/\epsilon^2)$ iterations, then the total number of iterations is $O((1/\epsilon^2)\log(1/\epsilon))$.*

Proof. We first show that ℓ is always a lower bound on f^*. Initially, we have $\ell \le f^*$. For any $v \in \mathbb{R}$, let $h^*(v)$ denote the optimal value of $\min_{x\in\mathcal{X}} h_v(x)$. Note that $h^*(v)$ is a non-increasing function in v. Moreover, $h^*(v)$ is minimized at $v = f^*$ with $h^*(f^*) = 0$. At the beginning of time step t, suppose that we have $\ell \le f^*$. If $h_v(x_v) > \epsilon/2$, then as x_v is an $\epsilon/2$-approximate solution to

$\min_{x\in\mathcal{X}} h_v(x)$, we have $h^*(v) \geq h_v(x_v) - \epsilon/2 > 0$. This means that $v < f^*$. Hence, for time step $t+1$, we have $\ell = v < f^*$. On the other hand, if $h_v(x_v) \leq \epsilon/2$, then ℓ remains unchanged for time step $t+1$. Therefore, by induction, we have $\ell \leq f^*$ for all time steps.

For u, we show that $h_u(x_u) \leq \epsilon/2$ is always satisfied. For the initial value of u, we know that $f^* \leq u$, so $h^*(u) \leq h^*(f^*) = 0$. As x_u is an $\epsilon/2$-approximate solution to $\min_{x\in\mathcal{X}} h_u(x)$, we have $h_u(x_u) \leq h^*(u) + \epsilon/2 = \epsilon/2$. Now, suppose that at the beginning of time step t, we have $h_u(x_u) \leq \epsilon/2$. If $h_v(x_v) > \epsilon/2$, then u remains unchanged for time step $t+1$, so $h_u(x_u) \leq \epsilon/2$ still holds. On the other hand, if $h_v(x_v) \leq \epsilon/2$, then we set $u = v$, in which case $h_u(x_u) \leq \epsilon/2$ holds for time step $t+1$.

To summarize, for each time step, we maintain

$$\ell \leq f^* \quad \text{and} \quad h_u(x_u) \leq \epsilon/2.$$

After the final time step N, we have $u - \ell \leq \epsilon/2$. This means that

$$g(x_u) \leq \epsilon/2 \quad \text{and} \quad f(x_u) - f^* \leq h_u(x_u) + u - f^* \leq \epsilon/2 + (u - \ell) \leq \epsilon,$$

as required. □

2.2 Polyak's switching Algorithm

Next, we describe Polyak's switching algorithm due to Polyak [Pol67]. Recall that Algorithm 1 requires $O((1/\epsilon^2)\log(1/\epsilon))$ iterations to find an ϵ-approximate solution to (18.1). Here, we incur an additional $O(\log(1/\epsilon))$ factor compared to the optimal iteration complexity of $O(1/\epsilon^2)$ for solving convex optimization problems. The reason is that we need to estimate f^* using a bisection method, which requires $O(\log(1/\epsilon))$ iterations. The algorithm avoids this issue by directly solving (18.1) without estimating f^*, thereby obtaining the optimal iteration complexity of $O(1/\epsilon^2)$.

The main idea is as follows. We apply gradient descent on the constraint function g until its value goes under ϵ. Then we apply gradient descent on the objective function f until the value of g goes above ϵ. We repeat this process until we find an ϵ-approximate solution to (18.1). This algorithm is guaranteed to terminate in $O(1/\epsilon^2)$ iterations.

Algorithm 2 Polyak's Switching Algorithm

Initial solution $x_1 \in \mathcal{X}$, desired accuracy level $\epsilon > 0$, step size $\eta > 0$, total number of iterations T.
for $t = 1, \ldots, T$ **do**
  if $g(x_t) \leq \epsilon$ **then**
    Set $d_t = \nabla f(x_t)$
  else
    Set $d_t = \nabla g(x_t)$.
  end if
  $x_{t+1} = \text{proj}_{\mathcal{X}}(x_t - \eta d_t)$.
end for
Return $\hat{x} = \text{argmin}\{f(x_t) : \ g(x_t) \leq \epsilon, \ t = 1, \ldots, T\}$.

