STA 141C: Homework 1

- Homework due in Canvas: 04/28/2019 at 11:59PM. Please follow the instructions provided in Canvas about homeworks, carefully.

1. **Linear algebra basics: 5 points**. Answer true or false for each of the question below and give justification.

   (a) A rectangular matrix of size $n \times m$ is a *linear* transformation.

   (b) Only square matrices have Eigenvalue decompositions.

   (c) `Power Method` can be used to find only eigenvectors (and not singular vectors).

   (d) Singular vectors are orthogonal to each other.

   (e) The complexity of multiplying two $n \times n$ matrices *approximately* is $\mathcal{O}(n^3)$.

2. **Python practice via statistical concepts: 10 points**. This questions helps you brush-up your numerical computing skills in python by implementing some basic statistics concepts. Let X be a random variable that takes values $+1, -1$ with equal probability. That is:

$$P(X = +1) = P(X = -1) = 1/2.$$

   Generate $N = 10,000$ datasets, each of which has n data points. For this simulation, we consider $n = \{10, 100, 1000, 10000\}$. (Hint: Write a function that samples from the uniform distribution between 0 and 1. If the result is less than 0.5, set it to -1. Otherwise, set it to 1). Let $\overline{X}_n^{(i)}$ be the sample average of i^{th} dataset, $\mu = E(X) = 0$ and $\sigma^2 = \mathrm{Var}(X) = 1$. (Hint: Once you compute the sample averages, you will not need the individual data points from each dataset. Therefore, to save memory, you need only store the $\overline{X}_n^{(i)}$ rather than all the data points. It is highly recommended that you do this to avoid freezing or crashing your computer). Plot and intepret the following:

   (a) $\log_{10}(n)$ v.s. $\overline{X}_n^{(1)} - \mu$;
   (Hint: This plot illustrates how the deviation $\overline{X}_n^{(1)} - \mu$ converges to 0 as n increases).

   (b) Draw $\log_{10}(n)$ v.s. $\frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\{|\overline{X}_n^{(i)} - \mu| > \epsilon\}$ for $\epsilon = 0.5, \epsilon = 0.1, \epsilon = 0.05$;
   (Hint 1: This plot illustrates the convergence of empirical averages to true expectation.)
   (Hint 2: For some statement S, the indicator function $\mathbb{I}\{S\}$ is defined as $\mathbb{I}\{S\} = 1$ if S is true and $\mathbb{I}\{S\} = 0$ otherwise.)

   (c) Draw histograms of $\sqrt{n}(\overline{X}_n^{(i)} - \mu)/\sigma$ for N datsets for $n = 10, n = 1,000, n = 10,000$. You may choose your histogram bins or you may let Python choose automatically–any meaningful plot will do.
   (Hint: This plot illustrates the Central Limit Theorem.)

3. **Principal Component Analysis: 10 points**. In this problem, we will run PCA algorithm on the MNIST data set. The data set consists of .pgm images corresponding to digits 0, 1 and 2. Follow the steps below.

(a) Load all the images in Python. The easiest way to load .pgm images is by using the opencv python package[1]. If you do so, you want to import cv2 package and use the command `cv2.imread('00002.pgm', -1)`. This would give you the grey-scale pixel image in the form of a matrix. You are welcome to use any other method to load the images in the form of a matrix.

(b) Once you read the image from the 3 different groups (0, 1 and 2), calculate the mean image corresponding to each group and plot it.

(c) Now convert each image into a vector and form a matrix, which is of size `number of images` $\times$ `number of dimensions` and standardize it.

(d) Now perform PCA on the standardized data matrix, with `number of components`$= 2$, using Python's in-built command.

(e) Transform the data into the two PC coordinates and draw a scatter-plot of principal component 1 versus principal component 2.

(f) Do you observe any cluster structure in the scatter-plot ? If so explain. If not, explain why not.

[1] https://docs.opencv.org/3.4/d0/de3/tutorial_py_intro.html