EE C227C Convex Optimization and Approximation Lecture 5 - 2/1/2022

Lecture 5: Mirror Descent

Lecturer: Jiantao Jiao *Scribe: Huong Vu, Zheng Liang*

1 Recap

Last week, we talked about Gradient Descent and Sub-Gradient method. There was a big confusion in Gradient Descent theory because in term of Gradient Descent analysis, if the gradient is larger, we make more progress; in term of Sub-Gradient Descent method, if the gradient is smaller, we make more progress. Until today, we haven't completely understood the confusion. A good idea for this confusion is Linear Coupling which somehow combines the two concepts.

In this lecture, we will learn more about what Sub-Gradient Descent does by understanding Mirror Descent.

2 Mirror Descent

In Mirror Descent, we assume function $f(x)$ satisfy bounded gradient condition i.e. there exists a $G > 0$ such that for all $x \in K$,

$$\|\nabla f(x)\|_* \leq G \tag{1}$$

Note:

- we are not assuming gradient is Lipschitz here. In Sub-Gradient method, we also didn't have a gradient; we just said the given vector has bounded norm.
- If the gradient exists, then the gradient is bounded.
- G plays similar role as L in previous lectures.
- $\|\cdot\|_*$ is not necessarily $l2$ norm. In Mirror Descent, it is very important to choose the right norm for the gradient.

We also assume good initialization: $\|x^0 - x^*\| \leq D$ in Mirror Descent.

Let's consider a special case where $\|\cdot\| = \|\cdot\|_2 = \|\cdot\|_*$. Suppose we have access to gradient and run $x^{t+1} = x^t - \eta\nabla f(x^t)$. Combining with the above assumptions, we have a guarantee as following:

$$f\left(\frac{1}{T}\sum_{i=0}^{T-1} x^t\right) - f(x^*) \leq \frac{1}{\eta T}\left(D^2 + T\frac{(\eta G)^2}{2}\right) \tag{2}$$

Unlike in the last lecture in which we proved a result about the best iterate, we have a convergence guarantee for the average of all iterates here.

Intuition: Even though in Sub-Gradient Method, the objective function might not decrease following the negative direction of the gradient, this method still works because we can construct a so-called potential function that actually decreases. This potential function is related to the distance between the current iterate point and the optimal point.

3 Regularization View of Optimization/Local Model View

We want to solve the following problem

$$\min_{x \in K} f(x) \tag{3}$$

where f is very complicated.

First, we can ask the oracle to get $(x_1, \nabla f(x_1)), (x_2, \nabla f(x_2)), (x_3, \nabla f(x_3)), \cdots$. Then we use the collected information to create a local model of $f(x)$ near x^t called $f_t(x)$ such that

$$x^{t+1} = \arg\min_{x \in K} f_t(x) \tag{4}$$

Properties of $f_t(x)$:

- Approximate $f(x)$ "well"
- Easy to optimize

Intuition: For a very complicated function $f(x)$ that we don't know how to optimize, if we can optimize at each step, then the number of iterations is very small.

3.1 View Gradient Descent as Local Model

We will follow Gradient Descent framework which assumes Lipschitz-gradient condition in this section. Keep in mind that the Lipschitz continuous gradient condition may not imply the bounded gradient condition. For instance, it may be the case that $G = \mathcal{O}(1)$, but there is no such bound on the Lipschitz constant of the gradient of f. Construct the following local approximation function

$$f_t(x) = f(x^t) + \langle \nabla f(x^t), x - x^t \rangle + \frac{L}{2} \|x - x^t\|_2^2 \tag{5}$$

Observation: With x^t fixed, $f_t(x)$ is a quadratic function in x and is easy to optimize. To solve $\arg\min_{x \in K} f_t(x)$, we take the derivative of $f_t(x^{t+1})$

$$0 = \nabla f_t(x^{t+1}) = \nabla f(x^t) + L(x^{t+1} - x^t) \tag{6}$$

Solving for x^{t+1} gives us $x^{t+1} = x^t - \frac{1}{L} \nabla f(x^t)$.
Observation: This is one form of Gradient Descent in the sense that we need to be careful with the choice of learning rate to be not too large and we also have $\frac{L}{2}$ in $f_t(x)$ which matches the definition in Gradient Descent.

When $L = \infty$, we can suppose no Lipschitz-gradient and let

$$f_t(x) = f(x_t) + \langle \nabla f(x_t), x - x_t \rangle \tag{7}$$

Then, we can try to approach the problem by applying the update rules as follows:

$$x^{t+1} := \arg\min_{x \in K} f(x^t) + \langle \nabla f(x^t), x - x^t \rangle \tag{8}$$

A downside of the above is that it is very aggressive - in fact, the new point x^{t+1} could be very far away from x^t. This is illustrated by considering $K = [-1, 1]$ and $f(x) = x^2$. If started at $x = 1$, and update using (8), the algorithm jumps indefinitely between -1 and 1. This is because one of these two points is always a minimizer of a linear lower bound of f over K. Thus, the sequence ${x^t}_{t \geq 0}$ never reaches 0 - the unique minimizer of f [1]. Indeed, $x = 1$ locally is a good local approximation to the original function. However, the idea of finding $\arg\min$ over the whole domain implies the $x = 1$ is a good global approximation while it is not. So, we can adjust the idea by only taking arg min over local neighborhood. The hardest part of this approach is to define the local neighborhood precisely.

