Introduction to Optimization Problems

Thursday, January 14, 2021 9:58 PM

An optimization problem: $\min_{x \in \mathbb{R}^n} f(x)$ s.t. $x \in C$ ("such that") or $\min_{x \in C} f(x)$

(objective function; constraint set)

Often constraints are given like $C = \{x : g_i(x) \le 0 \quad \forall i = 1, \dots, m\}$

Remark: $\min_x f(x) = -\max_x -f(x)$

so you can switch a minimization problem to a maximization problem (and vice-versa). So, wlog ("without loss of generality"), we'll usually stick w/ minimization (except in business and operations research, min. is more common than max.)

What kind of function is f?

A typical problem: an assistant professor earns \$100 every day, and they enjoy ice cream as well as cake.

The optimization problem is maximize the amount of ice cream and cake

$f_1(x_1)$ $f_2(x_2)$

(x_1: amount of \$ spent on ice cream)

such that $0 \le x_1$, $0 \le x_2$

and $x_1 + x_2 \le 100$ (budget).

Let $F(\vec{x}) = \begin{bmatrix} f_1(x_1) \\ f_2(x_2) \end{bmatrix}$, $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$. Want to maximize F

This isn't a well-defined problem! Is $\begin{bmatrix} 50 \\ 30 \end{bmatrix}$ better or worse than $\begin{bmatrix} 30 \\ 50 \end{bmatrix}$?

There's no total order on $\mathbb{R}^n$ (... well, there exists a weird one if you believe AC)

This is a multi-objective optimization problem. Very common, but mathematically unpleasant.

Our class will always assume $f : \mathbb{R}^n \to \mathbb{R}$ (not $f : \mathbb{R}^n \to \mathbb{R}^m$ — multi-obj.)

How do you deal w/ multi-objective?

1) look for Pareto-optimal points: $\begin{bmatrix} 50 \\ 30 \end{bmatrix}$ is clearly better than $\begin{bmatrix} 50 \\ 20 \end{bmatrix}$

means nothing is clearly better

2) convert to a scalar, eg.,

$\min_x \; f_1(x) + \lambda \cdot f_2(x)$ for some $\lambda > 0$ that reflects your preferences.

Choosing λ is often a big issue. For now, "not our problem"

Btw, what about $f: \mathbb{R} \to \mathbb{R}$ like you did in Calc. I ?

- We almost never consider this because it's so easy!
  Bisection, Newton, Secant methods work fine (since optimization closely related to root-finding)
  Even a grid search is reasonable in 1D.

So for us, "x" is usually a vector. I usually do not bother to write $\vec{x}$ or $\overrightarrow{x}$ or $\bar{x}$ or $\underline{x}$. Just x

More notation

min vs argmin $\quad 0 = \min_x (x-3)^2$

$3 = \operatorname{argmin}_x (x-3)^2$

argmin = values of the variable (x) that achieve the minimum

for breakout rooms or playposit

For some problems, the min value is most important
For other problems, the argmin is more important

Q1 Explain the difference between $\min_{x \in (0,1)} x^2$ and $\inf_{x \in (0,1)} x^2$

Q2 Explain intuitively what an open set is. Give a valid technical definition too.

Before we go further...
make it clear that optimization (w/o assumptions) is hopeless.

Ex 1 A variant of the Dirichlet function, $f: \mathbb{R} \to \mathbb{R}$

$$f(x) := \begin{cases} x & \text{if } x \in \mathbb{Q} \\ 1 & \text{if } x \in \mathbb{R} \setminus \mathbb{Q} \end{cases}, \quad \min_{x \in [0,1]} f(x) = 0, \text{ achieved at } x = 0$$

But without knowing the formula for x, just being able to numerically query $f(x)$, we'd never be able to find the minimum. No smoothness

So, let's add a smoothness assumption

Def: Lipschitz continuity $f: \mathbb{R}^n \to \mathbb{R}$ is L-Lipschitz continuous with respect to a norm $\|\cdot\|$ if

$$\forall x, y \in \mathbb{R}^n, \quad |f(x) - f(y)| \leq L \cdot \|x - y\|$$

Q3 What's a norm? Are there more than one norm? In $\mathbb{R}$, how many interesting norms are there?

Q4 Your friend says "all norms are equivalent"! Are they correct? And what does that mean? What are the implications for Lipschitz continuity?

Lipschitz continuity is stronger than uniform continuity but doesn't require differentiability

Def ℓ_p norms (aka ℓ^p or p-norms) Let $x \in \mathbb{R}^n$

For $1 \leq p < \infty$, $\|x\|_p := \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$ ← why do we need ?

