Chapter 0: Introduction and Preliminaries

Ernest K. Ryu

MATH 164: Optimization

University of California, Los Angeles
Department of Mathematics

Goal of this class

This class is about (mathematical) optimization.[1]

- Many engineering problems: We need to make a choice, and we want to make the "best" choice.
- Many scientific problems are: Nature will equilibrate at the "minimum" energy configuration, and we wish to find this configuration.
- Many data science problems: We want to find the model configuration (parameter) that "best" explains the data.

Mathematical optimization is the underlying math problem that abstracts away the engineering/scientific context. (Calculus is used to model physical systems, but calculating derivatives and integrals is independent of the physical context the calculus problems originate from.)

[1] Mathematical optimization contrasts with, say, compiler optimization or code optimization

Unconstrained optimization

An unconstrained optimization problem has the form

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x),$$

where $f \colon \mathbb{R}^n \to \mathbb{R}$ has appropriate assumptions.

We refer to x as the *optimization variable* or *decision variable* and f as the *objective function* or *loss function*.

In this class, we assume x is a continuous variable and that f is continuous and (usually) differentiable. Problems with such structure are referred to as *continuous optimization* problems.

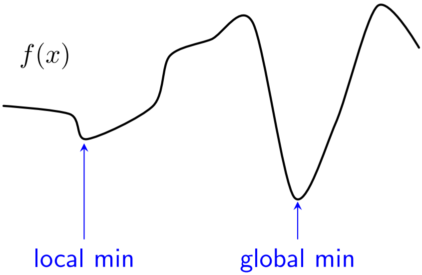
Problems with Boolean- or integer-values x are referred to as *combinatorial optimization* problems. (Not our focus.)

Local vs. global minima

$x_\star$ is a *local minimum* if $f(x) \geq f(x_\star)$ within a small neighborhood.[2]

$x_\star$ is a *global minimum* if $f(x) \geq f(x_\star)$ for all $x \in \mathbb{R}^n$

In the worst case, finding the global minimum of an optimization problem is difficult. (The class of non-convex optimization problems is NP-hard.)



[2] if $\exists\, r > 0$ s.t. $\forall\, x$ s.t. $\|x - x_\star\| \leq r \Rightarrow f(x) \geq f(x_\star)$

Minimization vs. maximization

Why consider minimization problems? Why not maximize?

Minimization and maximization problems are equivalent since

$$\underset{x \in \mathbb{R}^n}{\text{maximize}} \quad f(x) \qquad \Leftrightarrow \qquad \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad -f(x).$$

When maximizing, we refer to f as the *objective function*, *reward function*, and *merit function*.

Min vs. max: Choose what is more natural given the problem context.

The baseline convention is to minimize because of convexity.
More on this later.

Definition of solutions

For minimization problems, we define *solutions* to be global minimizers.

- A solution may or may not exist.
- A solution may or may not be unique.

Some refer to a local minimizer as a "local solution." We will not use this terminology.

Some refer to any (feasible) point as a " solution." (In a business context, if a company is selling you a "solution," this is an actionable plan that is hopefully decent, but there is no promise of the optimality of this plan.) We will not use this terminology.

In maximization problems, we define solutions to be global maximizers.

Solving unconstrained optimization with calculus

In calculus, you have actually seen some optimization.

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x),$$

and assume $f\colon \mathbb{R}^n \to \mathbb{R}$ is differentiable. Then,

$$\operatorname{argmin} f \subseteq \{x \in \mathbb{R}^n \mid \nabla f(x) = 0\}.$$

($\min f$ is the minimum *value* of f while $\operatorname{argmin} f$ is the set of *input* x's minimizing f.)

In other words, the minimizers must have zero-gradient ($\nabla f(x) = 0$ is a necessary condition). However, this is not a sufficient condition, and you did things like the second derivative test.

Solving unconstrained optimization with calculus

Consider

$$\underset{x,y\in\mathbb{R}}{\text{maximize}} \quad f(x,y) = 2xy + 2x - x^2 - 2y^2.$$

Then,

$$\nabla f(x,y) = \begin{bmatrix} 2y + 2 - 2x \\ 2x - 4y \end{bmatrix}.$$

Solving for $\nabla f(x,y) = 0$ yields $(x,y) = (2,1)$. Next, carry out the second derivative test.

$$\nabla^2 f(2,1) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2}(2,1) & \frac{\partial^2 f}{\partial x \partial y}(2,1) \\ \frac{\partial^2 f}{\partial x \partial y}(2,1) & \frac{\partial^2 f}{\partial y^2}(2,1) \end{bmatrix} = \begin{bmatrix} -2 & +2 \\ +2 & -4 \end{bmatrix}$$

Then $\det\left(\nabla^2 f(x)\right) > 0$ and $f_{xx}(2,1) < 0$, so $(2,1)$ is a local maximum of f. (Alternatively, we can note that both eigenvalues of $\nabla^2 f(x,y)$ are negative.) With a little bit more work, we can show that $(2,1)$ is the global maximum.

Very nice. We can do this all the time?

Can't solve unconstrained optimization with calculus

Consider minimizing the Mishra's Bird function

$$\underset{x,y\in\mathbb{R}}{\text{minimize}} \quad f(x,y) = \sin(x)e^{(1-\cos(y))^2} + \cos(y)e^{(1-\sin(x))^2} + (x-y)^2.$$

(This is a commonly used non-convex test function to evaluate the performance of optimization algorithms.) Then,

$$\nabla f(x,y) = \begin{bmatrix} \cos(x)e^{(1-\cos(y))^2} - 2\cos(y)\cos(x)(1-\sin(x))e^{(1-\sin(x))^2} + 2(x-y) \\ 2\sin(x)\sin(y)(1-\cos(y))e^{(1-\cos(y))^2} - \sin(y)e^{(1-\sin(x))^2} - 2(x-y) \end{bmatrix}.$$

Solving for $\nabla f(x,y) = 0$ analytically is impossible.

Recommended solution 1: Plot the 2D function and eyeball the solution.

Recommended solution 2: Take the eyeballed solution and run GD to refine it to local optimality.

Can't solve unconstrained optimization with calculus

Consider the ℓ_2-regularized logistic regression problem

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad \sum_{i=1}^{N} \log\left(1 + \exp(v_i^\intercal x)\right) + \frac{\lambda}{2}\|x\|^2,$$

for some $\lambda > 0$ and $v_1, \ldots, v_N \in \mathbb{R}$. (These arise in statistics and machine learning.)

Then,

$$\nabla f(x) = \lambda x + \sum_{i=1}^{N} \frac{1}{1 + \exp(-v_i^\intercal x)} v_i$$

Solving for $\nabla f(x, y) = 0$ analytically is impossible. When $d > 2$, plotting and eyeballing the solution is impossible.

We must use a numerical algorithm.

It turns out, there is a unique point satisfying $\nabla f(x) = 0$, and it can be computed reliably with GD.