Math 126 Problem 21: Sketched Regression and the Fast JL Transform

Spring 2027

Submission Instructions

Please upload your work to Gradescope by **11:59pm on Monday, May 3, 2027 (Pacific time).**

- 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 independently.
- Your solutions to theory questions must be written legibly, or typeset in LaTeX or markdown.
- I recommend that you write your solutions in a Python notebook using Google Colab.
- You should submit your solutions as a *single PDF* via the assignment on Gradescope.

Grading: The point of the problem set is for *you* to learn. To this end, I hope to disincentivize the use of LLMs by *not* grading your work for correctness. Instead, you will grade your own work by comparing it to my solutions. This self-grade is due the Friday *after* the problem set is due, also on Gradescope.
Original notebook: Google Colab.

Sketched Regression and the Fast JL Transform

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 A: Standard Solution

Load a regression problem of your choice and 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 B: Fast JL Transform

Now implement the fast JL transform as described in class. In particular, compute $\mathbf{\Pi A} = \mathbf{SHDA}$ 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 $\mathbf{HDx}$ using recursion.