For $p = \infty$, $\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$

These are all valid norms $1 \leq p \leq \infty$

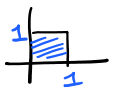
optimal Hölder conjugates $\frac{1}{p} + \frac{1}{q} = 1$

$\|x\|_1$, $\|x\|_2$ (aka Euclidean norm), $\|x\|_\infty$ most common

$\|x\|_1$ and $\|x\|_2^2$ are separable, i.e., a sum of components notation: x_i

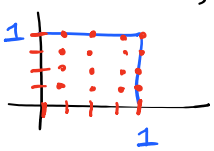
$\|x\|_2^2$ differentiable. Our favorite function!

Ex. 2 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 $\mathbb{R}^2$, $C =$ 

$\min_{x \in C} f(x)$. There's nothing we can do to solve that's better (in a worst-case sense) than the uniform grid method.

Pick $p+1$ points in each dimension, $\{0, \frac{1}{p}, \frac{2}{p}, \ldots, \frac{p}{p} = 1\}$

Ex $\mathbb{R}^2$, $p = 4$ 

make uniform grid w/ $(p+1)^n$ points.

Let x^* be a global optimal point. Then $\exists$ a grid point $\tilde{x}$

such that $\|x^* - \tilde{x}\|_\infty \le \frac{1}{2}\frac{1}{p}$.

Thus by Lipschitz continuity, $|f(x^*) - f(\tilde{x})| \le L \cdot \|x^* - \tilde{x}\|_\infty \le \frac{1}{2}\frac{L}{p}$

and we can find $\tilde{x}$ (or something better) by taking the discrete minimum of all $(p+1)^n$ grid points

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

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

and $f(x) - f^* \le \varepsilon$ $\quad (f^* := \min_{x \in C} f(x))$

So... our uniform grid method gives us an ε-optimal solution w/ $\varepsilon = \frac{L}{2p}$, and required $(p+1)^n$ function evaluations.

i.e., for a fixed ε, set $p = \frac{L}{2\varepsilon}$, so $\left(\frac{2L}{\varepsilon} + 1\right)^n$ function evaluations

i.e. $\approx \varepsilon^{-n}$ function evaluations.

For $\varepsilon = 10^{-6}$, $n = 100$, this is 10^{600} function evaluations.

BAD NEWS

take-aways

1. curse-of-dimensionality. Optimization in high dimensions is challenging.

   Google "Switch Transformer" language model, deep NN w/ $1.6 \cdot 10^{12}$ variables.

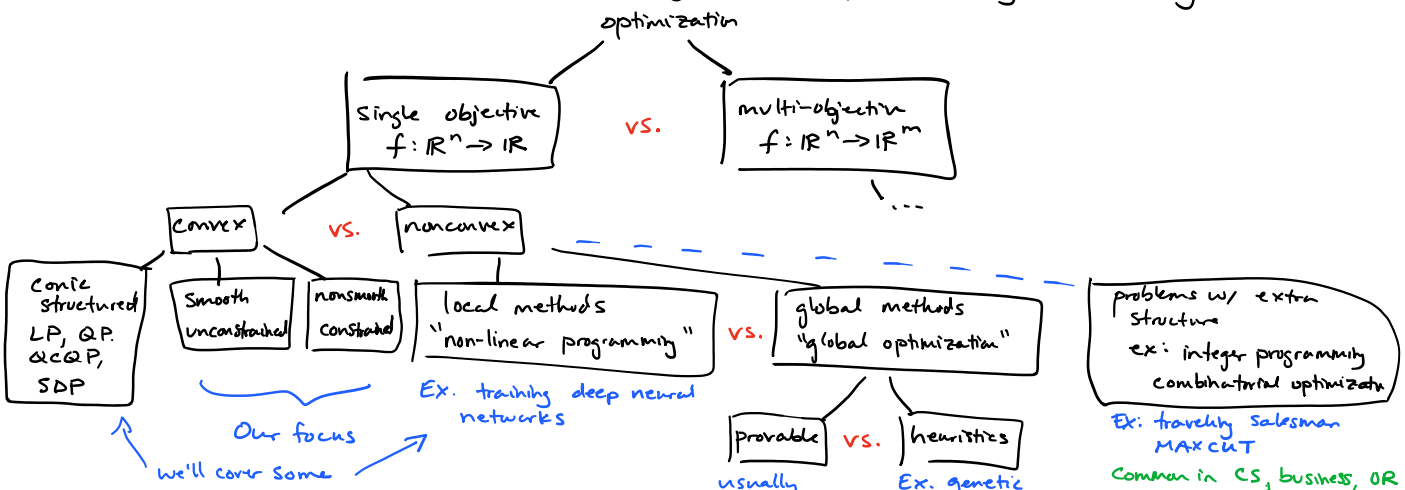
2. we need more assumptions.

In this class...

we need to first talk about assumptions for f (and C) before we can get to algorithms to solve these problems.

Types of optimization problems

This classification isn't the only way to do it, and may reflect my own biases



Simulated annealing is "provable" but for unrealistic parameter choices (too slow), and when it's used, it's a "heuristic"