Assignment 18 Solutions

1. What is the difference between supervised and unsupervised learning? Give some examples to illustrate your point ?

Ans: The main distinction between the two approaches is the use of labeled datasets. To put it simply, supervised learning uses labeled input and output data, while an unsupervised learning algorithm does not. Unsupervised learning models, in contrast, work on their own to discover the inherent structure of unlabeled data.

In Supervised learning, you train the machine using data which is well “labeled.” Unsupervised learning is a machine learning technique, where you do not need to supervise the model. For example, Baby can identify other dogs based on past supervised learning.

2. Mention a few unsupervised learning applications ?

Ans: The main applications of unsupervised learning include clustering, visualization, dimensionality reduction, finding association rules,recommendation systems, and anomaly detection.

`Clustering:` Unsupervised learning algorithms can be used for clustering, where similar data points are grouped together. This has applications in customer segmentation, image segmentation, document clustering, and anomaly detection.

`Dimensionality Reduction:` Unsupervised learning techniques like Principal Component Analysis (PCA) and t-SNE (t-Distributed Stochastic Neighbor Embedding) can be used to reduce the dimensionality of high-dimensional data while preserving its essential structure. Dimensionality reduction is useful for visualization, feature extraction, and noise reduction.

`Recommendation Systems:` Unsupervised learning can be used to build recommendation systems that suggest items or content based on user preferences. Techniques like collaborative filtering and matrix factorization are commonly used in recommender systems.

`Anomaly Detection:` Unsupervised learning algorithms can detect anomalies or outliers in a dataset by identifying data points that deviate significantly from the normal pattern. This has applications in fraud detection, network intrusion detection, and system health monitoring.

`Natural Language Processing (NLP):` Unsupervised learning is used in NLP for tasks like topic modeling, text clustering, and word embedding. Algorithms such as Latent Dirichlet Allocation (LDA) and Word2Vec learn the underlying structure of textual data without the need for explicit labels.

3. What are the three main types of clustering methods? Briefly describe the characteristics of each ?

Ans: The various types of clustering are:

- **Connectivity-based Clustering (Hierarchical clustering):** Hierarchical Clustering is a method of unsupervised machine learning clustering where it begins with a pre-defined top to bottom hierarchy of clusters. It then proceeds to perform a decomposition of the data objects based on this hierarchy,

hence obtaining the clusters. This method follows two approaches based on the direction of progress, i.e., whether it is the top-down or bottom-up flow of creating clusters.

- **Centroids-based Clustering (Partitioning methods):** Centroid based clustering is considered as one of the most simplest clustering algorithms, yet the most effective way of creating clusters and assigning data points to it. The intuition behind centroid based clustering is that a cluster is characterized and represented by a central vector and data points that are in close proximity to these vectors are assigned to the respective clusters.
- **Density-based Clustering (Model-based methods):** If one looks into the previous two methods that we discussed, one would observe that both hierarchical and centroid based algorithms are dependent on a distance (similarity/proximity) metric. The very definition of a cluster is based on this metric. Density-based clustering methods take density into consideration instead of distances. Clusters are considered as the densest region in a data space, which is separated by regions of lower object density and it is defined as a maximal-set of connected points.
- **Distribution-based Clustering:** Until now, the clustering techniques as we know are based around either proximity (similarity/distance) or composition (density). There is a family of clustering algorithms that take a totally different metric into consideration – probability. Distribution-based clustering creates and groups data points based on their likely hood of belonging to the same probability distribution (Gaussian, Binomial etc.) in the data.
- **Fuzzy Clustering:** The general idea about clustering revolves around assigning data points to mutually exclusive clusters, meaning, a data point always resides uniquely inside a cluster and it cannot belong to more than one cluster. Fuzzy clustering methods change this paradigm by assigning a data-point to multiple clusters with a quantified degree of belongingness metric. The data-points that are in proximity to the center of a cluster, may also belong in the cluster that is at a higher degree than points in the edge of a cluster. The possibility of which an element belongs to a given cluster is measured by membership coefficient that vary from 0 to 1.

4. Explain how the k-means algorithm determines the consistency of clustering ?

Ans: The k-means algorithm does not explicitly determine the consistency of clustering. Instead, it is an iterative algorithm that aims to minimize the within-cluster variance, also known as the sum of squared errors (SSE). The algorithm seeks to find cluster centroids that minimize the distance between data points and their assigned cluster centroid.

Here's an overview of how the k-means algorithm works:

`Initialization:` Initially, the algorithm randomly selects k data points from the dataset as the initial cluster centroids.

`Assignment Step:` In this step, each data point is assigned to the nearest cluster centroid based on a distance metric, commonly the Euclidean distance. The distance is calculated between each data point and the cluster centroids.

