Midterm Exam 2

- When you finish the exam, please upload your solutions to Gradescope by **Sunday 4/3 11:59pm**.
- This midterm is open book, open note, and you are free to use any computational resources like Python or Matlab.
- You may **not** talk to each other about the exam and you may **not** copy solutions from the internet.
- As you work, please recall the Tufts University statement on academic integrity:

  > "Academic integrity is the joint responsibility of faculty, students, and staff. Each member of the community is responsible for integrity in their own behavior and for contributing to an overall environment of integrity at the university."

1	/10
2	/15
3	/14
4	/1
Total	/40

1. Suppose we have a dataset X in $\mathbb{R}^{m \times D}$. This means we have m datapoints and D features. An explicit goal of PCA (and many of the other tools we use in data analysis!) is to reduce the dimension of data without losing the underlying variance of the data. The percent of variance explained by the k^{th} eigenvalue is given by

$$\frac{\mid \lambda_k \mid}{\sum_{i=1}^{D} \mid \lambda_i \mid}.$$

If we sort our eigenvalues in descending order, $\lambda_1 \geq ... \geq \lambda_D$ then the cumulative percent of variance for the first d eigenvalues is given by

$$\frac{\sum_{i=1}^{d} \mid \lambda_i \mid}{\sum_{i=1}^{D} \mid \lambda_i \mid}.$$

One way to to make sure that we choose the right number of principal components, is to choose the smallest possible d^* such that at least 95% of our variance is explained by the first d^* principal components, that is,

$$d^* = \min \left\{ d \;\; | \;\; \frac{\sum_{i=1}^{d} \mid \lambda_i \mid}{\sum_{i=1}^{D} \mid \lambda_i \mid} \geq .95 \right\}.$$

(a) *(2 points)* Intuitively, describe a dataset X for which d^* would be very small.

(b) *(2 points)* Intuitively, describe a dataset X for which d^* would be very large.

(c) *(2 points)* Suppose that X has the $D \times D$ identity matrix as its covariance matrix (this means that the covariance between any two features is 0). What are the eigenvalues associated to this covariance matrix? (*Hint: remember, eigenvalues can occur with multiplicity greater than 1.*)

(d) *(2 points)* If $D = 17$ for the dataset X decribed in part (c), what is the value of d^* in this case?

(e) *(2 points)* If $D = 1524$ for the dataset X decribed in part (c), what is the value of d^* in this case?

2. For this problem, we will improve on the k-means algorithm and use it for customer segmentation.

(a) *(3 points)* We will first develop a better version of the k-means algorithm. Your code should take as input a 2-dimensional dataset ($X \in \mathbb{R}^{m\times 2}$) and the number of clusters (k) and should output the labels for each data point (i.e., which cluster each data point belongs to) after the centroids stop moving. Assume that your dataset is already normalized.[You should use your HW4 code as a basis]. **Provide your entire k-means code with your submission. If for any reason, you can't complete this part correctly, use a correct k-means algorithm available either from the Lectures or the Homeworks to complete the rest of the questions.**

To initialize the clusters follow these steps:

(i) Select the first centroid, c_1, to be a random point from your dataset X.

(ii) Find the Euclidean distances of all data points to c_1 and save them in a matrix $D \in \mathbb{R}^{mxk}$ (eventually, the first column of D will have the distances to c_1 and the second column of D the distances to c_2 etc.)

(iii) Create a minimum distance vector $D_{\min}(i) = \min\{D(i,:)\}$ (Initially, there is only one column in D, but you will have more columns as you add centroids)

(iv) Turn the vector $D_{\min}$ into a probability vector $P(i) = \frac{D_{\min}(i)}{\mu_{D_{min}}}$, by diving with the sum of all the distances. (*Hint: remember that all entries of a probability vector must sum to 1.*)

(v) Turn P into a cumulative probability vector P_c such that $P_c(i) = \sum_{k<=i} P(k)$ (i.e., an element in P_c is equal to adding all elements up to, and including that index in the probability vector P). Check that the last element is $P_c(m) = 1$.

(vi) Generate a random number $r \in [0, 1]$ from a uniform distribution

(vii) Pick the second centroid, c_2, to be the data point that corresponds to the minimum cumulative probability greater or equal to r. In other words,

$$c_2 = (x_j, y_j) \text{ where } j = \min\{i | P_c(i) >= r\}$$

(viii) Repeat steps (ii) - (vii) using the same ideas but expanded for the number of centroids you have at each step, until you have a centroid for all clusters k

(b) *(1 point)* Congratulations! A leading marketing firm hired you because of your innovative k-means algorithm. You are asked to create a marketing strategy for a client that operates large shopping malls. Your manager gives you the following customer dataset `Mall_Customers.csv`, available in the Exam 2 folder on Canvas.

