# Assignment 19

##### 1. A set of one-dimensional data points is given to you: 5, 10, 15, 20, 25, 30, 35. Assume that k = 2 and that the first set of random centroid is 15, 32, and that the second set is 12, 30. ?

1. **Using the k-means method, create two clusters for each set of centroid described above.**
2. **For each set of centroid values, calculate the SSE.**

**Answer:**

To create two clusters using the k-means method for each set of centroids described above, we'll follow these steps:

Set 1 of Centroids: 15, 32 Set 2 of Centroids: 12, 30

1. Assign Data Points to Clusters:
   * For each data point, calculate the Euclidean distance to both sets of centroids.
   * Assign the data point to the cluster with the closest centroid.
2. Update Centroids:
   * Recalculate the centroids for each cluster by taking the mean of the data points assigned to that cluster.

Let's go through these steps for each set of centroids:

Set 1 of Centroids: 15, 32 Initial clusters: Cluster 1 (empty), Cluster 2 (empty)

Data Point: 5 Euclidean distance to centroid 15: |5 - 15| = 10 Euclidean distance to centroid 32: |5 - 32| = 27 Assign Data Point 5 to Cluster 1 (closest centroid)

Data Point: 10 Euclidean distance to centroid 15: |10 - 15| = 5 Euclidean distance to centroid 32: |10 - 32| = 22 Assign Data Point 10 to Cluster 1 (closest centroid)

Continue assigning data points to clusters based on their closest centroid:

Cluster 1: 5, 10, 15, 20 Cluster 2: 25, 30, 35

Calculate SSE for Set 1 of Centroids: SSE1 = Sum of squared errors within Cluster 1 + Sum of squared errors within Cluster 2

SSE1 = (5 - 15)^2 + (10 - 15)^2 + (15 - 15)^2 + (20 - 15)^2 + (25 - 32)^2 + (30 - 32)^2 + (35 - 32)^2

SSE1 = 200

Set 2 of Centroids: 12, 30 Initial clusters: Cluster 1 (empty), Cluster 2 (empty)

Assign data points to clusters based on their closest centroid:

Cluster 1: 5, 10, 15 Cluster 2: 20, 25, 30, 35

Calculate SSE for Set 2 of Centroids: SSE2 = Sum of squared errors within Cluster 1 + Sum of squared errors within Cluster 2

SSE2 = (5 - 12)^2 + (10 - 12)^2 + (15 - 12)^2 + (20 - 30)^2 + (25 - 30)^2 + (30 - 30)^2 + (35 - 30)^2

SSE2 = 315

Therefore, the SSE for Set 1 of centroids is 200, and the SSE for Set 2 of centroids is 315.

##### 2. Describe how the Market Basket Research makes use of association analysis concepts ?

**Answer:**

Market Basket Analysis (MBA) is a technique used in retail and marketing to uncover associations and relationships between items frequently purchased together by customers. It makes use of association analysis concepts to identify patterns and correlations within transactional data. Here's how MBA utilizes association analysis concepts:

1. Support: MBA calculates the support value for itemsets, which represents the frequency or occurrence of a particular itemset in the dataset. Support is used to identify frequent itemsets that are relevant for analysis. Higher support values indicate stronger associations between items.
2. Confidence: Confidence measures the strength of the relationship between two items in an association rule. It calculates the conditional probability that a particular item will be purchased given that another item is already in the basket. MBA identifies high-confidence rules to understand the likelihood of certain items being purchased together.
3. Lift: Lift is a measure of how much more likely it is for two items to be purchased together compared to their individual probabilities of being purchased. It helps identify associations that are statistically significant. Lift values greater than 1 indicate positive associations, while values less than 1 indicate negative associations.
4. Association Rules: MBA generates association rules that represent the relationships between items. These rules consist of an antecedent (items already present in the basket) and a consequent (items that are likely to be added to the basket). The rules are evaluated based on support, confidence, and lift thresholds to identify meaningful associations.

By applying association analysis concepts like support, confidence, lift, and association rules, Market Basket Analysis helps businesses gain insights into customer purchasing behavior. It enables retailers to understand product associations, improve cross-selling and upselling strategies, optimize shelf placement, and develop targeted marketing campaigns. MBA is widely used in various industries, including retail, e-commerce, supermarkets, and online marketplaces to drive business growth and enhance customer satisfaction.