`Update Step:` After assigning all data points to clusters, the cluster centroids are recalculated by taking the mean of the data points assigned to each cluster. This step involves computing the new centroids based on the current assignments.

`Iteration:` Steps 2 and 3 are repeated iteratively until convergence. Convergence occurs when the assignment of data points to clusters no longer changes significantly or when a specified number of iterations is reached.

`Minimization of SSE:` The k-means algorithm aims to minimize the within-cluster variance or SSE. SSE is calculated by summing the squared distances between each data point and its assigned cluster centroid. By minimizing SSE, the algorithm seeks to create compact and tightly clustered groups.

It's important to note that the k-means algorithm can get trapped in local optima, meaning it may converge to suboptimal clustering solutions. The choice of initial cluster centroids can affect the final clustering result. To mitigate this issue, the algorithm is often run multiple times with different initializations, and the clustering with the lowest SSE is selected.

Determining the consistency of clustering typically involves evaluating the clustering results using external measures or domain-specific criteria. Measures such as silhouette score, Dunn index, or Rand index can assess the quality and consistency of the clusters. Additionally, visual inspection and domain knowledge can provide insights into the meaningfulness and consistency of the clustering results.

Calculate the Within-Cluster-Sum of Squared Errors (WSS) for different values of k, and choose the k for

5. With a simple illustration, explain the key difference between the k-means and k-medoids algorithms ?

Ans: K-means attempts to minimize the total squared error, while k-medoids minimizes the sum of dissimilarities between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the k -means algorithm, k -medoids chooses datapoints as centers (medoids or exemplars).

6. What is a dendrogram, and how does it work? Explain how to do it ?

Ans: A dendrogram is a type of tree diagram commonly used in hierarchical clustering analysis to visualize the arrangement of data points based on their similarity or dissimilarity. It represents the clustering hierarchy by showing how individual data points or clusters are progressively merged into larger clusters.

Here's an explanation of how to create a dendrogram:

`Calculate Dissimilarity:` First, a dissimilarity or distance matrix is calculated based on the chosen distance metric. The distance metric determines how the dissimilarity between two data points or clusters is measured. Common distance metrics include Euclidean distance, Manhattan distance, and cosine similarity.

`Create Initial Clusters:` Each data point starts as its own individual cluster. If working with pre-clustered data, each existing cluster is treated as an individual entity.

`Merge Clusters:` The clustering algorithm iteratively merges the most similar or closest clusters based on the dissimilarity matrix. The choice of merging criteria depends on the specific clustering algorithm being used. One commonly used criterion is the minimum linkage, which merges clusters based on the minimum dissimilarity between any two points from different clusters.

`Construct the Dendrogram:` As clusters are merged, a dendrogram is constructed. Initially, each data point or cluster is represented as a leaf node. As clusters are merged, branch points are formed, and the tree structure grows vertically.

`Determine Cluster Similarity:` The vertical axis of the dendrogram represents the dissimilarity or similarity between clusters. The height of each branch point corresponds to the dissimilarity or distance between the merged clusters. Clusters that merge earlier in the process are generally more similar or have a lower dissimilarity.

Cut the Dendrogram: The dendrogram can be cut at a desired height to form a specific number of clusters. The height at which the dendrogram is cut determines the granularity of the clusters. Lower cuts produce more clusters, while higher cuts result in fewer, larger clusters.

The resulting dendrogram provides a visual representation of the hierarchical structure and relationship between the data points or clusters. It allows for the identification of similar groups or clusters and helps in determining an appropriate level of granularity for clustering analysis.

Dendrograms can be created using various software packages and programming languages. Popular options include Python libraries like SciPy and scikit-learn, R programming with packages like dendextend and cluster, or specialized data visualization tools like DendroUPGMA and Dendroscope. These tools provide functions and methods to calculate dissimilarity, perform hierarchical clustering, and visualize the resulting dendrogram.

Note:-A dendrogram is a diagram that shows the attribute distances between each pair of sequentially merged classes After each merging, the distances between all pairs of classes are updated. The distances

7. What exactly is SSE? What role does it play in the k-means algorithm ?

Ans: I am going to refer to it as SSE, which stands for Sum of Squared Errors. The regression line is the line made using the function we defined above. To get the SSE we calculate the distance for each of the data points from the regression line then square the it, then we add to the sum.

The SSE is defined as the sum of the squared Euclidean distances of each point to its closest centroid. Since this is a measure of error, the objective of k-means is to try to minimize this value. The purpose of this figure is to show that the initialization of the centroids is an important step.

8. With a step-by-step algorithm, explain the k-means procedure ?

Ans: k-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed apriori. The main idea is to define k centers, one for each cluster.

