Homework 4
APPM 4720/5720 Spring 2019
Randomized Algorithms

Due date: Friday, Feb 8 2019
Theme: Sketches

Instructor: Stephen Becker

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: [MATH] Let A be a $m \times n$ matrix, and $\sigma_{\max} = \|A\|$ the maximum singular value of A, and $\sigma_{\min}$ the minimum singular value of A (which may be zero). Prove that for all vectors $x, y \in \mathbb{R}^n$ that

$$\sigma_{\min}\|x - y\|_2 \leq \|Ax - Ay\|_2 \leq \sigma_{\max}\|x - y\|_2$$

(and note that there exist x, y such that these inequalities are tight, so they cannot be improved).

Problem 2: [PROGRAMMING] Using the 10×10^9 matrix U that is stored in the file `/rc_scratch/stbe1590/data_U.mat` on the research computing (rc) filesystem, compute its spectral norm $\|U\|$. The file was saved in the "version 7.3" Matlab file format, which uses HDF5. The file is about 72 GB, so I suggest *not* transferring it to your computer.

Deliverable: What is $\|U\|$? And how much time did your code spend reading the data from disk, and how much time did it take to actually do the computations?

Hints: on the rc login node, you must request a compute node. You should have access to the 2 special nodes on the Blanca cluster that the applied math department owns. To access these nodes, run `module load slurm/blanca` and then request an interactive session on one core of one of these nodes with the command `sinteractive --account=blanca-appm`. Now you have command line access on a compute node. If you want to run Matlab, it's simplest to do it without the GUI. First run `module load matlab` and then run `matlab -nodisplay` to launch a terminal version of Matlab. You can edit script files, and then call them from the Matlab command lines. If you use Python, you can use Jupyter (TBD).

In Matlab, you can load just parts of the file by using the `matfile` command, rather than the `load` command. Please read the documentation of `matfile`.

In Python, install the h5py package https://www.h5py.org/ to be able to read the .mat file. The `scipy.io.loadmat` package will not work.

There is a smaller file `/rc_scratch/stbe1590/data_U_small.mat` which holds a 10×10^5 matrix; you might find this helpful for checking whether your code is correct.

Problem 3: [PROGRAMMING] **Are random projections that good?** We often use random projections precisely because they are *agnostic* to the data. What if we custom tailer a projection to the data?

Load the dataset `/rc_scratch/stbe1590/MNIST_subsampled.mat`, which is a sub-sampled version of the MNIST dataset. The file contains two variables, `X` which contains a 784×3000 matrix of the data, and `labels` which contains 3000 labels. Each label is a digit 0–9, and each

column of the data matrix is a 784 length vector that represents a hand-drawn digit. You can reshape each column to be a 28×28 matrix if you'd like to graphically see what the look like. This .mat file was not saved in "version 7.3" Matlab format, so it is easy to read in Python using `scipy.io.loadmat`.

Perform the following three types of dimensionality reduction methods:

a) Write your own code to run PCA on the matrix, using either SVD or eigenvalue routines. Don't forget to center your variables first. Keep the top 2 or 3 principal components.

b) Use a Gaussian matrix to project the data to 2 or 3 dimensions.

c) Run the tSNE algorithm on the data to reduce it to 2 dimensions. You can download Matlab and Python code that runs tSNE from https://lvdmaaten.github.io/tsne/; alternatively, it is implemented in the Python scikit-learn package in `sklearn.manifold.TSNE` in case you already have that installed.

Deliverables: For each dimensionality reduction method, plot your data in 2 or 3 dimensions. You should color code it according to the true labels (or adjust the marker shape, etc.). Make some short concluding remarks about the effectiveness of each method (for your own benefit, you might want to make a 3D plot and then interact with it).

Problem 4: [JUST FOR FUN] (Not required) **The Waiting Time Paradox**. Buses arrive at a bus stop at random times according to a Poisson process with intensity λ. I arrive at time t and ask how much time $\mathbb{E}(W)$ on average I will have to wait for the next bus. There are two contradictory arguments.

a) "The lack of the memory of the Poisson process implies that the distribution of my waiting time should not depend on my arrival time. In this case $\mathbb{E}(W) = 1/\lambda$."

b) "The time of my arrival is chosen at random in the interval between two consecutive buses, and for reasons of symmetry, my expected waiting time should be half the time the expected time between two consecutive buses, that is $\mathbb{E}(W) = 1/(2\lambda)$."

Which answer is correct? Carefully explain why the other answer is incorrect. [exercise taken from https://statweb.stanford.edu/~candes/acm116/Hw/hw4.pdf; you can look on the internet *after* attempting the problem, and there are many good explanations and simulations and comparison with real data]