CSCI 183 HW 4

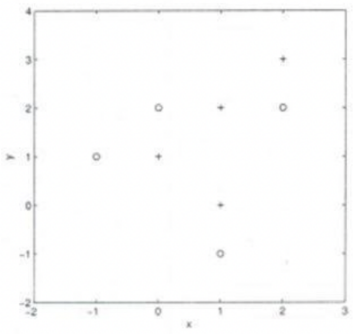
Lydia Myla and Parneet Kaur

Answer the following questions on K-NN:

1. **Suppose you have given the following data where x and y are the 2 input variables and Class is the dependent variable. (10 points)**

x	y	Class
−1	1	−
0	1	+
0	2	−
1	−1	−
1	0	+
1	2	+
2	2	−
2	3	+

- Below is a scatter plot which shows the above data in 2D space.



- What will be the Euclidean Distance between
  - the two data points A(2,2) and B(2,3)?
    - A: sqrt((2-2)^2 + (2-3)^2) = 1
  - the two data points C(1,-1) and D(1,0)?
    - A: sqrt((1-1)^2 + (0-(-1))^2) = 1
- Suppose you want to predict the class of new data point x=1 and y=1 using the Euclidean distance in 3-NN. In which class does this data point belong to and why?
  - A: If we use 3-NN, the data point will belong to the "+" class, represented by a "+" in the 2D graphs. This is because when the test point is plotted, we see that in the 3-NN circle, there are 3 "+" points. By using the majority rule, it is classified as the "+" point.
- In the previous question, if you are now wanting to use 7-NN instead of 3-NN, which of the following x=1 and y=1 will belong to?
  - A: If we use 7-NN instead of 3-NN, the data point will belong to the "-" class, represented by a circle in the 2D graph. This is because when the test point is plotted, we see that in the 7-NN circle, there are 4 "-" points and 3 "+" points. By using the majority rule, it is classified as the "-" point.

2. **State True/False for the following statements for k-NN classifiers? Justify your answer. [5 points]**

- classification accuracy is better with larger values of k
    - A: false, you need a k value that is not too large or too small
- The classification accuracy is best achieved with small values of k
    - A: false, you need a k value that is not too large or too small
- The hypothesis function is the most important aspect of k-NN
    - A: false, there isn't an exact hypothesis function for k-NN
- k-NN does not require an explicit training step
    - A: true, it uses the training data during the test time to make predictions
- k-NN is a non-parametric method of classification
    - A: true, theta values are not used in this form of classification

Answer the following questions on K-Means:

1. **For which of the following tasks might K-means clustering be a suitable algorithm. Select all that apply and justify your answer! (5 points)**

- Given a set of news articles from many different news websites, find out what are the main topics covered.
    - A: Yes, K-means clustering is a suitable algorithm for this task. We know that K-means clustering is an unsupervised learning algorithm that aims to partition a dataset into a predefined number of clusters (say k). In the context of news articles, the K-means algorithm can group the articles into clusters based on similarity in content. Therefore, by examining the articles within each cluster, we can infer the main topics covered by those articles. Articles within the same cluster are most likely to share common characteristics, such as subject matter, keywords, or themes, which ultimately allow us to infer the main topics covered within each cluster.
- Given historical weather records, predict if tomorrow's weather will be sunny or rainy.
    - A: No, K-means clustering is not a suitable algorithm for this task. K-means clustering does involve predicting future outcomes or dealing with time-series data (i.e. in this case, historical weather observations recorded at regular time intervals, such as temperature measurements taken every hour). Time-series data is important for weather forecasting because it can also allow users to verify the accuracy of weather forecasts by comparing the predicted weather conditions with present/observed weather predictions. However, K-means is an unsupervised learning algorithm, which does not have the target feature given, so there is generally no way of maximizing accuracy in K-means in comparison to supervised learning algorithms (e.g. Classification).
- From the user usage patterns on a website, figure out what different groups of users exist.

