CSCI 1051 Homework 4

January 29, 2024

Submission Instructions

Please upload your solutions by **5pm Friday February, 2024.**

- You are encouraged to discuss ideas and work with your classmates. However, you **must acknowledge** your collaborators at the top of each solution on which you collaborated with others and you **must write** your solutions and code independently.
- Your solutions to theory questions must be typeset in LaTeX or markdown. I strongly recommend uploading the source LaTeX (found here) to Overleaf for editing.
- I recommend that you write your solutions to coding questions in a Jupyter notebook using Google Colab.
- You should submit your solutions as a **single PDF** via the assignment on Gradescope. You can enroll in the class using the code GPXX7N.
- Once you uploaded your solution, **mark where you answered each part of each question**.

Problem 1

Consider the linear regression problem with $n \geq d$. Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be a feature matrix and $\mathbf{b} \in \mathbb{R}^n$ be a target vector. The regression problem is to find a minimizing vector

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

You previously showed that the optimal solution is $\mathbf{x}^* = \left(\mathbf{A}^\top \mathbf{A}\right)^{-1} \mathbf{A}^\top \mathbf{b}$. In this problem, you will compare computing the optimal solution exactly to computing it approximately using the fast Johnson-Lindenstrauss transform. We will use the MNIST dataset to build $\mathbf{A}$ and $\mathbf{b}$. The MNIST dataset consists of 28×28 pixel handrawn digits of numbers with the corresponding label.

Part 1 (1 point)

Using the code I provide in `regression.py`, compute the exact solution $\mathbf{x}^*$ and the mean squared error

$$\frac{1}{n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

If your code is anything like mine, it will be slow and return a a terrible solution due to *round off error.*

Part 2 (2 points)

Now implement the fast JL transform as described in class. In particular, compute $\mathbf{\Pi}\mathbf{A} = \mathbf{S}\mathbf{H}\mathbf{D}\mathbf{A}$ one column of $\mathbf{A}$ at a time. Recall that $\mathbf{S}$ is a sampling matrix, $\mathbf{H}$ is a Hadamard, and $\mathbf{D}$ is a diagonal matrix with a random sign.

When you are done, compute the mean squared error of your solution and comment on how it compares to the "exact" solution.

Hint: Computing $\mathbf{H}$ is too expensive so write a function to compute $\mathbf{H}\mathbf{D}\mathbf{x}$ using recursion. You can speed up the recursion by checking if there are any non-zeros in the vector.