Math 126 Problem 12: Locality Sensitive Hashing in the Wild

Spring 2027

Submission Instructions

Please upload your work to Gradescope by **11:59pm on Monday, March 22, 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.

Locality Sensitive Hashing in the Wild

To support its largely visual platform, Pinterest runs a massive image de-duplication operation built on Locality Sensitive Hashing for Cosine Similarity. You can read about the actual system here. All information and numbers below are otherwise purely hypothetical.

Pinterest has a database of $N = \mathbf{1\ billion}$ images. Each image in the database is pre-processed and represented as a vector $\mathbf{x} \in \mathbb{R}^d$. When a new image is pinned, it is also processed to form a vector $\mathbf{y} \in \mathbb{R}^d$ and checked for any existing duplicates or near-duplicates to $\mathbf{y}$ in the database.

Part A

Write the probability that a vector $\mathbf{y}$ is checked as a function of the cosine similarity $\cos(\theta(\mathbf{q}, \mathbf{y}))$. As the number of tables t increases, how does the probability change? As the number of bands r increases, how does the probability change?

Part B

Pinterest has two simultaneous goals:

1. If an image $\mathbf{y}$ has $\cos(\theta(\mathbf{x}, \mathbf{y})) = .98$, we want to check the image with probability $\geq 99\%$.
2. We want the total number of checks we make to be less than a constant C to be specified later.

Given these goals, your job is to design a multi-table LSH scheme to find candidate near-duplicates, which can then be checked directly against $\mathbf{x}$. To support this task, Pinterest has collected data on the empirical distribution of $\cos(\theta(\mathbf{x}, \mathbf{y}))$ for a typical new image $\mathbf{x}$. It roughly follows a bell-curve.

Based on the data below, describe how to set parameters for your LSH scheme to minimize the space (i.e., number of tables) used when achieving the above goals for $C = 1,000,000$ and $C = 200,000$. Justify your answers, and any assumptions you make.

If you code anything up to help calculate your answer, please attach the code. As in class, you can assume that each hash table has $m = O(N)$ slots and this is large enough to ignore lower order terms depending on $1/m$.

