Math 123 **Problem Set 5** **Due: Sunday 4/17 at 11:59 p.m. EST**

Instruction: Read the homework policy. You should submit a PDF copy of the homework and any associated codes on Gradescope. Your PDF must be a single file.

1. We consider the clustering of the two-moon data. The following problems should be implemented in a programming language of your choice.

(a) Load the data `Moon.mat` and `Moon_Label.mat` on Canvas (CSV files are also available). Using the Gaussian Kernel $\mathbf{W} = \exp^{-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2}{2\sigma^2}}$, construct the Laplacian. Note that $\boldsymbol{x}_i$ refers to the i-th data point.

(b) Describe the connectivity of the graph in the limiting cases (a) as $\sigma \to \infty$ and (b) as $\sigma \to 0$.

(c) Run K-means clustering on the spectral embedding of the data. Use the spectral embedding to be the the 2-nd and 3-rd eigenvectors of the Laplacian corresponding to the two nonzero smallest eigenvalues. Experiment with different values of σ. For your optimal result, plot the data and indicate the labels using colors (e.g. points assigned to cluster 1 will be labeled red and points assigned to cluster 2 will be labeled blue).

(d) For your optimal result, plot the spectral embedding of the data.

(e) What does your result in (c) inform you?

Remark 1: Due to numerical issues, it's possible that an eigenvalue that should be 0 becomes a very small number. You should consider eigenvalues to be 0 if its absolute value is less than 10^{-12} or $1e-12$ in this problem.
Remark 2: The same applies if you see a complex number. If the complex part is less than 10^{-12} or $1e-12$, you can proceed with the real part in this problem.
Remark 3: For a given choice of σ, you should run your K-means algorithm with multiple (≥ 10) initializations/replicates. For this problem, you can use an inbuilt k-means function in MATLAB or Python.

2. Consider a set of n distinct data points in $\mathcal{R}^2$ $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), ..., (x_n, y_n)\}$ with $n \gg 3$. We want to fit the data using quadratic regression in the least squares sense. The corresponding minimization problem is defined below.

$$\min_{c_0, c_1, c_2} \sum_{i=1}^{n} \left(y_i - [c_0 + c_1 x_i + c_2 x_i^2]\right)^2$$

(a) Prove that the above minimization problem is equivalent to the following optimization problem

$$\min_{\boldsymbol{c}} ||\boldsymbol{y} - \boldsymbol{A}\boldsymbol{c}||_2^2,$$

where $\boldsymbol{c} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix}$, $\boldsymbol{A} = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{bmatrix}$ and $\boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$.

(b) Prove that the least squares quadratic regression solution satisfies the equation $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{c} = \boldsymbol{A}^T\boldsymbol{y}$.

(c) Given the equation in (b), a solution for the least squares quadratic regression is $\boldsymbol{c}^* = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{y}$. Prove $\boxed{\boldsymbol{A} \text{ has full column rank}}$ is the necessary and sufficient condition for $\boxed{\boldsymbol{A}^T\boldsymbol{A} \text{ to be invertible}}$.
(Hint: use SVD and then consider a matrix is full rank if and only if all diagonal components of D are non-zero.)

(d) Prove that $\boldsymbol{A}$ has full column rank.
[**Remark**: The n data points are distinct.]
[**Hint**: Show that $\boldsymbol{A}\boldsymbol{c} = \boldsymbol{0}$ has a trivial solution. You can use the fundamental theorem of algebra.]

(e) Find the solution to the equation in (b) using the singular value decomposition of $\boldsymbol{A}$.

3. For this problem, we will do some data analysis on past movies based on different criteria. [For this problem you can use in-built functions in Matlab/Python for any models you need.]

(a) Load the data `tmdb_movies_data.csv` on Canvas.

(b) Do some data cleaning: Discard all entries that are missing a number or have a zero value for the feature *Budget* (that is the movie budget in dollars). Discard all entries that are missing a number or have a zero value for the feature *Revenue* (that is the movie revenue in dollars) Discard all movies with a *Popularity* below 2. Remove all features apart from the following: *Popularity, Budget, Revenue, Runtime, Release year*.

(c) We will test how well a k-nearest neighbors algorithm (kNN) can identify whether a movie was created pre- or post-2000s. Create a new feature column *Post-2000* that has a value of 0 if the movie was released pre-2000s and 1 if the movie was released on or after 2000 (use the existing feature *Release year*). Randomly select 1000 movies from the dataset to be the "testset" Y, and the remaining movies to be the "training set" X (do not include the feature *Release year*). Run the kNN algorithm for $k = 5, 10, 15, 20, 25$ neighbors to classify the movies in Y as post-2000 or not (i.e., predict the feature *Post-2000*). In each case of k, how many out of the 1000 movies did kNN get right?

(d) Your friend, who is also a movie fanatic, accidentally realizes you are a data analysis master and asks you to help them with their movie project. They tell you that they have a set of movies they want you to find the expected revenue of a movie in dollars, if they give you the values for all the other features *Popularity, Budget, Runtime, Release year*. You quickly think that you can build a linear regression model to predict the revenue from the other features. After you do so, can you interpret the regression coefficients for the features *Popularity, Budget, Runtime, Release year*? Which of the features are most important? [*Think whether you need to do other data pre-processing before you build your regression model (e.g., standardizing).*]

(e) Your friend looks at the regression model and is slightly disappointed because they thought regression is too "basic". They ask you whether you can predict the revenue using kNN. Do you think you can do something like that and how? If you can, predict the revenue in dollars of a movie called "MATH123", which had a popularity score of 4, a budget of $10M$ dollars, run for 96 minutes, and was released in 2005.