From the example of $f(x) = x^2$, intuitively, we know that we need to move locally and respect the local information at the same time.

General paradigm of Mirror Descent is to add a regularizer. Let

$$\begin{aligned} x^{t+1} &= \arg\min_{x\in K}(D(x, x^t) + \eta(f(x^t) + \langle \nabla f(x^t), x - x^t\rangle)) \qquad (9) \\ &= \arg\min_{x\in K}(D(x, x^t) + \eta\langle \nabla f(x^t), x\rangle) \qquad (10) \end{aligned}$$

Geometric Interpretation: We want to choose a particular point x over the domain K but still respect two constraints:

- We want x to not move too far away from x_t i.e. we want to stay locally. This is controlled by $D(x, x^t)$.
- We want x to go along the negative direction of $\nabla f(x^t)$.

In Sub-Gradient method, we also follow this idea with $D(x, x^t) = \frac{1}{2}\|x - x^t\|_2^2$. In general, it is usually hard to choose the right distance function.

4 Exponential Gradient Descent

Consider the n-dimensional **probability simplex**

$$\Delta_n := \{p \in [0, 1]^n : \sum_{i=1}^{n} p_i = 1\}$$

and a convex optimization problem

$$\min_{p\in\Delta_n} f(p) \qquad (11)$$

From previous sections, we know that the general form of an algorithm we would like to construct is

$$p^{t+1} := \arg\min_{p\in\Delta_n} \{D(p, p^t) + \eta\langle\nabla f(p^t), p\rangle\} \qquad (12)$$

Here, D is a distance function on the probability simplex Δ_n.

Recall: The two conditions we have for the approximating function is that (1) f_t approximates f well; (2) f_t is easy to optimize. Therefore, there are only a few selected distance function D we would consider, although in this course we want to present a general theory.

Note: In Mirror Descent, there is an important notion of favorable geometry. If your domain is of certain shape, then the optimization can be done faster. If the domain is of another shape, the optimization will be slower. In previous lecture, we have mentioned that it would be great to have the domain of Euclidean ball or simplex as the domain but if the domain is an ℓ_∞ ball, then it is terrible.

4.1 Kullback-Leibler Divergence

Here we will introduce the Kullback-Leibler divergence as an important example of distance functions. Then we will extend it to a generalized form and demonstrate some important properties. The definition of the Kullback-Leibler divergence (KL divergence) over the probability simplex Δ_n is as follows:

$$D_{KL}(p,q) := \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i} \tag{13}$$

For the corner cases where $p_i = 0$, we use $\lim_{x\to 0^+} x\log(x) = 0$ to obtain the function value (which means the i-th term is 0). For any $q_i = 0$, we require that $p_i = 0$ too. Sometimes we also write $D_{KL}(p,q)$ as $D_{KL}(p \parallel q)$ since p and q can be viewed as two probability distributions. KL divergence has many important properties. Here is an important property we will use latter.

Lemma 1. *For any p and $q \in \Delta_n$, we have*

$$D_{KL}(p,q) \geq 0, \ \textit{and}\ D_{KL}(p,q) = 0 \iff p = q \tag{14}$$

We can extend the definition of KL divergence to D_H on $\mathbb{R}^n_{\geq 0}$. For any x, $y \in \mathbb{R}^n_{\geq 0}$,

$$D_H(x,y) = \sum_{i=1}^{n} x_i \log \frac{x_i}{y_i} + \sum_{i=1}^{n} (y_i - x_i) \tag{15}$$

If we restrict this function to the probability simplex Δ_n, it will become the KL divergence D_{KL}. We need to notice that even if x or y is not restricted to the probability simplex, we still have $D_H \geq 0$.

4.2 Algorithm Implementation

After introducing the background, we will discuss the algorithm implementation of EGD in this part. The pseudo-codes are shown below.

Algorithm 1 Exponential gradient descent (EGD)

```
$p^0 = \frac{1}{n}\mathbb{1}$
for $t = 0, 1, ..., T-1$ do
    $g^t := \nabla f(p^t)$
    $w_i^{t+1} := p_i^t \exp(-\eta g_i^t)$
    $p_i^{t+1} := w_i^{t+1} / \|w^{t+1}\|_1$
end for
$\bar{p} := \frac{1}{T}\sum_{t=0}^{T-1} p^t$
```

The inputs of the EGD algorithm are

1. The first-order oracle of the convex function $f : \Delta_n \to \mathbb{R}$;
2. The step size (or learning rate in machine learning background) η;
3. The time steps T.

