IE 539: Convex Optimization **KAIST, Fall 2022**
Lecture #12: Constrained optimization October 11, 2022
Lecturer: Dabeen Lee

1 Outline

In this lecture, we study

- Projected gradient descent,
- Conditional gradient method,
- Lower bounds on the iteration complexity of gradient methods.

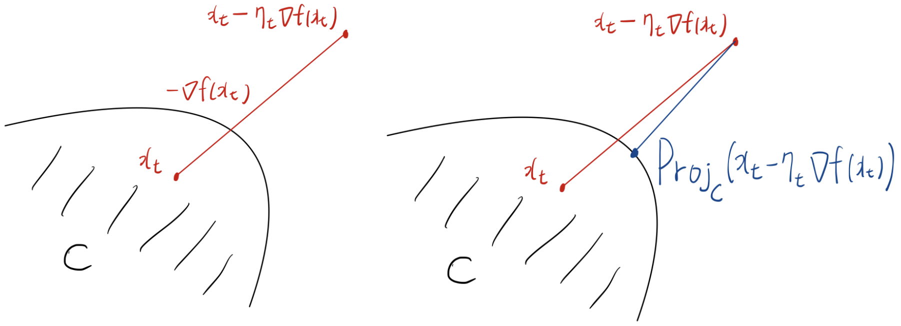
2 Projected gradient descent

In some of our previous lectures that the gradient decesnt method converges to an optimal solution of an unconstrained convex minimization problem under various settings. The gradient descent method updates the iterate by the rule

$$x_{t+1} = x_t - \eta_t \nabla f(x_t)$$

where x_t is the tth iterate, η_t is the step size for iteration t, and $\nabla f(x_t)$ is the gradient of the objective function f at x_t. If f is not differentiable, we may take a subgradient $g \in \partial f(x_t)$ at x_t instead of the gradient.

For constrained optimization, however, the update rule does not necessarily generate a feasible solution. A natural fix for this is that we take the projection of the point $x_t - \eta_t \nabla f(x_t)$ onto the feasible set C. What we have just discussed is basically the projected gradient descent method! It is basically gradient descent with projection. To formalize, let us give a pseudo-code of the projected gradient descent method. In Algorithm 1, we use the operator $\mathrm{Proj}_C(\cdot)$, which is formally defined as



$x_t - \eta_t \nabla f(x_t)$
$-\nabla f(x_t)$
x_t
C
$x_t - \eta_t \nabla f(x_t)$
x_t
$\mathrm{Proj}_C(x_t - \eta_t \nabla f(x_t))$
C

Figure 12.1: Infeasible point after a gradient descent update and projection

$$\mathrm{Proj}_C(z) = \operatorname*{argmin}_{x \in C} \frac{1}{2}\|x - z\|_2^2 \quad \text{for } z \in \mathbb{R}^d.$$

Then it is straightforward that

$$\mathrm{Proj}_C(z) = \operatorname*{argmin}_{x \in C} \|x - z\|_2,$$

Algorithm 1 Projected gradient descent method

```
Initialize $x_1 \in C$.
for $t = 1, \ldots, T$ do
    $x_{t+1} = \mathrm{Proj}_C\{x_t - \eta_t \nabla f(x_t)\}$ for a step size $\eta_t > 0$.
end for
Return $x_{T+1}$.
```

and in words, $\mathrm{Proj}_C(z)$ is a point C that is closest to point z with respect to the ℓ_2 norm distance. Although we have discussed the following lemma in a previous lecture, we include it again to make this note self-contained.

