Lecture 10

Monday, February 10, 2025 12:50 AM

Earlier, you learned how to solve linear programming problems by hand using the *simplex method*. The goal for the next few weeks is to learn numerical methods for nonlinear programming problems (more commonly called *nonlinear optimization problems*). These methods are, in general, tedious to do by hand, even with a pocket calculator.

Nonlinear optimization problem is a problem of finding min/max value of a nonlinear function f. Most functions from real-life applications have more than one variable (multivariable functions). We will start with optimization problems for functions of a single variable.

Let $f = f(x)$ be a single-variable function. How do you find min/max value of this function? Note that min/max don't always exist. For example, $f(x) = x$ doesn't have min/max on the entire real line, but it has both min and max on the interval $[-2,3]$. In Calculus I, you learned the **Extreme Value Theorem**:

If f is a continuous on a closed interval $[a, b]$ then both $\min_{[a,b]} f$ and $\max_{[a,b]} f$ exist.

You also learned how to find min/max of a function using **Fermat's theorem**, which says that if f is differentiable then min/max, if occurs at $c \in (a, b)$, must be a critical number of f, i.e. $f'(c) = 0$.

However, the equation $f'(x) = 0$ is not always easy to solve. You will learn how to solve such an equation numerically later. The most natural way to find the min/max of a function f is perhaps by graphing f and locate the position of min/max on the graph. This visual method sounds attractive, but it requires you to be able to evaluate $f(x)$ at a lot of values of x. Imagine that you are somewhere on a mountain (on a 2D plane) and you want to reach the bottom. The mountain can be thought as the graph of a function f. You want to reach its minimum. You don't know what this function is, so you can't graph it. However, with proper measuring instrument, you can tell the slope of the mountain at your position. This information turns out to be enough to solve the problem.

Let's consider a simple and intuitive algorithm called **gradient descent** to find minimum. The counterpart to find maximum is called *gradient ascent*. However, since maximum can be converted into minimum easily by multiplying the function by -1, we will only consider the problem of finding minimum.

Starting at an *initial guess* x_0 for the location of the minimum. We then update this guess by x_1, then update it again by x_2, and so on. Intuitively, $x_{n+1} - x_n$ should be propotional to $-f'(x_n)$.

$$x_{n+1} = x_n - \alpha f'(x_n)$$

where α is called the *learning rate*.

Example: Approximate the minimum of $f(x) = x^2$ with $n = 5$, initial guess $x_0 = 1$, and learning rate $\alpha = 0.4, 0.6, 1, 2$.

Example: Approximate the maximum value of $y = y(x)$ where function y satisfies the differential equation $y' = y^2 - x$, $y(0) = 0.7$.