- Step 1: Choose the number of clusters k.
- Step 2: Select k random points from the data as centroids.
- Step 3: Assign all the points to the closest cluster centroid.
- Step 4: Recompute the centroids of newly formed clusters.
- Step 5: Repeat steps 3 and 4.

9. In the sense of hierarchical clustering, define the terms single link and complete link ?

Ans: In the context of hierarchical clustering, "single link" and "complete link" refer to different criteria or methods for determining the dissimilarity or distance between clusters during the merging process. These criteria define how the distance between two clusters is calculated based on the dissimilarities between their constituent data points.

`Single Link (or Minimum Linkage):` Single link clustering determines the distance between two clusters by considering the minimum dissimilarity or distance between any two points from different clusters. In other words, it measures the similarity between clusters based on the closest pair of points, one from each cluster. The single link criterion tends to create long, elongated clusters and is sensitive to outliers or noise in the data.

`Complete Link (or Maximum Linkage):` Complete link clustering, on the other hand, calculates the distance between two clusters by considering the maximum dissimilarity or distance between any two points from different clusters. It measures the similarity between clusters based on the farthest pair of points, one from each cluster. The complete link criterion tends to create compact, spherical clusters and is more robust to outliers.

The choice between single link and complete link clustering depends on the nature of the data and the desired characteristics of the clusters. Single link clustering is more sensitive to noise and can create elongated clusters, but it can capture elongated structures or clusters with varying densities. Complete link clustering tends to produce more compact clusters, but it can struggle with handling varying cluster densities.

It's important to note that these are just two specific linkage criteria commonly used in hierarchical clustering, and there are other variations as well, such as average link, Ward's method, and centroid linkage. Each criterion has its own strengths and weaknesses, and the choice depends on the specific problem and the underlying characteristics of the data.

`short note:-` In single-link (or single linkage) hierarchical clustering, we merge in each step the two clusters whose two closest members have the smallest distance (or: the two clusters with the smallest minimum pairwise distance). Complete-link clustering can also be described using the concept of clique.

10. How does the apriori concept aid in the reduction of measurement overhead in a business basket analysis? Give an example to demonstrate your point ?

Ans: The Apriori concept is a key algorithm used in market basket analysis, which aims to identify relationships and associations among items frequently purchased together by customers. It helps in reducing measurement overhead by employing a technique called "pruning," which avoids unnecessary computations and focuses on relevant itemsets.

The concept of Apriori aids in reducing measurement overhead in the following ways:

`Support-Based Pruning:` The Apriori algorithm uses a minimum support threshold to identify frequent itemsets. The support of an itemset is the proportion of transactions in the dataset that contain that itemset. By setting a minimum support threshold, the algorithm prunes infrequent itemsets from further consideration. This pruning step reduces measurement overhead by eliminating itemsets that are unlikely to be significant for analysis.

`Downward Closure Property:` The Apriori algorithm leverages the downward closure property, which states that any subset of a frequent itemset must also be frequent. This property allows the algorithm to avoid generating and counting all possible combinations of items, focusing only on the itemsets that meet the minimum support threshold. By exploiting this property, Apriori prunes itemsets that cannot be frequent based on the infrequent subsets they contain.

`Apriori Principle:` The Apriori principle states that if an itemset is infrequent, then its supersets will also be infrequent. This principle allows the algorithm to stop generating and counting itemsets once no more frequent supersets can be formed. This further reduces the measurement overhead by avoiding unnecessary computations for itemsets that cannot be frequent based on the already analyzed subsets.

Here's an example to illustrate how the Apriori concept reduces measurement overhead:

Suppose we have a dataset of customer transactions in a supermarket. The Apriori algorithm aims to find associations between items frequently purchased together. Let's consider a minimum support threshold of 0.1 (meaning an itemset must appear in at least 10% of transactions to be considered frequent).

1. Initially, the algorithm identifies frequent individual items such as milk, bread, and eggs, based on their support.
2. The algorithm then generates and counts itemsets of size 2 (pairs of items) containing these frequent items. For example, it may analyze the combination {milk, bread} and {milk, eggs}.
3. Using support-based pruning and the Apriori principle, the algorithm avoids generating and counting itemsets that contain infrequent subsets. For instance, if {milk, bread} is infrequent, the algorithm does not explore larger itemsets containing {milk, bread}.

By pruning infrequent itemsets and avoiding unnecessary computations, the Apriori concept reduces measurement overhead, focusing only on relevant and potentially significant itemsets. This helps in efficient market basket analysis and the identification of meaningful associations among items purchased together by customers.

`short note :-` Apriori algorithm assumes that any subset of a frequent itemset must be frequent. Its the algorithm behind Market Basket Analysis. So, according to the principle of Apriori, if {Grapes, Apple, Mango} is frequent, then {Grapes,Mango} must also be frequent. Here is a dataset consisting of six transactions."

In [ ]:

```
1
```