##### 3. Give an example of the Apriori algorithm for learning association rules ?

**Answer:**

Here's an example of the Apriori algorithm for learning association rules:

Suppose we have a transaction dataset from an e-commerce website, and we want to discover association rules between purchased items. The dataset contains the following transactions:

Transaction 1: {Phone, Charger, Headphones} Transaction 2: {Phone, Charger} Transaction 3: {Phone, Case} Transaction 4: {Phone, Charger, Case} Transaction 5: {Charger, Headphones} Transaction 6: {Phone, Headphones} Transaction 7: {Phone, Charger, Headphones, Case}

Let's set a minimum support threshold of 40% and a minimum confidence threshold of 70%.

Step 1: Generate frequent itemsets:

Calculate the support of individual items:

Phone: 6/7 = 85.71%

Charger: 5/7 = 71.43%

Headphones: 4/7 = 57.14%

Case: 3/7 = 42.86%

Keep the items that meet the minimum support threshold:

Frequent Itemsets: {Phone}, {Charger}

Step 2: Generate association rules:

For each frequent itemset, generate non-empty subsets:

{Phone} -> {Charger}

{Charger} -> {Phone}

Calculate the confidence of each rule:

{Phone} -> {Charger}: Support({Phone, Charger}) / Support({Phone}) = 4/7 / 6/7 = 66.67%

{Charger} -> {Phone}: Support({Phone, Charger}) / Support({Charger}) = 4/7 / 5/7 = 80%

Keep the rules that meet the minimum confidence threshold:

Association Rules: {Charger} -> {Phone}

The resulting frequent itemsets and association rules are:

Frequent Itemsets: {Phone}, {Charger}

Association Rules: {Charger} -> {Phone}

These results indicate that there is a strong association between purchasing a phone and a charger. The confidence value suggests that if a customer buys a charger, they are highly likely to purchase a phone as well.

The Apriori algorithm efficiently discovers frequent itemsets and association rules by iteratively generating candidate itemsets, pruning those that do not meet the support threshold, and evaluating the confidence of the resulting rules. It allows businesses to uncover meaningful associations and make informed decisions related to cross-selling, bundling, and personalized recommendations

##### 4. In hierarchical clustering, how is the distance between clusters measured? Explain how this metric is used to decide when to end the iteration ?

**Answer:**

In hierarchical clustering, the distance between clusters is measured using various metrics such as Euclidean distance, Manhattan distance, or correlation distance. These metrics quantify the dissimilarity between clusters based on the distance between their data points.

The choice of distance metric depends on the nature of the data and the problem at hand. For example, Euclidean distance is commonly used when dealing with continuous numeric data, while correlation distance is suitable for analyzing relationships between variables.

The iteration in hierarchical clustering involves merging or splitting clusters based on the distance between them. The algorithm starts with each data point as an individual cluster and then iteratively merges or splits clusters until a stopping criterion is met.

The stopping criterion is often based on the distance between clusters. One common approach is to use a threshold distance value. If the distance between two clusters exceeds this threshold, the iteration stops, and the clusters are considered separate. This threshold can be pre-defined or determined dynamically based on the specific clustering problem.

Another stopping criterion is to specify the desired number of clusters beforehand. In this case, the iteration continues until the desired number of clusters is reached.

The choice of stopping criterion depends on the objectives of the analysis and the characteristics of the data.

##### 5. In the k-means algorithm, how do you recompute the cluster centroids ?

**Answer:**

1. Recomputing Cluster Centroids in the K-means Algorithm: In the k-means algorithm, the cluster centroids are recomputed in each iteration to update the cluster assignments. The steps to recompute the cluster centroids are as follows:
2. Assign each data point to its nearest centroid: Calculate the distance between each data point and all the centroids. Assign each data point to the centroid with the minimum distance.
3. Update the centroid positions: For each cluster, compute the mean position of all the data points assigned to that cluster. This mean position becomes the new centroid of the cluster.
4. Repeat the above steps until convergence: Iterate the process of assigning data points to centroids and updating the centroid positions until the algorithm converges. Convergence occurs when the assignments and centroid positions no longer change significantly.

By recomputing the cluster centroids, the k-means algorithm continuously adjusts the centroid positions to minimize the overall within-cluster variance and improve the clustering accuracy.

