Convex Optimization

Jaden Wang

January 19, 2021

Contents

Chapter 1

Theoretical Foundation

1.1 Introduction

An optimization problem looks like

$$\min_{x \in C} f(x)$$

where $f(x)$ is the **objective function** and $C \subseteq \mathbb{R}^n$ is the **constraint set**. C might look like

$$C = \{x : g_i(x) \leq 0 \ \forall \ i = 1, \ldots, m\}.$$

Remark. We can always turn a maximization problem into a minimization problem as the following:

$$\min_x f(x) = -\max_x -f(x).$$

Therefore, WLOG, we will stick with minimization.

Example. An assistant professor earns \$100 per day, and they enjoy both ice cream and cake. The optimization problem aims to maximize the utility (*e.g.* happiness) of ice cream $f_1(x_1)$ and of cake $f_2(x_2)$. The constraints we have is that $x_1 \geq 0, x_2 \geq 0$, and $x_1 + x_2 \leq 100$.

To maximize both utility, it might be natural to define

$$F(\operatorname{vec} x) = \begin{pmatrix} f_1(x_1) \\ f_2(x_2) \end{pmatrix}, \operatorname{vec} x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

and maximize F. However, this isn't a well-defined problem, because *there is no total order on* $\mathbb{R}^n$! That is, we don't have a good way to compare whether a vector is bigger than another vector, except in the cases when the same direction of inequality can be achieved for all components of two vectors and a partial order can be established. For this kind of **multi-objective** optimization problem, we can look for Pareto-optimal points in these special cases. We can also try to convert the output into a scalar as the following:

$$\min_x f_1(x) + \lambda \cdot f_2(x_2)$$

for some $\lambda > 0$ that reflects our preference for cake vs ice cream. But this can be subjective.

Thus, For the remainder of this class, we are only going to assume $f : \mathbb{R}^n \to \mathbb{R}$.

Moreover, for $f : \mathbb{R} \to \mathbb{R}$, it's very easy to solve by using root finding algorithms or grid search. So since interesting problems occur with vector inputs, we will simply use x to represent vectors.

Notation. min asks for the minimum value, whereas arg min asks for the minimizer that yields the minimum value.

1.1.1 Lipschitz continuity

Example. Let's consider a variant of the Dirichlet function, $f : \mathbb{R} \to \mathbb{R}$

$$f(x) = \begin{cases} x & \text{if } x \in \mathbb{Q} \\ 1 & \text{if } x \in \mathbb{R} \setminus \mathbb{Q} \end{cases}$$

Then the solution to the problem

$$\min_{x \in [0,1]} f(x) = 0$$

is $x = 0$ by observation. However, the function is not smooth and a small perturbation can yield wildly different values. Thus, it is not tractable to solve this numerically.

This requires us to add a smoothness assumption:

Definition: Lipschitz continuity

$f : \mathbb{R}^n \to \mathbb{R}$ is ***L*-Lipschitz continuous** with respect to a norm $\|\cdot\|$ if for all $x, y \in \mathbb{R}^n$,

$$|f(x) - f(y)| \leq L \cdot \|x - y\|.$$

Note. Lipschitz continuity implies continuity and uniform continuity. It is a stronger statement because it tells us *how* the function is (uniformly) continuous. However, it doesn't require differentiability.

Definition: l_p norms

For $1 \leq p < \infty$,

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}.$$

For $p = \infty$,

$$\|x\|_\infty = \max_{1 \le i \le n} |x_i|.$$

Remark. $\|x\|_1$ and $\|x\|_2^2$ have separable terms as they are sums of their components. $\|x\|_2^2$ is also differentiable which makes it the nicest norm to optimize.

Example. Let $f : \mathbb{R}^n \to \mathbb{R}$ be L-Lipschitz continuous w.r.t. $\|\cdot\|_\infty$. Let $C = [0,1]^n$, *i.e.* in $\mathbb{R}^2$, C is a square. To solve the problem

$$\min_{x \in C} f(x),$$

since we have few assumption, there is no better method (in the worst case sense) than the **uniform grid method**. The idea is that we pick $p+1$ points in each dimension, *i.e.* $\{0, \frac{1}{p}, \frac{2}{p}, \ldots, 1\}$, so we would have $(p+1)^n$ points in total. Let x^* be a global optimal point, then there exists a grid point $\widetilde{x}$ s.t.

$$\|x^* - \widetilde{x}\|_\infty \le \frac{1}{2} \cdot \frac{1}{p}.$$

Thus by Lipschitz continuity,

$$\begin{aligned} |f(x^*) - f(\widetilde{x})| &\le L \cdot \|x^* - \widetilde{x}\|_\infty \\ &\le \frac{1}{2} \frac{L}{p} \end{aligned}$$

So we can find $\widetilde{x}$ by taking the discrete minimum of all $(p+1)^n$ grid points.

In (non-discrete) optimization, we usually can't exactly find the minimizer, but rather find something very close.

Definition: epsilon-optimal solution

x is a **ε-optimal solution** to $\min_{x \in C} f(x)$ if $x \in C$ and

$$f(x) - f^* \le \varepsilon$$

where $f^* = \min_{x \in C} f(x)$.

Our uniform grid method gives us an ε-optimal solution with $\varepsilon = \frac{L}{2p}$, and requires $(p+1)^n$ function evaluations. Writing p in terms of ε, we have $p = \frac{L}{2\varepsilon}$

so equivalently it requires $\left(\frac{2L}{\varepsilon}+1\right)^n$ function evaluations, which approximately is ε^{-n}.

For $\varepsilon = 10^{-6}$, $n = 100$, it requires 10^{600} function evaluations. This is really bad!

Take-aways from this example:

- curse-of-dimensionality: there can be trillions of variables in a Google Neural Network. It would be intractable using the grid method.
- we need more assumptions to allow us to use more powerful methods.

1.1.2 categorization