Lemma 12.1. *Let $x \in C$ and $z \in \mathbb{R}^d$. Then*

$$(Proj_C(z) - z)^\top (Proj_C(z) - x) \leq 0 \quad \textit{for all } x \in C.$$

Proof. We can apply the optimality condition to the definition $\mathrm{Proj}_C(z) = \mathrm{argmin}_{x \in C} \frac{1}{2}\|x - z\|_2^2$ for $z \in \mathbb{R}^d$. The gradient of $\frac{1}{2}\|x - z\|_2^2$ at $x = \mathrm{Proj}_C(z)$ is $(\mathrm{Proj}_C(z) - z)$. Then the statement is precisely the optimality condition for $\mathrm{Proj}_C(z)$. □

By definition, x_{t+1} is the point in C that is closest to $x_t - \eta_t \nabla f(x_t)$ with respect to the ℓ_2 distance. Moreover, we have another interpretation of the update rule based on the following.

$$\begin{aligned} x_{t+1} &= \operatorname*{argmin}_{x \in C} \left\{ \frac{1}{2}\|x - x_t + \mu_t \nabla f(x_t)\|_2^2 \right\} \\ &= \operatorname*{argmin}_{x \in C} \left\{ f(x_t) + \nabla f(x_t)^\top (x - x_t) + \frac{1}{2\eta_t}\|x - x_t\|_2^2 \right\}, \end{aligned}$$

which means that x_{t+1} is the solution in C minimizing the quadratic approximation of f at x_t.

Hereinafter, we introduce notations y_{t+1} to denote $x_t - \eta_t \nabla f(x_t)$ for simpler presentations. Then the update rule can be written as

$$\begin{aligned} y_{t+1} &= x_t - \eta_t \nabla f(x_t), \\ x_{t+1} &= \mathrm{Proj}_C(y_{t+1}) \end{aligned}$$

for $t = 1, \ldots, T$. The analysis of projected gradient descent is quite similar to that of gradient descent for unconstrained minimization. The following is useful to make the analysis for gradient descent go through for the case of projected gradient descent.

Lemma 12.2. *For any t, we have*

$$\|x_{t+1} - x^*\|_2 \leq \|y_{t+1} - x^*\|_2$$

where x^* is an optimal solution to $\min_{x \in C} f(x)$.

Proof. We use Lemma 12.1 and the fact that $x_{t+1} = \mathrm{Proj}_C(y_{t+1})$. By Lemma 12.1,

$$(x_{t+1} - y_{t+1})^\top (x_{t+1} - x^*) \leq 0.$$

Since $x_{t+1} - y_{t+1} = x_{t+1} - x^* + x^* - y_{t+1}$, the inequality implies that

$$\|x_{t+1} - x^*\|_2^2 \leq (y_{t+1} - x^*)^\top (x_{t+1} - x^*) \leq \|y_{t+1} - x^*\|_2 \|x_{t+1} - x^*\|_2$$

where the last inequality is due to the Cauchy-Schwarz inequality. Dividing each side by $\|x_{t+1} - x^*\|_2$, we obtain the result. □

By Lemma 12.2, we deduce that

$$\begin{aligned}\|x_{t+1}-x^*\|_2^2 &\leq \|y_{t+1}-x^*\|_2\\ &= \|x_t-x^*\|_2^2 - 2\eta_t \nabla f(x_t)^\top (x_t-x^*) + \eta_t^2 \nabla f(x_t)^2\\ &\leq \|x_t-x^*\|_2^2 - 2\eta_t (f(x_t)-f(x^*)) + \eta_t^2 \nabla f(x_t)^2,\end{aligned}$$

which appears in the convergence analysis of gradient descent for Lipschitz continuous functions. Note that the only difference from the unconstrained case is the first inequality, which used to be an equality for the unconstrained case where $y_{t+1} = x_{t+1}$. Based on this, we recover the same convergence theorem for projected gradient descent for the case of Lipschitz continuous functions. In fact, we can work over the projected subgradient method, which is as the name suggests the subgradient method with projection for the constrained minimization.

Algorithm 2 Projected subgradient method

```
Initialize $x_1 \in C$.
for $t = 1, \dots, T$ do
    Obtain a subgradient $g_t \in \partial f(x_t)$.
    $x_{t+1} = \mathrm{Proj}_C \{x_t - \eta_t g_t\}$ for a step size $\eta_t > 0$.
end for
Return $(\sum_{t=1}^T \eta_t)^{-1} \sum_{t=1}^T \eta_t x_t$.
```

The following theorem shows the convergence of the projected subgradient method for functions that have bounded subgradients.