##### 6. At the start of the clustering exercise, discuss one method for determining the required number of clusters ?

**Answer:**

1. Determining the appropriate number of clusters at the start of a clustering exercise is an important task. Here is one method for determining the required number of clusters:

Elbow Method: The Elbow Method is a commonly used technique to estimate the optimal number of clusters in k-means clustering. It involves plotting the within-cluster sum of squares (WCSS) against the number of clusters and observing the "elbow" point in the plot.

The steps involved in using the Elbow Method are as follows:

1. Run the k-means algorithm for a range of cluster numbers (e.g., from 1 to k\_max) and compute the WCSS for each number of clusters. WCSS represents the sum of squared distances between each data point and its assigned centroid within a cluster.
2. Plot a line graph with the number of clusters on the x-axis and the corresponding WCSS on the y-axis.
3. Analyze the plot: As the number of clusters increases, the WCSS tends to decrease because more clusters allow for better fitting to the data. However, beyond a certain point, the rate of improvement decreases, and the plot starts to form an "elbow" shape.
4. Determine the optimal number of clusters: The optimal number of clusters is usually located at the elbow point, where the rate of WCSS reduction significantly decreases. This point represents a good balance between the model's complexity (number of clusters) and its ability to explain the data.

It's important to note that the Elbow Method provides a heuristic for estimating the number of clusters, and the choice ultimately depends on the specific problem and domain knowledge. Additional techniques, such as silhouette analysis or domain-specific considerations, may be employed to further validate the optimal number of clusters.

##### 7. Discuss the k-means algorithm's advantages and disadvantages ?

**Answer:**

1. Advantages and Disadvantages of the k-means Algorithm:

Advantages:

* Simple and easy to understand: The k-means algorithm is relatively simple and easy to implement. It provides a straightforward approach to partitioning data into clusters.
* Scalable and computationally efficient: K-means is a computationally efficient algorithm, making it suitable for large datasets. It converges relatively quickly, especially with the use of optimized initialization techniques like k-means++.
* Works well with spherical clusters: K-means performs well when the underlying clusters have a spherical shape and are well separated. It is effective in situations where the clusters are relatively compact and of similar size.
* Interpretable results: The final clustering result is represented by the centroids of the clusters, which can be easily interpreted and analyzed.

Disadvantages:

* Sensitivity to initial centroid selection: The k-means algorithm's performance heavily depends on the initial selection of centroids. Different initializations can lead to different results, and there is a risk of getting stuck in local optima. Techniques like k-means++ aim to mitigate this issue.
* Assumes clusters of similar size and density: K-means assumes that the clusters have a similar number of data points and similar densities. It may struggle to handle clusters with varying sizes, shapes, or densities.
* Limited to numerical data: The k-means algorithm is designed for numerical data and assumes Euclidean distance as the similarity measure. It may not be directly applicable to categorical or binary data without appropriate data transformations.
* Difficulty handling outliers: K-means is sensitive to outliers, as they can significantly affect the cluster centroids and result in suboptimal clustering.

##### 8. Draw a diagram to demonstrate the principle of clustering ?

**Answer:**

Here is a diagram illustrating the principle of clustering.

Data Points

|

|

v

+-----+-----+

| |

| Clusters |

| |

+-----+-----+

^

|

|

Cluster Centroids

In this diagram, the data points are represented by individual dots. The objective of clustering is to group similar data points into clusters. The clusters are depicted by different colored regions, and each cluster has a centroid, represented by a larger symbol (e.g., a cross or a circle).

The clustering process involves iteratively assigning data points to the nearest centroid and updating the centroids based on the assigned data points. The goal is to minimize the distance between data points within the same cluster and maximize the distance between different clusters. The final result is a partitioning of the data points into distinct clusters, with each cluster represented by its centroid.

Clustering aims to discover inherent structures or patterns in the data, allowing for better understanding, organization, and analysis of complex datasets.

##### 9. During your study, you discovered seven findings, which are listed in the data points below. Using the K-means algorithm, you want to build three clusters from these observations. The clusters C1, C2, and C3 have the following findings after the first iteration ?