The final output of the EGD algorithm is $\bar{p}$. The initial solution is set to p^0, which is the uniform distribution. At each time step t, we calculate the gradient g^t of the function f, update each entry of p with an exponential factor $e^{-\eta g_i^t}$ to get w, and normalize w to get p^{t+1} (the solution of the next time step). Finally, we obtain the approximated solution $\bar{p}$ by averaging all intermediate solutions p_i.

4.3 Asymptotic Analysis

In this part, we will explore the asymptotic time complexity of EGD given an error bound ϵ. One important theorem is the accuracy guarantees of EGD. The theorem is as follows:

Theorem 2. *Suppose that $f : \Delta_n \to \mathbb{R}$ is a convex function, $p^\star$ is the minimizer of the convex function $f(\cdot)$ over the set Δ_n, and $\|\nabla f(p)\|_{+\infty} \leq G$ for all $p \in \Delta_n$ [1], given an error bound ϵ, if we let*

$$\eta := \Theta\left(\frac{\sqrt{\log(n)}}{G\sqrt{T}}\right); \quad T := \Theta\left(\frac{G^2 \log(n)}{\epsilon^2}\right), \tag{16}$$

then we can guarantee that the final approximation error is bounded by ϵ.

$$f(\bar{p}) - f(p^\star) \leq \epsilon \tag{17}$$

To prove this theorem, we will introduce several lemmas first.

Lemma 3. *Consider an arbitrary vector $q \in \mathbb{R}^n_{\geq 0}$, and another arbitrary vector $g \in \mathbb{R}^n$, the following optimization problem*

$$w^\star := \arg\min_{w \in \mathbb{R}^n_{\geq 0}} \left(D_H(w, q) + \eta\langle g, w\rangle\right) \tag{18}$$

has a closed-form solution

$$w_i^\star = q_i \exp(-\eta g_i) \tag{19}$$

And we have a similar lemma where the variable to be optimized is in Δ_n.

Lemma 4. *Consider an arbitrary vector $q \in \mathbb{R}^n_{\geq 0}$, another arbitrary vector $g \in \mathbb{R}^n$, the following optimization problem*

$$p^\star := \arg\min_{p \in \Delta_n} \left(D_H(p, q) + \eta\langle g, p\rangle\right) \tag{20}$$

has a closed-form solution

$$p_i^\star = w_i^\star / \|w\|_1$$

Lemma 5. *For any p and the uniform distribution p^0 in the probability simplex Δ_n, we have*

$$D_{KL}(p, p^0) \leq \log(n) \tag{21}$$

Proof The proof of this lemma is as follows.

$$\begin{aligned} D_{KL}(p, p^0)) &= \sum_{i=1}^n p_i \log\left(\frac{p_i}{1/n}\right) \\ &= \sum_{i=1}^n p_i \log(p_i) + \sum_{i=1}^n p_i \log(n) \\ &= \sum_{i=1}^n p_i \log(p_i) + \log(n) \end{aligned}$$

[1] $\|x\|_{+\infty}$ is the supremum norm, which equals to $\max_{i \in \{1,..,n\}} |x_i|$

Since $p_i \in [0,1]$, we have $p_i \log(p_i) \leq 0$, thus $\sum_{i=1}^{n} p_i \log(p_i) \leq 0$ and $\sum_{i=1}^{n} p_i \log(p_i) + \log(n) \leq \log(n)$. □

Lemma 6. *For the $\bar{p}$ obtained from the EGD algorithm and any point $p' \in \Delta_n$, we have*

$$f(\bar{p}) - f(p') \leq \frac{1}{T} \sum_{t=0}^{T-1} \langle g^t, p^t - p' \rangle \tag{22}$$

Proof To prove this, we can firstly relax the left hand side using $f(\bar{p}) \leq \frac{1}{T} \sum_{t=0}^{T-1} f(p^t)$ (Jensen's inequality, $f(\cdot)$ is convex, and $\bar{p}$ is obtained by averaging $p^0, ..., p^{T-1}$), and the fact that $f(p') \geq f(p) + \langle \nabla f(p), p' - p \rangle$ for any p and p' in the domain Δ_n (a property of convex functions) in these steps:

$$\begin{aligned} f(\bar{p}) - f(p') &\leq \frac{1}{T} \sum_{t=0}^{T-1} f(p^t) - f(p') \\ &= \frac{1}{T} \sum_{t=0}^{T-1} (f(p^t) - f(p')) \\ &\leq \frac{1}{T} \sum_{t=0}^{T-1} \langle \nabla f(p^t), p^t - p' \rangle \\ &= \frac{1}{T} \sum_{t=0}^{T-1} \langle g^t, p^t - p' \rangle \end{aligned}$$

□

This inequality shows that the final error $f(\bar{p}) - f(p^\star)$ (substitute p' with $p^\star$) can be somewhat bounded by the gradient $g(\cdot)$. So far, we haven't use any information given by the fact that $p^\star$ is the minimizer.

4.4 To Be Continued

Lecture 5 ends here. For more details about the theorem, you can find the rest of the proof in lecture 6 or the reading material [1].

References

[1] N. K. Vishnoi, *Algorithms for convex optimization.* Cambridge University Press, 2021.