Theorem 12.3. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function such that $\|g\|_2 \leq L$ for any $g \in \partial f(x)$ for every $x \in \mathbb{R}^d$. Let $\{x_t : t = 1, \dots, T\}$ be the sequence of iterates generated by the projected subgradient method with step size $\eta_t = \|x_1 - x^*\|_2 / L\sqrt{T}$ for each t. Then*

$$f\left(\frac{1}{T}\sum_{t=1}^T x_t\right) - f(x^*) \leq \frac{L\|x_1-x^*\|_2}{\sqrt{T}}$$

where x^* is an optimal solution to $\min_{x\in C} f(x)$.

Moreover, we also recover the same "asymptotic" convergence rate for strongly convex, smooth, and strongly convex & smooth functions. In particular,

Theorem 12.4. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a β-smooth convex function, and let $\{x_t : t = 1, \dots, T\}$ be the sequence of iterates generated by gradient descent with step size $\eta_t = 1/\beta$ for each t. Then*

$$f(x_T) - f(x^*) \leq \frac{3\beta\|x_1-x^*\|_2^2 + f(x_1) - f(x^*)}{T}$$

where x^* is an optimal solution to $\min_{x\in C} f(x)$.

3 Conditional gradient method

We saw that the projected gradient descent minimizes a smooth function with a convergence rate of $O(1/T)$. There are a couple of issues.

1. The projection step onto the feasible set C can be expensive.
2. We have used the ℓ_2 norm to define smoothness.

Each projection step essentially amounts to solving an optimization problem, which can be difficult depending on the structure of C. Even for the case when C is a polyhedron, the projection onto C can an expensive procedure. The second point is that in our analysis of gradient descent for smooth functions, there are parts that do need smoothness with respect to the ℓ_2 norm. It is often the case that smoothness in the ℓ_2 norm is implied by smoothness in another norm, e.g., the ℓ_1 norm.

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq \sqrt{d}\|\nabla f(x) - \nabla f(y)\|_\infty \leq \sqrt{d}\beta\|x-y\|_1 \leq d\beta\|x-y\|_2.$$

The implication of this inequality is the following. Even if a function is smooth in the ℓ_1 norm with a tiny smoothness parameter β, the smoothness parameter with respect to the ℓ_2 norm can blow up by a factor of dimension d, in which case we lose the desired dimension-free property.

Motivated by these two issues, we consider the conditional gradient method, introduced by Frank and Wolfe in 1956 [FW56]. Named after the author, the conditional gradient method is often referred to as the Frank-Wolfe algorithm. A pseudo-code of the method is given as follows.

Algorithm 3 Frank-Wolfe algorithm

```
Initialize $x_1 \in C$.
for $t = 1, \dots, T-1$ do
    Take $v_t \in \mathrm{argmin}_{v \in C} \nabla f(x_t)^\top v$.
    Update $x_{t+1} = (1-\lambda_t)x_t + \lambda_t v_t$ for some $0 < \lambda_t < 1$.
end for
Return $x_T$.
```

The main component of the conditional gradient method is to compute the direction v_t by solving

$$\min_{v \in C} \nabla f(x_t)^\top v$$

whose objective is a linear function. In particular, when C is a polyhedron, it is just a linear program. This is in contrast to the projected gradient descent which has a quadratic objective for each projection step. For this reason, the conditional gradient method is called “projection-free”.

Another difference compared to the projected gradient descent is that the direction we take for an update can be different from $-\nabla f$. We provide Figure 12.2 for a pictorial description of the update rule. v_t is a point up to which we can move as far as we can in the direction of $-\nabla f(x_t)$ within C. Then we take a convex combination of the current point x_t and v_t to obtain the new iterate x_{t+1}.

