EE C227C Convex Optimization and Approximation Lecture 2 - 01/20/2022

Lecture 2: The Complexity of Optimization

Lecturer: Jiantao Jiao *Scribe: Ishaq Aden-Ali, Eric Zhao*

In this lecture we will discuss two topics: 1) The complexity of minimizing the class of all n-dimensional ℓ_∞-Lipschitz continuous functions, and 2) introduce the notion of convexity which allows us to minimize non-linear functions via a "local-to-global" phenomena where local information allows us to find global optima.

1 Minimizing Lipschitz Functions

Suppose that we wish to minimize a function $f : \mathbb{R}^n \to \mathbb{R}$,

$$\min_{x \in B_n} f(x), \tag{1}$$

where we define B_n to be an n-dimensional box

$$B_n = \{x \in \mathbb{R}^n \mid 0 \le x^{(i)} \le 1, \ \forall i \in [n]\}, \tag{2}$$

where $x^{(i)}$ is the i-th entry of the vector x. Not that for any positive integer n, $[n] = \{1, 2, ..., n\}$. Furthermore, assume that the function f is ℓ_∞*-Lipschitz* which means that for all x, y in the box B_n,

$$|f(x) - f(y)| \le L\|x - y\|_\infty, \tag{3}$$

for some $L > 0$. Here $\|x\|_\infty = \max_{1 \le i \le n} |x^{(i)}|$ is the ℓ_∞ norm. Intuitively, we need to make such an assumption since this guarantees that the output of the function doesn't change abruptly when the input is perturbed slightly.

We now attempt to formuate a "method" to solve our optimization problem. Since we know so little about the function f, as a first attempt we can try and use the "uniform grid method" which takes as input a parameter $p \ge 1$ (measuring granularity) and does 3 steps:

1. Form a grid consisting of p^n points where each point in the grid

$$x_\alpha = \begin{pmatrix} \frac{2i_1 - 1}{2p} \\ \frac{2i_2 - 1}{2p} \\ \vdots \\ \frac{2i_n - 1}{2p} \end{pmatrix}$$

   is parameterized by some $\alpha = (i_1, i_2, \ldots, i_n) \in [p]^n$, where $1 \le i_j \le p$.

2. Among all the points x_α in the grid, find the point $\bar{x}$ with the smallest objective function value.
3. Output $(\bar{x}, f(\bar{x}))$.

The uniform grid method we have described is quite simple, but how well does it perform? The following theorem gives us an upper bound on the error of our output $(\bar{x}, f(\bar{x}))$.

Theorem 1. *Let $f^* = \min_{x \in B_n} f$ be the global optimal value. Then,*

$$f(\bar{x}) - f^* \le \frac{L}{2p}. \tag{4}$$

The quantity on the left hand side of equation (4) is referred to as the "suboptimality gap". Notice that this can never be negative since f^* is the global optimum. We now prove Theorem 1.

Proof We begin by first splitting the box B_n into sets $X_\alpha = \{x \in B_n \mid \|x - x_\alpha\|_\infty \le 1/2p\}$. Notice that the sets X_α partition B_n, i.e.

$$B_n = \bigcup_{\alpha \in [p]^n} X_\alpha. \tag{5}$$

Let x^* be a global solution obtaining the value f^*. Notice that there must be an index α^* such that $x^* \in X_{\alpha^*}$, which implies that $\|x^* - x_{\alpha^*}\|_\infty \le 1/2p$. We can thus conclude that

$$\begin{aligned}
& f(\bar{x}) - f(x^*) && \quad \textit{$\bar{x}$ minimizes the objective over all x_α} \\
& \le f(x_{\alpha^*}) - f(x^*) && \quad \textit{ℓ_∞-Lipschitz assumption} \\
& \le L\|x^* - x_{\alpha^*}\|_\infty \\
& \le \tfrac{L}{2p}
\end{aligned}$$

□

We can better interpert this result by asking what we need to set the granularity p to be in order to get ϵ error. In other words, we want to find $\bar{x} \in B_n$ such that $f(\bar{x}) - f^* \le \epsilon$. We immediately obtain the following corollary from Theorem 1.

Corollary 2. *The analytical complexity*[1] *of the problem is upper bounded by*

$$\left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor + 1 \right)^n \tag{6}$$

Proof This is an immediate consequence of the accuracy of the uniform grid method. Set $p = \lfloor L/2\epsilon \rfloor + 1$, which implies that $p \ge L/2\epsilon$. From Theorem 1 it follows that

$$\begin{aligned}
f(\bar{x}) - f^* &\le \frac{L}{2p} \qquad (7) \\
&\le \epsilon. \qquad (8)
\end{aligned}$$

□

In optimization, for high accuracy we should think of ϵ being in the range 10^{-6} to 10^{-8}. So the analytic complexity achieved by the uniform grid method seems to be very pessimistic. Perhaps surprisingly, it turns out that in some sense this is the best we can achieve. Before we prove this, we need to introduce the notion of a resisting oracle.

