**KMeans**

K-means clustering is a type of unsupervised machine learning algorithm used to classify a given dataset into clusters based on their similarity. The main objective of the algorithm is to minimize the sum of squared distances between the data points and their respective cluster centroid.

The algorithm starts with an initial set of k centroids, where k is the number of clusters that we want to divide the data into. Then, the data points are assigned to the closest centroid based on the Euclidean distance. After all the data points are assigned to the clusters, the centroids are recalculated as the mean of the data points in that cluster. This process is repeated until the centroids no longer move or a maximum number of iterations is reached.

K-means is a widely used algorithm for clustering in various fields such as computer science, data mining, pattern recognition and image processing. It's simple and easy to implement, and it's known for its computational efficiency. However, it has some limitations, such as the assumption of spherical cluster shape and the sensitive to initial centroid position.

K-means clustering is a method of clustering, which is a technique for grouping similar data points together. The goal of the algorithm is to partition a set of data points into K clusters, where each cluster is defined as a group of points that are similar to each other and dissimilar to points in other clusters.

The algorithm works by first initializing K cluster centroids, where K is the number of clusters desired. This is typically done by randomly selecting K data points to serve as the centroids. Then, each data point is assigned to the cluster associated with the nearest centroid. Once all data points have been assigned, the cluster centroid is recomputed as the mean of all the data points assigned to that cluster. This process is repeated until the cluster assignments no longer change or a maximum number of iterations is reached.

K-means is a popular and widely used algorithm due to its simplicity and efficiency. However, it is sensitive to initial centroid position and can get stuck in local minima, which can lead to suboptimal results. There are also variations of K-means, such as K-means++ which is a more sophisticated initialization method that addresses this issue.

K-means clustering is a method of clustering, which is a technique for grouping similar data points together. The goal of the algorithm is to partition a set of data points into K clusters, where each cluster is defined as a group of points that are similar to each other and dissimilar to points in other clusters.

The algorithm works by first initializing K cluster centroids, where K is the number of clusters desired. This is typically done by randomly selecting K data points to serve as the centroids. Then, each data point is assigned to the cluster associated with the nearest centroid. Once all data points have been assigned, the cluster centroid is recomputed as the mean of all the data points assigned to that cluster. This process is repeated until the cluster assignments no longer change or a maximum number of iterations is reached.

K-means is a popular and widely used algorithm due to its simplicity and efficiency. However, it has some limitations, one of the main disadvantage is that it's sensitive to initial centroid position and it can stuck in local minimum. It also assumes that the clusters are spherical and equally sized, which may not be the case for all datasets. Additionally, the algorithm requires the number of clusters to be specified in advance, which can be difficult to determine.

There are also variations of K-means, such as K-means++ which is a more sophisticated initialization method that addresses the issue of getting stuck in local minima. Another variation is the fuzzy K-means algorithm, which allows data points to belong to more than one cluster with varying degrees of membership.

K-means is widely used in many applications, such as image compression, image segmentation, and customer segmentation in marketing. It also can be used as a preprocessing step for other machine learning algorithms, such as dimensionality reduction and anomaly detection.

**Introduction**

Machine learning is a method of teaching computers to learn from data, without being explicitly programmed. It is a subset of artificial intelligence that uses statistical techniques to enable computers to learn from data and make predictions or decisions without human intervention.

Unsupervised learning is a type of machine learning model where an algorithm is trained on an unlabeled dataset, where the desired output is not known. The goal of unsupervised learning is to discover hidden patterns or structure in the data. Unlike supervised learning, unsupervised learning does not use labeled data, and instead the algorithm must find a way to infer the underlying structure of the data on its own.

K-means clustering is a method of unsupervised learning, which is a technique for grouping similar data points together. The final deliverable of this coursework is the implementation of K-means clustering algorithm on the students unique id to generate a personal data matrix with different results depending on the id.

Along with it is an overview of machine learning, unsupervised learning and K-means clustering technique with the explanation of the implementation of this technique in MATLAB.

**Unsupervised Learning**

Unsupervised learning is a type of machine learning where an algorithm is trained on an unlabeled dataset, where the desired output is unknown. The goal of the algorithm is to find structure or patterns in the data, rather than to make predictions.

There are several types of unsupervised learning, including:

• Clustering: The goal is to group similar data points together, without any prior knowledge of what the groups should be. Example: K-means, Hierarchical Clustering

• Dimensionality Reduction: The goal is to reduce the number of features in the data while retaining as much information as possible. Example: PCA, t-SNE

• Anomaly Detection: The goal is to identify data points that are unusual or do not conform to the general pattern. Example: One-Class SVM, Isolation Forest

Unsupervised learning is used in a variety of applications such as and anomaly detection.

It is important to note that unsupervised learning is generally harder to evaluate than supervised learning, as there is no clear notion of accuracy.

**Implementation**

Implementation of K means clustering

We need to implement three methods to perform given tasks:

1. Randomly initializing K centroids

2. Assigning data points to K clusters

3. Re-computing K centroids

The first step will be carried out at the beginning of the training process. In contrast, steps 2 and 3 will be iterated until convergence. So let's begin by putting the first approach for random initialization of K centroids into practice.

 The above-described initCunitod will produce K centroids, which are random indices from X. The dataset's indices will be randomly rearranged via the rand per algorithm (X).

 The get Closest Centroids method calculates the Euclidean distance between each cluster centroid and each data point, assigning the data point to the cluster centroid with the smallest distance.



 Recalculating the cluster centroids is the final technique we must use. The mean of all the data points in each cluster will be calculated for this purpose, and the mean will be designated as the new centroid of the cluster.

