VCET TECHZETTE विसीईटी ज्ञानपत्र
ISSN: 2584-0886 (Online)

Unsupervised Algorithm : K - means

Sonam Dahanukar

It is a clustering algorithm and as the name suggests it comes in the domain of unsupervised learning algorithms. An unsupervised algorithm is used when the data doesn't come in a labeled with predefined categories. Among all the algorithms K-means offers an efficient means to partition data in different groups.

Need of developing a Clustering algorithms:

Imagine a scenario where we are having a very large dataset with no clear labels or classification. So then it becomes a very difficult task to gain some insights from such kind of data. This is the moment where a clustering algorithm like K-means step in. The algorithm creates groups/clusters on the basis of the similarities present in the given data. This cluster helps us to gain insights from the given unlabeled dataset and help us in decision making.

K-means concept:

The core concept of the K-means algorithm is quite simple yet powerful. The 'K' in K-means algorithm stands for the number of clusters we aim to create in the dataset.

Working of K- means:

1. Initialization: Start by taking 'K' number of data points fromm the dataset as initial cluster centroid.

2. Assignment Step: Assign each data point to the nearest centroid based on the Euclidean distance.

3. Update Step: After assigning each data point to a nearest centroid again calculate the cluster's centroid based on the mean of the data points assigned to the cluster.

4. Iteration: Repeat assignment and update step until we meet our convergence criteria. The minimal criteria of the cluster is that there should be minimal change in the cluster's centroid.

VCET TECHZETTE विसीईटी ज्ञानपत्र
ISSN: 2584-0886 (Online)

Practical implementation of K-means algorithm:

Suppose we have the following dataset with two-dimensional points:

(2, 10), (2, 5), (8, 4), (5, 8), (7, 5), (6, 4), (1, 2), (4, 9)

And let's say we want to partition this dataset into two clusters.

Step 1: Initialization

We randomly select two points as initial centroids. Let's say we choose (2, 10) and (5, 8) as our initial centroids.

Step 2: Assignment

We calculate the distance of each point to each centroid and assign each point to the closest centroid. Using Euclidean distance, the assignments would be:

Cluster 1 (Centroid: (2, 10)):

(2, 10), (2, 5), (1, 2)

Cluster 2 (Centroid: (5, 8)):

(8, 4), (7, 5), (6, 4), (4, 9)

Step 3: Update

We recalculate the centroids based on the mean of the points assigned to each cluster. The new centroids would be:

New Centroid for Cluster 1: (1.67, 5.67)

New Centroid for Cluster 2: (6.25, 5.5)

Step 4: Iteration

We repeat steps 2 and 3 until convergence. Let's say after another iteration, the assignments remain the same, and the centroids don't change significantly.

Finalization

ISSN: 2584-0886 (Online)

The final clusters would be:

Cluster 1 (Centroid: (1.67, 5.67)):

(2, 10), (2, 5), (1, 2)

Cluster 2 (Centroid: (6.25, 5.5)):

(8, 4), (7, 5), (6, 4), (4, 9)

This is a simplified example of how K-means clustering works. In practice, the algorithm iterates until convergence based on a predefined criterion, such as no significant change in cluster centroids.

Inorder to have a practical implementation of K-means algorithm we can use this python codes snippet:

```
from sklearn.cluster import KMeans

# Initialize K-means with the desired number of clusters (K)

kmeans = KMeans(n_clusters=K)

# Fit the model to the data

kmeans.fit(data)

# Obtain cluster centroids and labels

centroids = kmeans.cluster_centers_

labels = kmeans.labels_
```

Key consideration while working with the K-means algorithm:

K-means is a very straight forward algorithm but there are several key aspects we need to consider while working with it.

1. Choosing the right or optimal amount of clusters is necessary. Several techniques such as elbow method or silhouette analysis can be used for choosing the right amount of clusters.

ISSN: 2584-0886 (Online)

2. Selecting the initial amount of centroid is a very sensitive process because the initial selection of the centroid can impact the final clustering result. Techniques like K-means++ can help us to mitigate such kind of sensitivity.

3. Depending on the initial centroid the algorithm can converge into an local optima so for that reason we should run the algorithm multiple times with different initialization inorder to mitigate this issue.

Application of K-means algorithm:

1. Customer segmentation: K-means helps the companies to classify their customers in different groups. It can classify the customers based on their preference or shopping pattern.

2. Image compression: By clustering similar pixels together and representing them with the cluster's centroid it is possible to reduce the amount of data required to store images without any significant loss of image quality.

3. Anomaly detection: By clustering normal data points together anomalies or outliers can be identified as datapoints fall outside the cluster. It can be used in various domains including finance, cybersecurity and manufacturing.

4. Network traffic analysis: By clustering network data, pattern inductive or malicious activities can be detected and can be solved. It is very useful in cybersecurity in order to find out potential threats.

These are a few examples which showcase the versatility and practicality of K-means algorithms across different domains.

In conclusion, the K-means algorithm serves as a very strong algorithm inorder to gain insights from unsupervised datasets. It is a very robust algorithm for clustering with its simple principles to uncover the patterns from the given dataset. Because of this it has become one of the most useful tools of data scientists and data analysts. As we discover the depth of data science K-means remains a beacon in unveiling the mysterious patterns from the dataset.