Basic information and policies

- Name: Yijia Zhao
- Office Hour: Friday 10 am - 12 pm in MS 8349
- Email: yijiazhao@ucla.edu (with lecture section number in the subject)
- Homework policy:
  - Must be turned in through the respective submission portals on Bruinlearn. No grace period for late submissions.
  - A 72-hour extension if submitted with a documentation showing the date and reason for inability to submit on time. Do not email the documentation.
  - Extensions beyond 72 hours will only be granted in person by the instructor.
  - Regrade requests should be sent to the TA and the instructor. Submissions will be graded in their entirety and any grade adjustments will be considered final.
  - It is the student's responsibility to check grades on BruinLearn in a timely manner so that any grade issues are resolved before the quarter is over.

Optimization Problem

- The objective of this course is to study algorithms of how to solve

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

  where $f(\cdot)$ is the objective function we aim to minimize, x is the argument of the function, D is the domain of interest of the argument x.
- We primarily focus on unconstrained optimization problem, assuming constrained problems have been transformed to unconstrained ones, i.e., $D = \mathbb{R}^n$.
- Example — Linear regression:

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|y - X\beta\|_2^2$$

Unconstrained Optimization Problem

To solve the problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

we require:

1. Solve the gradient equation $\nabla f(x) = 0$.
2. For the solution $x^\star$ of the gradient equation, evaluate the Hessian matrix of $f(x)$ at $x^\star$:
   - If $H(x^\star) \succ 0$, then $x^\star$ is a (local) minimum.
   - If $H(x^\star) \prec 0$, then $x^\star$ is a (local) maximum.
   - Otherwise, $x^\star$ is a saddle point.

Gradient Calculation

- Rules
  1. Gradient $\nabla g(x) = \begin{bmatrix} \frac{\partial g(x)}{\partial x_1} \\ \frac{\partial g(x)}{\partial x_2} \\ \cdots \\ \frac{\partial g(x)}{\partial x_n} \end{bmatrix}$.
  2. If $g(x) = w^\top x$ where $w, x \in \mathbb{R}^n$, then $\nabla g(x) = w$.
  3. If $g(x) = Ax$ where $A \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$, then $\nabla g(x) = A^\top$.
  4. If $g(x) = x^\top Ax$ where $A \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$, then $\nabla g(x) = (A + A^\top)x$.
  5. Chain rule: $\nabla_x f(g(x)) = \frac{df}{dg} \times \nabla_x g$.
- Examples
  1. $f(x_1, x_2) = ax_1^2 + bx_2^2 + cx_1x_2 + dx_1 + ex_2 + f$
  2. $f(x_1, x_2) = \frac{1}{1+e^{-(ax_1+bx_2+c)}}$
  3. $f(x_1, x_2) = \ln(1 + e^{-(ax_1+bx_2+c)})$

Iterative Descent Algorithm

- Iterative descent algorithm:
  1. Initial guess: x_0.
  2. Iteratively generate $x_1, x_2, \cdots, x_k, x_{k+1}, \cdots$ according to the rule $x_{k+1} = x_k + \eta_k d_k$ such that $f(x_{k+1}) \leq f(x_k)$.
- Factors in design:
  1. Descent direction d_k such that $[\nabla f(x_k)]^\top d_k < 0$.
  2. Step size η_k which makes the algorithm converge in fewer iterations. (Selection of η_k matters!)
  3. Stopping criterion
- Gradient descent algorithm: $d_k = -\nabla f(x_k)$, i.e., $x_{k+1} = x_k - \eta_k \nabla f(x_k)$.