Homework 7
APPM 5650 Fall 2021
Randomized Algorithms

Due date: Monday, Oct 11 2021
Theme: Regression

Instructor: Prof. Becker
Revision date: 10/4/2021

Instructions Collaboration with your fellow students is allowed and in fact recommended, although direct copying is not allowed. Please write down the names of the students that you worked with. The internet is allowed for basic tasks only, not for directly looking for solutions.

An arbitrary subset of these questions will be graded.

Problem 1: [READING] Read section 1.1 "Puzzle 1: Finding Missing Numbers" from "Data Streams: Algorithms and Applications" by S. Muthukrishnan (Foundations and Trends® in Theoretical Computer Science, 2005).

Read more if you are interested.

For fun, here's a variant of the "hat puzzle," slightly related to the types of ideas Muthukrishnan things about. Suppose there are 100 people, all in a line and facing the front of the line (so the person at the back can see everyone in front of them, the next person can see everyone except the person behind them, and the person in front can't see anyone). Each person has a red or blue hat on, and they don't know what color their hat is. The game is as follows: starting with the person at the back of the line, each person in turn says either "red" or "blue." The goal is to devise a strategy so that as many people as possible say the color that corresponds to their own hat (i.e., beat the 50% success rate of random guessing). What's the best strategy?

Deliverable: none required.

Problem 2: [CODING] Faster least squares. Load the MNIST data we used previously, which gives a matrix X which is 784×3000. Make 5 copies of this matrix and merge all the copies horizontally, then transpose it to get a new matrix (call it A to follow our usual convention), which is now $M \times N$ with $M = 15000$ and $N = 784$. Construct a random vector x and compute $b = Ax + z$ where z is a small amount of noise. Compute the least squares estimator $x_{\text{LS}} = \text{argmin}_x \|Ax - b\|_2$ using a standard software package such as Matlab's' `mldivide` aka backslash, or Python's `numpy.linalg.lstsq`. See code below for sample Matlab code to set this up:

```
load MNIST_subsampled % loads X which is 784 x 3000
A           = repmat( X, 1, 5 )';
[M,N]       = size(A);
xSignal     = randn(N,1);
b           = A*xSignal + 1*randn(M,1);
xLS         = A\b; % least squares estimator
```

For at least three types of sketches S (Gaussian, count sketch, some type of fast Johnson-Lindenstrauss, Haar, sub-sampling, very sparse), solve the following sketched least-squares problem (again, using a standard software package):

$$\hat{x} = \underset{x}{\text{argmin}} \|SAx - Sb\|_2$$

How accurate is $\hat{x}$ compared to x_{LS}? Is the sketched approach actually faster?

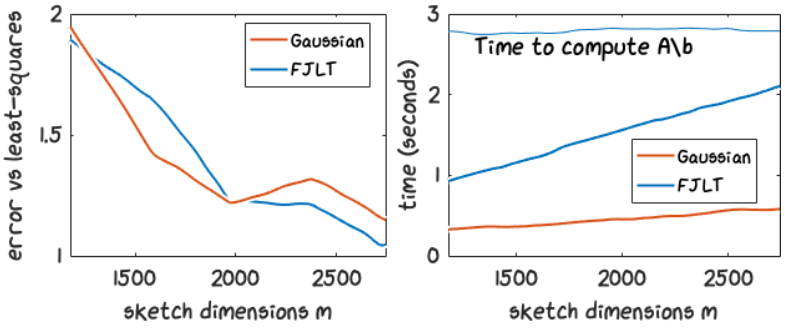

Figure 1: Examples for the type of output we want on problem 2.

Deliverable: investigate the accuracy and speed of sketched regression as a function of the number of rows m in S. Make plots of error and time vs m, as in Fig. 1. When reporting "error," there are many different things you can report. Choose one or two sensible error metrics.

Bonus (not required): do your accuracy results change significantly if we change to a different matrix A? You might hypothesize that it would not affect the Gaussian sketch, but may affect other sketches. Is this true?

Bonus (not required): try a leverage score sampling approach. You'll first need to look into the approximate SVD.

Hint: You can re-use your sketching code from previous assignments. If there was an error with that code, you're welcome to use a classmates' sketching code or one on the class webpage. For example, Matlab users, you can use `sketch.m` from the Github repository github.com/stephenbeckr/randomized-algorithm-class/blob/master/Code/. Hopefully later we'll add a Python (or even Julia) version; if you want to contribute to this, let the instructor know.