Definition 3. *A* ***Resisting Oracle*** *is an adversary that answers each call of the method in the worst possible way that is compatible with previous answers. Once the method is done making calls to the oracle, the oracle constructs a problem instance with the lowest possible accuracy that perfectly fits all the information collected by the method (queries and answers).*

Problem Setting	
Model	$\min_{x \in B_n} f(x)$ where f is ℓ_∞-Lipschitz
Oracle	Zero-order, when model queries x and receives $f(x)$
Goal	$f(\bar{x}) - f^* \leq \epsilon$

Figure 1: A table summarizing the problem setting.

Under the assumption that the answers to queries made by the method come from a resisting oracle, we can show a lower bound on the analytical complexity of *any* method in this setting.

Theorem 4. *For $\epsilon < \frac{L}{2}$, the analytical complexity of optimizing the class of ℓ_∞-Lipschitz functions is at least*

$$\left\lfloor \frac{L}{2\epsilon} \right\rfloor^n. \tag{9}$$

Proof Let $p = \left\lfloor \frac{L}{2\epsilon} \right\rfloor \geq 1$. Recall our partitioning of B_n into sets X_α. Assume there is a method $\mathcal{M}$ which needs $N < p^n$ calls to the oracle to guarantee that it solve all problems in the class. We will apply this method against a resisting oracle which answers every queried point $x \in B_n$ with $f(x) = 0$. Because there are p^n choices for α and $N < p^n$, there must exist an index $\hat{\alpha}$ such that no query was made to any point in $X_{\hat{\alpha}}$. We will thus construct the function

$$\hat{f}(x) = \min\{0, L\|x - x_{\hat{\alpha}}\|_\infty - \epsilon\}. \tag{10}$$

It is easy to verify that this function is ℓ_∞-Lipschitz with constant L, and that the global optimum value is $\hat{f}^* = -\epsilon$ which is attained at $x_{\hat{\alpha}}$. Furthermore, $\hat{f}$ differs from 0 only inside the set $X_{\hat{\alpha}}$, so this function agrees with the previous answers given by the oracle. Since the method $\mathcal{M}$ claims the minimum value is 0, the accuracy of any method that uses $N < p^n$ calls cannot be better than ϵ. Thus, any method achieving error less than ϵ must make $N \geq p^n = \left\lfloor \frac{L}{2\epsilon} \right\rfloor^n$ calls. □

To summarize, we have seen that the uniform grid method achieves an upper bound of $\left(\left\lfloor \frac{L}{2\epsilon} \right\rfloor + 1\right)^n$ on the analytical complexity for this optimization problem, and (assuming a resisting oracle) there is a lower bound of $\left\lfloor \frac{L}{2\epsilon} \right\rfloor^n$ (for ϵ small enough). So if one wants to stick with black box optimization, we need more assumptions about our function class to get better methods.

2 Convexity

The importance of convexity comes out when we want to work with blackbox algorithms for optimization, which implies some kind of local method. We will discuss natural assumptions for a local method and show this leads to the convex property.

First, we propose the following natural assumptions on a function family $\mathcal{F}$ that we hope to optimize over.

1. For any $f \in \mathcal{F}$, we want a first order optimality condition. In other words, $\nabla f(x^*) = 0$ if and only if x^* is a globally optimal. This is one weak way of restricting ourselves to a local-type search method.

2. Linear combinations are closed in $\mathcal{F}$. Formally, if $f_1, f_2 \in \mathcal{F}$, then for any $\alpha, \beta \geq 0$ we want to guarantee that $\alpha f_1 + \beta f_2 \in \mathcal{F}$.

[1] See lecture 1 notes for the definition of analytical complexity.

3. $\mathcal{F}$ should contain all linear functions. In other words, any function that can be written as $\ell(x) = \langle a, x\rangle + b \in \mathcal{F}$ for any a, b in our field.

We now argue that these three properties are enough to recover the property of convexity. The proof will also lead us to recover one definition of convexity.