 That's it; we now have everything we need to use the K-Means. The final piece of code (provided below) will execute the stated methods till maximum iterations are achieved. You can use various strategies for stopping criteria (such as not assigning data points to distinct clusters), but for the sake of simplicity, I'm going to use a maximum number of repetitions in this case. Automatically check the resources listed at the end of the tutorial if you are considering choosing K.

 For each K value, report the mean performance using the Silhouette measure and plot the Silhouette for each cluster (each K value) as shown in Figure1

 For k=2

K-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. The algorithm proceeds as follows:

1. Select k initial centroids for the k clusters. These can be chosen randomly from the data points, or by some other method.

2. Assign each data point to the cluster whose centroid is closest (where the distance metric is typically Euclidean distance).

3. Compute the new centroid of each cluster by taking the mean of all the points assigned to that cluster.

4. Repeat steps 2 and 3 until the centroids no longer move (i.e., they have stabilized).

5. Return the final assignments of data points to clusters.

It is important to note that the number of clusters, k, must be specified in advance. Additionally, the algorithm is sensitive to the initial conditions, so it is common to run the algorithm multiple times with different initial centroids and choose the best result.

Implement

K-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean.

To perform K-means Clustering an initial K centroids should be selected for a K number of clusters randomly from the data points. Next each data point should be assigned to the cluster whose centroid is closest. Then it is required to compute a new centroid for each cluster by taking the mean of all the points assigned to that specific cluster. The second and third step should be repeated for a desired number of times or until the centroids can no longer move. Finally, the final assignments of the data points should be returned to the clusters. The explanation of how this process was achieved in the coursework and the how the other requirements were fulfilled will be explained briefly below with code samples.

Initially the dataset is loaded with the student ID and assigned to X and then the Row count and Column count is displayed. Then two arrays are initialized to store the mean of each column and standard deviation of each column. Finally mean, standard deviation and histogram of each column is displayed in terminal and figure respectively using a for loop.

Next the Covariance and the Correlation Matrices of the data set X is set as cov\_matrix and cor\_matrix respectively to be stored in the workspace and the variable K which is the range of number of Clusters that we desire is also defined. The covariance and correlation matrices are stored in the workspace and can be viewed if required.

In this segment a range of values (clusters 3:5) is defined as K and K-means clustering is done with the defined K, the mean silhouette score of the K Cluster is entered into an array and the silhouette plot is generated. The silhouette will be generated from the predicted\_cluster\_indices which was produced as a result due to the K-means clustering and the data set. This block will repeat itself with the above for each instance of different cluster values in K.

Next the max or the best mean silhouette score of the mean silhouette scores array generated by different K values (Clusters) is obtained and will be printed on the terminal along with the plot of Mean Silhouette Scores of all K values.

Finally, a for loop is initiated to perform K-means clustering and obtain color maps to generate the scatter plot with its title, legend and axis for each number of K.

The result will include eleven figures with four histograms for each column , three mean silhouette score figures for each of K , one silhouette score plot plotting the three mean silhouette scores of the range of K and three scatter plots for each cluster. Furthermore mean and standard deviation of each column, Mean silhouette score for each of K and the max index of mean silhouette scores will be displayed on the terminal. Along with the covariance and correlation matrices being stored in the workspace. All of this will be displayed on the appendix.

**Stopping criteria**

The stopping criteria for k-means clustering are the conditions that determine when the algorithm should stop iterating. There are several ways to define the stopping criteria, but the two most common ones are:

1. Maximum number of iterations: This is the most straightforward method. The algorithm is set to run for a fixed number of iterations and stops after that, regardless of the cluster assignments. This method is simple to implement but may not be optimal, as the algorithm may stop before the cluster assignments have stabilized.

2. Convergence: This method is based on the idea that the algorithm has converged when the cluster assignments no longer change. The algorithm stops iterating when the cluster assignments for all data points remain the same for a given number of consecutive iterations, or when the changes in cluster assignments are below a certain threshold. This method is more sensitive to the specific characteristics of the dataset and can lead to more accurate cluster assignments.

Another stopping criteria is based on the improvement of the cost function, such as the sum of squared distances between points and their cluster centroid. The algorithm stops when the cost function is not improving enough or when it reaches a certain threshold.

It is important to note that the choice of stopping criteria will depend on the specific requirements of the application and the characteristics of the dataset. In some cases, a combination of these criteria may be used for more robust results.

The stopping criteria for k-means clustering are the conditions that determine when the algorithm should stop iterating. There are several ways to define the stopping criteria:

Maximum number of iterations: This is the most straightforward method. The algorithm is set to run for a fixed number of iterations, and stops after that, regardless of the cluster assignments. This method is simple to implement, but may not be optimal as the algorithm may stop before the cluster assignments have stabilized.

Convergence: This method is based on the idea that the algorithm has converged when the cluster assignments no longer change. The algorithm stops iterating when the cluster assignments for all data points remain the same for a given number of consecutive iterations, or when the changes in cluster assignments are below a certain threshold. This method is more sensitive to the specific characteristics of the dataset and can lead to more accurate cluster assignments.

Improvement in cost function: This method is based on the improvement of the cost function, such as the sum of squared distances between points and their cluster centroid. The algorithm stops when the cost function is not improving enough or when it reaches a certain threshold.

It is important to note that the choice of stopping criteria will depend on the specific requirements of the application and the characteristics of the dataset. In some cases, a combination of these criteria may be used for more robust results. In addition, stopping criteria must be chosen carefully, as it can impact the performance of the algorithm and the final outcome.