Feature	Meaning
Customer	Participant index 1-200
Gender	Male/Female
Age	Age of adult participant
Income	Annual income in thousands of dollars
Spending score	a 1-100 score based on spending amount

The *Customer* feature is simply an index so you can remove it. As a first step, normalize the feature columns *Age, Income, Spending score*. Save the normalized dataset as $X \in \mathbb{R}^{200\times 4}$. **Show an output of X.**

(c) *(3 points)* To get used to your new task, first we will consider a 2-dimensional subset of the dataset. Work with the *Income, Spending score* features only. Consider *Income* as the x-coordinate and

Spending score as the y-coordinate. Run K-means with $k = 5$ clusters using the algorithm you developed in part (a). **Plot your clustering result, i.e. show a scatter plot of the points in different colors corresponding to their K-means label.** *Hint: You may want to run your k-means algorithm a few times to check that the resulting clusters are correct.*

(d) *(2 points)* You are questioning whether 5 clusters is a good number for this problem. Create a process to measure the inertia E_n(by measuring the distance between each data point and its centroid, squaring this distance, and summing these squares across all clusters). Repeat the clustering with k = 2, 3, 4, ..., 10. **Plot the inertia for k = 2, 3, 4, ..., 10. Was $k = 5$ a good number of clusters to pick?**

The inertia of the K-means objective is defined as follows:

$$E_n = \sum_{x_i \in C_1} (x_i - \mu_1)^2 + \sum_{x_i \in C_2} (x_i - \mu_2)^2 + ... + \sum_{x_i \in C_n} (x_i - \mu_n)^2,$$

where C_1, C_2, ..., C_n are the clusters we have for the specific case, and μ_1, μ_2, ..., μ_n are their respective centroids.

(e) *(3 points)* You are still not convinced whether you picked the right number of clusters and are thinking of checking using an alternative approach you learned in MATH123, the silhouette analysis. For each data point in your 2-D dataset, compute the silhouette coefficient:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

where $a(i)$ is the average distance between point i and all the other data points in the same cluster i belongs, and $b(i)$ is the minimum average distance between point i and all the other data points in the clusters i does not belong (first you have to compute the average distances between i and all the other data points of clusters i does not belong, and then take the minimum of those $k - 1$ averages). **Develop a code to compute the silhouette coefficient of your 200 data points for the clustering result you completed with $k = 5$.** *Note: If for any reason you cannot complete this part, you can use a build-in function to continue with the rest of the questions.*

(f) *(2 points)* The silhouette coefficient is a measure of how similar an object is to its own cluster compared to other clusters. The coefficient ranges from -1 to $+1$, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a negative value, then the clustering configuration may have too many or too few clusters. **Compute the silhouette coefficient values from the previous step for your data points, do you think $k = 5$ was a good number of clusters based on this criterion? What is the average $s(i)$?**

(g) *(1 points)* Your manager walks in as you got your first 2-D clustering result. She asks you to prepare a report for the mall client, answering the folliowing:

- **Can you give a description that characterizes the customers in each cluster, based on their income and spending habits?**
- **We want to market a new clothing product from a luxury brand that is launching exclusively in our mall, which customer cluster should we email about it? Why?**
- **What about a new restaurant chain that is opening up in the mall food hall offering affordable meals? Which customer cluster should we email about it?**

3. For this problem you will need the datasets `rings.csv` and `rings_labels.csv` available in the Midterm 2 folder on Canvas.

   (a) *(3 points)* Plot the 2-dimensional dataset `rings.csv`. You will see that it looks like two concentric circles. Call this dataset X, and notice that $X \in \mathbb{R}^{1000\times 2}$. The points in the inner circle have a clear affinity to one another, as do the points on the outer circle, and yet this presents a difficult clustering problem. **Explain why a regular PCA approach will not be effective in separating these data points into distinct clusters.**

   (b) *(4 point)* For any two data points x and y the radial basis kernel function is given by

$$k(x, y) = \exp\left\{-\frac{|| x - y ||^2}{2\ell^2}\right\}$$

   where $|| x - y ||^2$ denotes the squared Euclidean distance between x and y, and ℓ is a parameter measuring the expected reach of any single point (i.e. when ℓ is small we expect any single data points to have very little influence of the whole data set). Using the radial basis function with $\ell = 1$ compute the kernel matrix, K. **Without using any built in kernel or kernelPCA solvers, show the output for K.**

   (c) *(4 points)* Compute the first two principle components, $\{\alpha_1, \alpha_2\}$ of K (you can use a built-in eigen-solver to do this). Recall that we can transform each data point in $x \in X$, into the i^{th} principle component space by summing across

$$x' = \sum_{j=1}^{1000} \alpha_{ij} K(x, x_j)$$

   Compute the transformed dataset $X' \in \mathbb{R}^{1000\times 2}$ by doing this for each point and each principle component. **Show your output for X'.**

   (d) *(3 points)* Now let's check whether your attempt at clustering was successful. **Plot your transformed dataset and color it according to the labels in `rings_labels.csv`. How does it look?**

4. *(1 point)* What's something fun you're going to do to treat yourself after handing in this exam?