Definition 12.5. We say that a differentiable function $f : \mathbb{R}^d \to \mathbb{R}$ is *β-smooth* with respect to a norm $\|\cdot\|$ for some $\beta > 0$ if

$$\|\nabla f(x) - \nabla f(y)\|_* \leq \beta\|x-y\|$$

holds for any $x, y \in \mathbb{R}^d$ where $\|\cdot\|_*$ denotes the dual norm of $\|\cdot\|$.

The next theorem shows that conditional gradient descent converges with rate $O(1/T)$ for any smooth function with repsect to an arbitrary norm.

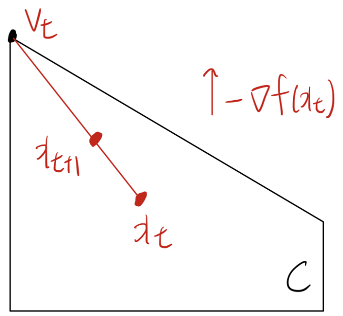


Figure 12.2: Illustration of an update from conditional gradient descent

Theorem 12.6. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function that is β-smooth with respect to a norm $\|\cdot\|$ for some $\beta > 0$. Let $\{x_t : t = 1, \dots, T\}$ be the sequence of iterates generated by conditional gradient descent with $\lambda_t = 2/(t+1)$ for each t. Then for any $t \geq 2$,*

$$f(x_t) - f(x^*) \leq \frac{2\beta R^2}{t+1}$$

where x^* is an optimal solution to $\min_{x\in C} f(x)$ and $R = \sup_{x,y\in C}\|x-y\|$.

Proof. Note that

$$\begin{aligned} f(x_{t+1}) - f(x_t) &\leq \nabla f(x_t)^\top (x_{t+1} - x_t) + \frac{\beta}{2}\|x_{t+1} - x_t\|^2 \\ &= \lambda_t \nabla f(x_t)^\top (v_t - x_t) + \frac{\beta}{2}\|x_{t+1} - x_t\|^2 \\ &\leq \lambda_t \nabla f(x_t)^\top (x^* - x_t) + \frac{\beta}{2}\|x_{t+1} - x_t\|^2 \\ &\leq \lambda_t (f(x^*) - f(x_t)) + \frac{\beta}{2}\|x_{t+1} - x_t\|^2 \end{aligned}$$

where the first inequality is from the β-smoothness of f, the first equality follows from $x_{t+1} = (1-\lambda_t)x_t + \lambda_t v_t$, the second inequality is due to the definition of $v_t = \operatorname{argmin}_{v\in C} \nabla f(x_t)^\top v$, and the last inequality is by the convexity of f. Since

$$\|x_{t+1} - x_t\| = \lambda_t \|v_t - x_t\| \leq \lambda_t R,$$

it follows that

$$\begin{aligned} f(x_{t+1}) - f(x^*) &\leq (1-\lambda_t)(f(x_t) - f(x^*)) + \frac{\beta \lambda_t^2 R^2}{2} \\ &= \frac{t-1}{t+1}(f(x_t) - f(x^*)) + \frac{2\beta R^2}{(t+1)^2}. \end{aligned}$$

By this inequality, it follows that

$$f(x_2) - f(x^*) \leq \frac{\beta R^2}{2} \leq \frac{2\beta R^2}{3}.$$

Then by the induction hypothesis,

$$f(x_{t+1}) - f(x^*) \leq \frac{2(t-1)+2}{(t+1)^2}\beta R^2 = \frac{t}{(t+1)^2} 2\beta R^2 \leq \frac{1}{t+2}\beta R^2,$$

as required. □

4 Lower bounds on the iteration complexity of gradient methods

We learned and analyzed the convergence rate of gradient descent and the subgradient method. In particular, for Lipschitz continuous functions, we know that the subgradient method guarantees the convergence rate of $O(1/\sqrt{T})$ and requires $O(1/\epsilon^2)$ iterations to achieve the error bounded by ϵ. For smooth convex functions, gradient descent achieves $O(1/T)$ convergence rate, and the number of required iterations to bound the error by ϵ is $O(1/\epsilon)$. For functions that are both smooth and strongly convex, the convergence rate of gradient descent is $O(\gamma^T)$ for some $0 < \gamma < 1$, and the number of required iterations is $O(\log(1/\epsilon))$ to achieve an error of ϵ.