Theorem 5. *Any function $f \in \mathcal{F}$ where $\mathcal{F}$ satisfies all properties (1), (2), and (3) is convex.*

Proof Take any function $f \in \mathcal{F}$. Assume we are working in the reals and fix a choice of $x_0 \in \mathbb{R}^n$. Then consider the hypothetical function ϕ defined as,

$$\phi(y) = f(y) - \langle \nabla f(x_0), y\rangle. \tag{11}$$

We now claim $\phi(y) \in \mathcal{F}$. $-\langle \nabla f(x_0), y\rangle$ is a linear function and therefore in $\mathcal{F}$ by property (3). Since $\phi(y)$ is then the sum of two functions in $\mathcal{F}$, by property (2), ϕ is also in $\mathcal{F}$.

Next, we use property (1) to work with a first-order-condition on ϕ. Consider its gradient of $\phi(y)$ at $y = x_0$,

$$\nabla\phi(y)|_{y=x_0} = \nabla f(x_0) - \nabla f(x_0) = 0 \tag{12}$$

By property (1), $\phi(y)$ has a global optimum at x_0. Therefore, for any other y in the domain of f

$$\phi(y) \geq \phi(x_0) = f(x_0) - \langle \nabla f(x_0), x_0\rangle. \tag{13}$$

Substituting the definition of ϕ into the left-hand-side and rearranging, we obtain,

$$f(y) \geq f(x_0) + \langle \nabla f(x_0), y - x_0\rangle. \tag{14}$$

Noting that Equation 14 exactly describes a convex function concludes our proof. □

We can formalize our derivation of a convex function with the following definitions.

Definition 6. *A set $Q \subseteq \mathbb{R}^n$ is called convex if for any $x, y \in Q$ and $\alpha \in [0, 1]$, we have that $\alpha x + (1-\alpha)y \in Q$.*

Definition 7. *A continuously differentiable[2] function is called convex on a convex set Q, with notation $f \in \mathcal{F}^1(Q)$, if for any $x, y \in Q$,*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x\rangle. \tag{15}$$

At a high-level, an important idea around convexity is that if you want a local method and need to exploit property (1), you might as well assume convexity. There are some contradictions of this, such as in deep neural networks.

We now consider some useful algorithms that exploit convexity, which we will refer to as first-order algorithms. Today, we will discuss gradient descent.

First, we make the following assumptions on our function f.

1. f is convex.
2. Gradient of f is L-Lipschitz continuous if for all x, y in the domain of f,

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|. \tag{16}$$

For now, we will assume these norms are all Euclidean; we will revisit this in a later lecture.

[2] Continuous differentiability may not be necessary, but we will assume this for now.

Instead of assuming Lipschitz continuity, you could argue for a norm bound on ∇f: $\exists B, \forall x : \|\nabla f(x)\| \leq B$. This might make more sense in than Lipschitz continuity in some contexts, and we'll analyze gradient descent using this property in later lectures. However, for now, we focus on Lipschitz continuity. The following lemma makes clear why we opt for this assumption.

Lemma 8. *(Theorem 2.1.5 in Nesterov [1]) A function satisfying convexity and gradient L-Lipschitz continuity has the following three interpretations,*

1. $f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|x - y\|^2$

   This tells us that the function f is upper-bounded by the sum of a line and a quadratic function. If L is small, the upper bound is tighter as the non-negative quadratic term shrinks. As $L \to \infty$, our Lipschitz continuity assumption tells us less about f and the bound becomes meaningless.

2. $f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2$

   This interpretation shows that $f(y)$ is also lower-bounded by a linearly supported function.

3. $\langle \nabla f(x) - \nabla f(y), x - y \rangle \geq \frac{1}{L}\|\nabla f(x) - \nabla f(y)\|^2$

   When $L \to \infty$, this tells us that the left hand side is lower-bounded by zero. This means that, once we move along a certain direction, the change in gradient is always aligned with our movement direction.

Now we describe a greedy algorithm for gradient descent.

1. Start with a datapoint $x_0 \in \mathbb{R}^n$.
2. For subsequent steps, $k \geq 0$, update our point to

$$x_{k+1} = x_k - h_k \nabla f(x_k). \tag{17}$$

   where h_k is our learning rate at timestep k.

Lemma 9. *Assume f is convex and its gradient is L-Lipschitz. The sequence x_k produced by the greedy gradient descent algorithm with learning rate h satisfies, at each timestep $k \geq 0$,*

$$f(x_{k+1}) - f(x_k) \leq -h(1 - \frac{Lh}{2})\|\nabla f(x_k)\|^2. \tag{18}$$

This guarantee explains the 'descent' in gradient descent. We will go over the proof of gradient descent in the next lecture.

References

[1] Y. Nesterov, *Introductory lectures on convex optimization: A basic course.* Springer Science & Business Media, 2003, vol. 87.