ROB 101 - Fall 2021

Optimization: Second-Order Unconstrained

November 17, 2021

UNIVERSITY OF MICHIGAN

Learning Objectives

- Mathematics is used to describe physical phenomena, pose engineering problems, and solve engineering problems.
- We show how linear algebra and computation allow you to use a notion of "optimality" as a criterion for selecting among a set of solutions to an engineering problem.

Outcomes

- Arg min should be thought of as another function in your toolbox,
$$x^* = \underset{x \in \mathbb{R}^m}{\arg\min}\, f(x).$$

- We will add to our knowledge of derivatives, specifically, second derivative.

- Second-order optimization methods are based on root finding.

Optimization as a Root Finding Problem: the Hessian

- Objective function $f : \mathbb{R}^m \to \mathbb{R}$.

- First-order *necessary* condition for x to be a local extremum of f is $\nabla f(x) = 0$.

- Therefore, our locally minimizing solutions are roots of the derivative (gradient) of the objective function.

- We can think of the gradient as a map $\nabla f : \mathbb{R}^m \to \mathbb{R}^m$ (column vector).

- We consider $\nabla f : \mathbb{R}^m \to \mathbb{R}^m$ and seek a root $\nabla f(x_0) = 0$.
- Note that the domain and range are both $\mathbb{R}^m$ and thus this is the nonlinear equivalent of solving a square linear equation $Ax - b = 0$.
- We recall that $\det(A) \neq 0$ was our magic condition for the existence and uniqueness of solutions to $Ax - b = 0$.

- Let x_k be our current approximation of a root of the function ∇f.

- We write the linear approximation of ∇f about x_k as

$$\nabla f(x) \approx \nabla f(x_k) + H_k(x - x_k), \; H_k := \frac{\partial}{\partial x}\nabla f(x_k) = \nabla^2 f(x_k).$$


WE NEED TO GO
DEEPER

- $H_k := \frac{\partial}{\partial x} \nabla f(x_k) = \nabla^2 f(x_k)$, called *Hessian*, is the Jacobian of ∇f.

- Or when we use the row vector convention, the Hessian of a function $f : \mathbb{R}^m \to \mathbb{R}$ is the Jacobian of the transpose of the gradient of the function

$$\nabla^2 f(x) := \frac{\partial}{\partial x} \nabla f(x)^\mathsf{T}.$$

- The Hessian is a symmetric square matrix of second-order partial derivatives of a scalar-valued function $f : \mathbb{R}^m \to \mathbb{R}$.

$$\nabla f(x)^{\mathsf{T}} := \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix} \in \mathbb{R}^m$$

$$H(x) := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \frac{\partial^2 f}{\partial x_2\,\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 f}{\partial x_n\,\partial x_1} & \frac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \in \mathbb{R}^{m \times m}$$

- We want to chose x_{k+1} so that $\nabla f(x_{k+1}) = 0$.
- $\nabla f(x_{k+1}) = \nabla f(x_k) + H_k(x_{k+1} - x_k) = 0$.
- Define $\Delta x_k := x_{k+1} - x_k$. Then $x_{k+1} = x_k + \Delta x_k$.
- $\nabla f(x_{k+1}) = 0 \implies H_k \Delta x_k = -\nabla f(x_k)$.

Notation

- $A \succeq B \quad \Leftrightarrow \quad A - B$ is positive semidefinite
- $A \succ B \quad \Leftrightarrow \quad A - B$ is positive definite

Remark

An $n \times n$ symmetric matrix is positive definite if and only if all of its eigenvalues are positive.

Remark

An $n \times n$ symmetric matrix is positive semi-definite if and only if all of its eigenvalues are non-negative.

- First-order *necessary* condition

$$\nabla f(x) = 0$$

- Second-order *necessary* condition

$$\nabla f(x) = 0 \quad \text{and} \quad H(x) \succeq 0$$

- Second-order *sufficient* condition

$$\nabla f(x) = 0 \quad \text{and} \quad H(x) \succ 0$$

Objective function: $f(x) = \frac{1}{2}\|Ax - b\|^2$

- Gradient: $\nabla f(x) = A^\mathsf{T} Ax - A^\mathsf{T} b$,
- $\nabla f(x^\star) = 0 \Rightarrow A^\mathsf{T} Ax^\star = A^\mathsf{T} b$ (Normal Equations),
- Hessian: $H(x) = A^\mathsf{T} A \succ 0$.

Assumption

- $A \in \mathbb{R}^{n \times m}$ *and* $b \in \mathbb{R}^n$
- $n \geq m \Leftrightarrow A$ *is a tall matrix*
- $\operatorname{rank}(A) = m$ *(i.e., columns of A are linearly independent)*

Damped Newton-Raphson Algorithm

We use a step size $0 < \alpha < 1$ to control each update size (damped Newton).

1. Start with an initial guess x_0 ($k = 0$).
2. If $\|\nabla f(x_k)\| = 0$, then the algorithm is converged.
3. Solve the linear system $H_k \Delta x_k = -\nabla f(x_k)$.
4. Update the decision variable via $x_{k+1} = x_k + \alpha \Delta x_k$.
5. Repeat (go back to 2) until convergence.

Let's switch to the Julia notebook.

Next Time

- Affine Spaces & Hyperplanes
- Read Chapter 13 of ROB 101 Book