Theorem 18.4. *Suppose that f and g are L-Lipschitz continuous. If we set the step size $\eta = \epsilon/L^2$ and run Algorithm 2 for $T = O(1/\epsilon^2)$ iterations, then it returns an ϵ-approximate solution to (18.1).*

Proof. Note that for any t, we have $\|d_t\| \leq L$. By the non-expansiveness of the projection operator, we have

$$\begin{aligned}\|x_{t+1} - x^*\|^2 &= \|\mathrm{proj}_{\mathcal{X}}(x_t - \eta d_t) - \mathrm{proj}_{\mathcal{X}}(x^*)\|^2 \\ &\leq \|x_t - \eta d_t - x^*\|^2 \\ &= \|x_t - x^*\|^2 - 2\eta d_t^\top (x_t - x^*) + \eta^2 \|d_t\|^2 \\ &\leq \|x_t - x^*\|^2 - 2\eta d_t^\top (x_t - x^*) + \eta^2 L^2.\end{aligned}$$

Then it follows that

$$\frac{1}{T}\sum_{t=1}^{T} d_t^\top (x_t - x^*) \leq \frac{1}{2\eta T}\|x_1 - x^*\|_2^2 + \frac{\eta}{2}L^2.$$

This implies that

$$\min_{1\leq t\leq T} d_t^\top (x_t - x^*) \leq \frac{1}{2\eta T}\|x_1 - x^*\|_2^2 + \frac{\eta}{2}L^2.$$

For $t \in \{1, \ldots, T\}$, if $g(x_t) \leq \epsilon$, then by the convexity of f, we have $d_t = \nabla f(x_t)$ and

$$f(x_t) - f(x^*) \leq \nabla f(x_t)^\top (x_t - x^*) = d_t^\top (x_t - x^*).$$

On the other hand, if $g(x_t) > \epsilon$, then by the convexity of g, we have $d_t = \nabla g(x_t)$ and

$$g(x_t) - g(x^*) \leq \nabla g(x_t)^\top (x_t - x^*) = d_t^\top (x_t - x^*).$$

As $g(x^*) \leq 0$, we have

$$\epsilon < g(x_t) \leq d_t^\top (x_t - x^*) + g(x^*) \leq d_t^\top (x_t - x^*).$$

We have that for each t, combining the above two cases, we have

$$d_t^\top (x_t - x^*) \geq \min\left\{\epsilon, \min\left\{f(x_t) - f(x^*) :\ g(x_t) \leq \epsilon,\ t = 1, \ldots, T\right\}\right\}.$$

This implies that

$$\begin{aligned}\min\left\{\epsilon, \min\left\{f(x_t) - f(x^*) :\ g(x_t) \leq \epsilon,\ t = 1, \ldots, T\right\}\right\} &\leq \min_{1\leq t\leq T} d_t^\top (x_t - x^*) \\ &\leq \frac{1}{2\eta T}\|x_1 - x^*\|_2^2 + \frac{\eta}{2}L^2.\end{aligned}$$

Setting $\eta = \epsilon/L^2$ and

$$T \geq \frac{2L^2}{\epsilon^2}\|x_1 - x^*\|_2^2,$$

we have

$$\min\left\{\epsilon, \min\left\{f(x_t) - f(x^*) :\ g(x_t) \leq \epsilon,\ t = 1, \ldots, T\right\}\right\} \leq \frac{1}{4}\epsilon + \frac{1}{2}\epsilon < \epsilon.$$

This means that

$$\min\left\{f(x_t) - f(x^*) :\ g(x_t) \leq \epsilon,\ t = 1, \ldots, T\right\} < \epsilon.$$

Hence, the output $\hat{x}$ of Algorithm 2 is an ϵ-approximate solution to (18.1), as required. □

References

[Nes18] Yurii Nesterov. *Lectures on Convex Optimization*. Springer Publishing Company, Incorporated, 2nd edition, 2018. 2

[Pol67] B.T. Polyak. A general method for solving extremal problems. *Dokl. Akad. Nauk SSSR*, 174(1):33–36, 1967. 2, 2.2