15.093: Optimization **Fall 2023**

Assignment 4: Non-linear optimization

Assigned: October 19; Due: November 01.

Convergence rate of gradient descent [40 pts]

Consider an M-smooth function $f : \mathbb{R}^n \mapsto \mathbb{R}$. We consider the following optimization problem:

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

We apply the gradient descent method with constant step size $\alpha \leq 1/M$.

(a) Show that the following inequality holds for each iteration k:

$$\|\nabla f(\boldsymbol{x}^k)\|^2 \leq \frac{2}{\alpha}(f(\boldsymbol{x}^k) - f(\boldsymbol{x}^{k+1}))$$

(b) Show that the following inequality holds for each iteration k:

$$\min_{\ell=0,\cdots,k-1} \|\nabla f(\boldsymbol{x}^\ell)\| \leq \sqrt{\frac{2}{\alpha k}(f(\boldsymbol{x}^0) - z^*)}$$

(c) Bound the number of iterations required to reach a solution $\overline{\boldsymbol{x}}$ such that $\|\nabla f(\overline{\boldsymbol{x}})\| \leq \varepsilon$. Comment on the rate of convergence, compared to the results seen in class.

Newton's method [20 pts]

Define the following cubic functions

$$\begin{aligned} f(x) &= x^3 - 12x^2 + 10x + 3 \\ g(x) &= x^3 + 2x \end{aligned}$$

a. Implement Newton's method to minimize f. Using a starting point of 10, plot the solution over 30 iterations. Repeat with a starting point of 2.5. Comment briefly on the performance of the algorithm.

b. Implement Newton's method to minimize g. Using a starting point of 5, plot the solution over 30 iterations. Comment briefly on the performance of the algorithm.

Gradient descent vs. stochastic gradient descent [40 pts]

You have access to two datasets, each with $n = 1,200$ observations. Each dataset comprises two covariates stored in 1,200×2 matrices $\boldsymbol{X}^A$ and $\boldsymbol{X}^B$, and 1,200×1 label vectors $\boldsymbol{y}^A$ and $\boldsymbol{y}^B$. The goal is to build a *single* linear regression model for both datasets, with the following loss function:

$$\min_{\beta_1,\beta_2} \quad \frac{1}{2n}\sum_{i=1}^{n} \min\left\{ \left(y_i^A - \beta_1 X_{i1}^A - \beta_2 X_{i2}^A\right)^2, \left(y_i^B - \beta_1 X_{i1}^B - \beta_2 X_{i2}^B\right)^2 \right\}$$

You have access to the following data files:

– dataA.csv: A matrix of size 1,200×3 that stores the first covariate $X^A_{11}, \cdots, X^A_{n1}$ (first column), the second covariate $X^A_{12}, \cdots, X^A_{n2}$ (second column), and the label $y^A_1, \cdots, y^A_n$ (third column).
– dataB.csv: A matrix of size 1,200×3 that stores the first covariate $X^B_{11}, \cdots, X^B_{n1}$ (first column), the second covariate $X^B_{12}, \cdots, X^B_{n2}$ (second column), and the label $y^B_1, \cdots, y^B_n$ (third column).

We will implement gradient descent and stochastic gradient descent for this problem. Throughout, use a starting point of $\boldsymbol{\beta}^0 = (45, -15)$, 200 iterations, and a constant learning rate.

a. Provide a contour plot of the loss function, for $\beta_1 \in [-25, 75]$ and $\beta_2 \in [-40, 40]$. What is the shape of the function? What are the implications for unconstrained optimization algorithms?

b. Implement the gradient descent method with a learning rate of 0.01. Add the algorithm's trajectory on the contour plot from Question a. Repeat with a learning rate of 0.1, and provide a new trajectory plot. Comment briefly on the performance of the algorithm.

c. Implement the stochastic gradient descent method with a learning rate of 0.01. Run the algorithm three times, with three random seeds. Report a contour plot of the cost function showing the three trajectories of your algorithm. Comment briefly.