    - A: Yes, K-means clustering is a suitable algorithm for this task. K-means can cluster users based on their usage patterns, identifying different groups of users with similar behaviors. For example, users who frequently visit certain pages, spend a lot of time on the website, or make similar types of interactions (e.g. watching certain videos or reading articles on websites) can be clustered together.
- Given a database of information about your users, automatically group them into different market segments.
    - A: Yes, K-means clustering is a suitable algorithm for this task. K-means, being an unsupervised learning algorithm, does not require prior knowledge of market segments and results. A database can be very large or relatively medium in size, but K-means is suitable for any size (especially larger datasets). Given that the task required "automatic" grouping/clustering, K-means can identify similarities among users and cluster them accordingly without the need for heavy manual computation. This simplicity contributed to the algorithm's speed.
- Given sales data from a large number of products in a supermarket, figure out which products tend to form coherent groups (say are frequently purchased together) and thus should be put on the same shelf.
    - A: Yes, K-means clustering is a suitable algorithm for this task given that we want to partition it into K mutually exclusive clusters. If certain products are frequently purchased together, they are likely to exhibit similar characteristics or patterns in the sales data, making them suitable for clustering. Although determining the optimal K clusters, the algorithm still demonstrates practical applicability in comparison to other unsupervised learning methods such as hierarchical clustering. Hierarchical clustering can become computationally expensive as the number of products at the supermarket (i.e. our whole dataset being used) increases.
- Given sales data from a large number of products in a supermarket, estimate future sales for each of these products.
    - A: No, K-means clustering is not a suitable algorithm for this task. For predicting future sales for each product in a supermarket, we would have to use Regression (i.e. a supervised learning algorithm) because regression involves estimating a continuous variable (in this case, sales volume or revenue) based on different input features. K-means does not use labeled data and, rather, partitions data into K clusters (with each cluster represented by its centroid) based on identifiable similarities. Given that Regression models provide a quantitative result, essential for predicting future sales accurately, this method should be used in comparison to K-means.

2. **Suppose you have an unlabeled dataset $\{x^{(1)}, \ldots, x^{(m)}\}$. You run K-means with 50 different random initializations and obtain 50 different clusters of the data. What is the recommended way for choosing which one of these 50 clusters to use? Explain your answer. (5 points)**

- Plot the data and the cluster centroids, and pick the clustering that gives the most "coherent" cluster centroids.
    - A: Plotting data and cluster centroids for 50 different initializations becomes impractical for large datasets with numerous data points and dimensions. Visualizing high-dimensional data and multiple sets of cluster centroids can be overwhelming as well. Alongside this, visual inspection of a "coherent" cluster can be difficult due to its subjectivity and, in some cases, overlapping clusters.
- Manually examine the clusters and pick the best one.
    - A: This method would not be practical in this case because of the large size of the dataset itself. Initially, it may appear that the cluster is visually well-separated, but it may not have the best clustering when evaluated using objective metrics, such as inertia. Manual inspection introduces the possibility of bias, as the interpretation of what constitutes a "good" cluster may vary based on the individual conducting the inspection. This bias, then, can lead to inconsistent or unreliable cluster selection.
- The only way to do so is if we also have labels for our data.
    - A: Having labeled data is important for supervised learning algorithms/models, it is generally not required in unsupervised tasks. In this task, we specifically know that we have an unlabeled dataset, so this method is not applicable over here.
- For each of the clusters, compute 'Inertia', and pick the one that minimizes the sum of this.
    - A: This is the most reasonable approach out of the all provided methods. Inertia provides a quantitative measure for how well the data points are clustered around the centroid. That is, it quantifies the compactness of the clusters, where lower inertia indicates tighter clusters. Additionally, in K-means clustering, the goal is to minimize the cluster sum of squares, which is the same as minimizing inertia. Given that we do not have any mentioned outliers in this case, we do not need to worry about inflation of inertia values (i.e. any biases).

