CS/ECE 3501: Foundations of Data Analysis **Due Fri 3/29**

Homework 3: Linear Regression & SVD

Instructions: Submit a single Jupyter notebook (.ipynb) of your work to canvas by 11:59pm on the due date. All code should be written in Python. **Be sure to show all the work involved in deriving your answers! If you just give a final answer without explanation, you may not receive credit for that question.**

You may discuss the concepts with your classmates, but write up the answers entirely on your own. Do not look at another student's answers, do not use answers from the internet, and do not show your answers to anyone.

1. (60%) In this problem we will be analyzing data from the Old Faithful geyser in Yellowstone National Park. (See `https://en.wikipedia.org/wiki/Old_Faithful` to learn more!) Download the provided CSV data (`faithful.csv`), which consists of two variables: the duration of eruptions in minutes (`eruptions` column) and the length of time until the next eruption (`waiting` column). **Hint:** The function `numpy.asarray()` can convert a pandas column into a numpy array.

   (a) Load the `eruptions` column as your `x` variable and the `waiting` column as your `y` variable. Plot the data with a scatter plot. Do you think there is a relationship between eruption time and waiting time?

   (b) What is the mean of the eruption time? What is the mean of the waiting time? Use these values to center your `x` and `y` data.

   (c) Using the dot product formula we discussed in class, compute the correlation of eruption and waiting time. Does the value (and sign) of the correlation match what you would expect from the plot?

   (d) Using the formulas from lecture for $\hat{\alpha}$ and $\hat{\beta}$, compute the intercept and slope for a linear regression. Now plot a scatterplot with your regression line on top of it. How does the value and the sign of the slope compare to the correlation?

   (e) Say you are watching the Old Faithful geyser, and you time an eruption to be 2.2 minutes. Based on your regression analysis, how long should you expect to wait for the next eruption?

   (f) Using the formula for the R^2 statistic from class, what is the proportion of variance explained by your regression?

2. (40%) In this problem you are going to use SVD to compute an optimal rotation matrix to align two shapes. This is known as the **Orthogonal Procustes Problem** (see more here: `https://en.wikipedia.org/wiki/Orthogonal_Procrustes_problem`).

   Load the two matrices `hand1.dat` and `hand2.dat`. They are x and y coordinates of points of two hand shape outlines. Each row is a point, and there are 72 points, giving you two 72×2 matrices, A_1 and A_2. Now the optimal rotation that aligns hand shape 2 (A_2) with hand shape 1 (A_1) with the following steps:

   - Create the matrix $A_1^T A_2$.
   - Compute the SVD: $USV^T = A_1^T A_2$.

- The optimal rotation is $R = UV^T$.

Do the following:

(a) Plot the two hand shapes by connecting consecutive points with line segments. (You might want to use two different colors for the two hands.)

(b) Perform the steps outlined above to find the optimal rotation that aligns hand 2 into hand 1. What is the angle of rotation between these two hands?

(c) Rotate hand 2 using the optimal rotation you found. Plot the two hands again (this time with hand 2 rotated). Do they align with each other?

3. (**Bonus.** 15%) Consider two regression models (linear vs. quadratic):

$$y_i = ax_i + b + \epsilon_i, \tag{1}$$

$$y_i = ax_i^2 + bx_i + c + \epsilon_i, \tag{2}$$

(a) Assume we have n samples: $\{x_1, \cdots, x_n\}, i \in \{1, \cdots, n\}$ with their corresponding y values: $\{y_1, \cdots, y_n\}$ (as shown in Figure 1). Which model do you think would fit the data better? Briefly explain why.

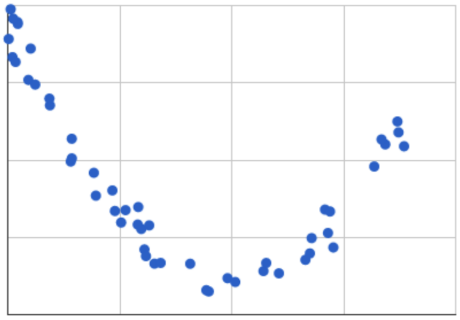

Figure 1: Regression data samples.

(b) Write out the optimization energy function of the regression model you pick in the question above by using least squares minimization. Derive the optimal value of a.

4. (**Bonus.** 10%) [Revisiting kNN classification.] One of the biggest problems with kNN classifiers is that they are very expensive to apply at test time, even if you use clever data structures and clever applications of the triangle inequality. One way to speed up a kNN classifier would just be to have fewer training points. Suppose you're given a training set of N labeled pairs $(x_n, y_n)_{n=1}^N$, and we want to throw out some subset of these points. What criteria would you use for deciding which points to throw out? Sketch an algorithm for doing so.