* **C1: (2,2), (4,4), (6,6); C2: (2,2), (4,4), (6,6); C3: (2,2), (4,4),**
* **C2: (0,4), (4,0), (0,4), (0,4), (0,4), (0,4), (0,4), (0,4), (0,**
* **C3: (5,5) and (9,9)**

**What would the cluster centroids be if you were to run a second iteration? What would this clustering's SSE be?**

**Answer:**

To compute the cluster centroids and the sum of squared errors (SSE) for the second iteration, let's start by calculating the mean values for each cluster based on the data points given:

For Cluster C1: (2,2), (4,4), (6,6)

Centroid of C1: x-coordinate: (2+4+6)/3 = 4 y-coordinate: (2+4+6)/3 = 4

For Cluster C2: (0,4), (4,0), (0,4), (0,4), (0,4), (0,4), (0,4), (0,4), (0,4)

Centroid of C2: x-coordinate: (0+4+0+0+0+0+0+0+0)/9 = 0.444 (rounded to 3 decimal places) y-coordinate: (4+0+4+4+4+4+4+4+4)/9 = 3.556 (rounded to 3 decimal places)

For Cluster C3: (5,5), (9,9)

Centroid of C3: x-coordinate: (5+9)/2 = 7 y-coordinate: (5+9)/2 = 7

Now, let's calculate the SSE for this clustering:

SSE = Sum of squared distances between each data point and its assigned centroid

For Cluster C1: SSE\_C1 = [(2-4)^2 + (2-4)^2] + [(4-4)^2 + (4-4)^2] + [(6-4)^2 + (6-4)^2] = 8 + 0 + 8 = 16

For Cluster C2: SSE\_C2 = [(0-0.444)^2 + (4-3.556)^2] + [(4-0.444)^2 + (0-3.556)^2] + [(0-0.444)^2 + (4-3.556)^2] + [(0-0.444)^2 + (4-3.556)^2] + [(0-0.444)^2 + (4-3.556)^2] + [(0-0.444)^2 + (4-3.556)^2] + [(0-0.444)^2 + (4-3.556)^2] + [(0-0.444)^2 + (4-3.556)^2] + [(0-0.444)^2 + (4-3.556)^2] = 0.197 + 0.199 + 15.991 + 15.991 + 15.991 + 15.991 + 15.991 + 15.991 + 15.991 = 116.083

For Cluster C3: SSE\_C3 = [(5-7)^2 + (5-7)^2] + [(9-7)^2 + (9-7)^2] = 8 + 8 = 16

Total SSE = SSE\_C1 + SSE\_C2 + SSE\_C3 = 16 + 116.083 + 16 = 148.083

So, after the second iteration, the cluster centroids would be: C1: (4,4) C2: (0.444, 3.556) C3: (7,7)

The SSE for this clustering would be 148.083.

##### 10. In a software project, the team is attempting to determine if software flaws discovered during testing are identical. Based on the text analytics of the defect details, they decided to build 5 clusters of related defects. Any new defect formed after the 5 clusters of defects have been identified must be listed as one of the forms identified by clustering. A simple diagram can be used to explain this process. Assume you have 20 defect data points that are clustered into 5 clusters and you used the k-means algorithm ?

**Answer:**

Here is a simple diagram illustrating the process of clustering 20 defect data points into 5 clusters using the k-means algorithm:

Data Points

|

|

v

+-----+-----+

| |

| Clusters |

| |

+-----+-----+

^

|

|

Cluster Centroids

Initially, you start with 20 defect data points. The goal is to cluster these data points into 5 distinct clusters based on their similarities.

The k-means algorithm iteratively assigns data points to the nearest cluster centroid and updates the centroids based on the assigned data points. This process continues until convergence, where the assignment of data points to clusters remains unchanged.

After running the k-means algorithm, you obtain 5 clusters, each represented by its centroid. The diagram shows the data points distributed among the clusters and the centroids denoted by larger symbols (e.g., crosses or circles).

The cluster formation is based on the similarities between the defect data points, with data points in the same cluster being more similar to each other compared to data points in different clusters. The k-means algorithm finds the optimal centroids that minimize the distance between data points within the same cluster and maximize the distance between different clusters.

Once the clustering is complete, any new defect that arises needs to be assigned to one of the existing clusters based on its similarity to the data points in those clusters. This ensures that the new defect is classified within one of the predefined forms identified by the clustering process.

In this way, the k-means algorithm helps in organizing and categorizing defects into distinct clusters, allowing for better understanding, analysis, and management of the software flaws during testing.