3. **Which of the following statements are true? Select all that apply and explain your answer for each choice. (5 points)**

- On every iteration of K-means, the loss function (inertia) should either stay the same or decrease; in particular, it should not increase.
    - A: True; In K-mean the goal is to minimize the inertia. We want the points within the same cluster to be similar to each other. Hence, the distance between them should be as low as possible. During every iteration of the algorithm, data points

are assigned to the nearest centroid, and then centroids are updated based on the squared Euclidean distance. Since the goal is to minimize inertia, the algorithm should converge towards a solution with lower inertia. Specifically, convergence is when the centroids no longer change significantly or until a specified number of iterations is reached.

- A good way to initialize K-means is to select K (distinct) examples from the training set and set the cluster centroids equal to these selected examples.
    - A: True; This is just one of the two ways to select centroids. The method is often straightforward and easy to implement in comparison to other methods. For example, if the dataset is: (1, 2), (2, 3), (3, 4), (4, 5), then randomly selecting K distinct examples may mean we choose (2,3) and (3,4) as the initial centroids.
- K-Means will always give the same results regardless of the initialization of the centroids.
    - A: False; K-means clustering aims to converge on an optimal set of centroids and applies successive iterations to do so. Therefore, the more optimal the positioning of the initial centroids, the fewer iterations there will be for convergence. This indicates that K-means is sensitive to the initial cluster centroids. For example, if initial centroids are chosen in close proximity to one another, it might lead to clusters that have a lot of overlap and fail to separate points into distinct clusters.
- Once an example has been assigned to a particular centroid, it will never be reassigned to another different centroid.
    - A: False; - When the cluster center is re-calculated and changed for the next iteration of the K-means algorithm, the points/examples that are now closer to the new centroid may be re-classified to another cluster (depending on the value). The K-means algorithm is iterated until the centroids and the cluster assignments no longer change. Until this happens, a point may change which centroid it is assigned to.
- For some datasets, the “right” or “correct” value of K (the number of clusters) can be ambiguous, and hard even for a human expert looking carefully at the data to decide.
    - A: True; This can happen when there are overlapping clusters, uneven data dimensionality, or complex data distribution. Another case may be with a large number of data points, where the underlying structure of the data may become more complex, making it harder to discern the optimal number of clusters. On the other hand, with fewer data points, it may be difficult to identify meaningful clusters if any.
- The standard way of initializing K-means is setting it to be equal to a vector of zeros.
    - A: False; The typical methodology of K-means clustering is randomly selecting K data points as initial cluster centroids or using another methodology. Placing all centroids at the origin (where all feature values are zero) assumes that the most representative position for each cluster center is at this point. In most cases, the

data points spread out across the feature space. The centroid should be placed at a position that captures the central tendency of the data points belonging to that cluster.

4. **Use K-Means Algorithm to create three clusters - [10]. You may choose to code or solve it on paper.**

Point	Coordinates
A1	(2,10)
A2	(2,6)
A3	(11,11)
A4	(6,9)
A5	(6,4)
A6	(1,2)
A7	(5,10)
A8	(4,9)
A9	(10,12)
A10	(7,5)
A11	(9,11)
A12	(4,6)
A13	(3,10)
A14	(3,8)
A15	(6,11)

- Assume A2(2, 6), A7(5,10) and A15(6,11) are initialized centers of the clusters. Just show until the first iteration.
    - A:

```
import numpy as np
centers = np.array([[2, 6], [5, 10], [6, 11]])
data_points = np.array([[2, 10], [11, 11], [6, 9], [6, 4], [1, 2], [4, 9], [10, 12],
                        [7, 5], [9, 11], [4, 6], [3, 10], [3, 8]])

labels = []
for point in data_points:
    distances = [np.sqrt(np.sum((point - center) ** 2)) for center in centers]
    labels.append(np.argmin(distances))

new_centers = np.array([data_points[labels == i].mean(axis=0) for i in
range(len(centers))])

print("Initial Cluster Centers:")
print(centers)
print("After First Iteration:")
print(labels)
print("Updated Cluster Center:")
print(new_centers)
```