A natural question is as to whether we can find an algorithm that achieves a better convergence rate. Regarding this question, we conceptualize the oracle complexity of an algorithm. An oracle for convex minimization $\min_{x\in C} f(x)$ takes a point x in C as an input and returns its function value $f(x)$ as well as the first-order information, i.e., the gradient $\nabla f(x)$ or a subgradient $g_t \in \partial f(x)$. Then the oracle complexity of an oracle-based algorithm counts the number of oracle calls to terminiate. An oracle-based algorithm can be illustrated as follows. Basically, it picks a new solution based on the history of past iterates and their first-order information.

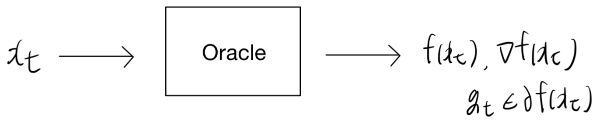


Figure 12.3: Oracle that returns the function value and the first-order information

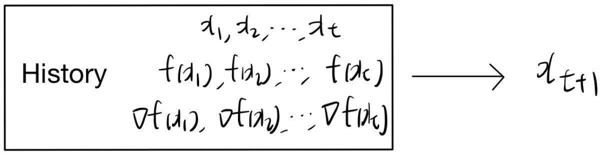

$x_1, x_2, \cdots, x_t$
History
$f(x_1), f(x_2), \cdots, f(x_t)$
$\nabla f(x_1), \nabla f(x_2), \cdots, \nabla f(x_t)$
x_{t+1}

Figure 12.4: Illustration of an oracle-based algorithm

We present lower bound results on the oracle complexity given by Nemirovski and Yudin in 1983 [NY83] (see also Nesterov [Nes03] and Bubeck [Bub15]). We make the assumption that $x_1 = 0$ and x_{t+1} belongs to the span of $g_1, \ldots, g_t$ where $g_s \in \partial f(x_s)$.

Theorem 12.7 (See [Bub15]). *There exists a convex and L-Lipschitz continuous function $f : \mathbb{R}^d \to \mathbb{R}$ for some $L > 0$ such that iterates $x_1, \ldots, x_t$ with $t \leq d$ generated by any oracle-based algorithm satisfies the following:*

$$\min_{1\leq s\leq t} f(x_s) - \min_{x\in B_2(R)} f(x) \geq \frac{RL}{2(1+\sqrt{t})}$$

where $B_2(R) = \{x \in \mathbb{R}^d : \|x\|_2 \leq R\}$ and $R > 0$.

Theorem 12.8 (See [Bub15]). *There exists a convex and β-smooth fuction $f : \mathbb{R}^d \to \mathbb{R}$ with respect to the ℓ_2-norm for some $\beta > 0$ such that iterates $x_1, \ldots, x_t$ with $t \leq (d-1)/2$ generated by any oracle-based algorithm satisfies the following:*

$$\min_{1\leq s\leq t} f(x_s) - \min_{x\in\mathbb{R}^d} f(x) \geq \frac{3\beta\|x_1 - x^*\|_2^2}{32(t+1)^2}.$$

Theorem 12.9 (See [Bub15])**.** *There exists a β-smooth and α-strongly convex fuction $f : \mathbb{R}^d \to \mathbb{R}$ with respect to the ℓ_2-norm for some $\beta \geq \alpha > 0$ such that x_t with $t \geq 1$ generated by any oracle-based algorithm satisfies the following:*

$$f(x_t) - \min_{x \in \mathbb{R}^d} f(x) \geq \frac{\alpha}{2} \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{2(t-1)} \|x_1 - x^*\|_2^2.$$

References

[Bub15] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3–4):231–357, 2015. 4, 12.7, 12.8, 12.9

[FW56] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956. 3

[Nes03] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003. 4

[NY83] Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983. 4