Project Report on

K-MEANS CLUSTERING

- Done By

L VIKRAM SIMHA REDDY

Table of Content

1. Project Definition ……..3

2. Literature Survey ……...4

3. K-means Algorithm ……...5

4. Elbow Method ……...6

5. Confusion Matrix ……..7

6. Implementation ……8

7. Python Code for Implementation ……..9

8. Evaluation & Result ……10

9. Conclusion ……...11

1. Project Definition

Unsupervised learning refers to the process of learning in machines in which an algorithm is trained using unlabeled data; therefore, the outputs are undefined. The unsupervised learning approach is one in which a model discovers or identifies patterns and relationships or structures within data without the guide of any known target variable.

Clustering is one of the important techniques in unsupervised learning. It regroups a set of objects in such a manner that objects belonging to the same group are similar to one another than to those belonging in another group. The similarity is according to some metric that could be based on Euclidean distance, cosine similarity, or other measures depending on the nature of the data.

Clustering is one of the most common techniques in unsupervised learning applied to understand or discover hidden patterns and groupings in data. Using clustering, we will be able to uncover any natural groupings in data that could not be visually identified.

K-means clustering is one of the most straightforward and widely applied clustering algorithms, where the goal is to divide a dataset into K clusters such that every data point is assigned to the cluster with the closest mean. While the K-means clustering algorithm has desirable features of being simple and efficient for large data sets, it does require the number of clusters, K, to be specified in advance, and it can also be somewhat sensitive to the initial placement of centroids.

2. Literature Survey

There are several clustering approaches. These are partitioning (eg. K- means, kmedoids), hierarchical (eg. DIANA, AGNES, BIRCH), viscosity- grounded (eg. DBSACN, OPTICS), grid- grounded (eg. STING, crowd), model grounded (eg. EM, COBWEB), frequent pattern- grounded (eg. p- Cluster), stoner- quided or constraint- grounded (eg. COD), and link-grounded (eg. SimRank, LinkClus) clustering approaches. utmost of these are explained and some of them originally proposed in the book of Kaufman and Rousseeuw in 1990 which are partitioning,

The most frequent system which is applied to documents is hierarchical clustering system. In 1988, Willett applied agglomerative clustering styles to documents by changing the computation system of distance between clusters. These algorithms have several problems with clusters that chancing stopping point is veritably delicate and they run too sluggishly for thousands of documents. Hierarchical clustering algorithms are applied to documents for several times by Zhao and Karypis and in 2005 they tried to ameliorate agglomerative clustering algorithm by adding constrains.

K- means and its variants, which are partitioning clustering algorithms that produce a non hierarchical clustering conforming of k clusters, are applied to documents. These algorithms are more effective and scalable, and their complexity is direct to the number of documents. A Another disadvantage of k- means is that the wrong estimation of the value of k leads to worse delicacy. also, k- means can stick on a original outside due to aimlessly chosen original centroids.

3. K-Means Algorithm

K-means clustering algorithm is known to be efficient in clustering large data sets. This algorithm is one of the simplest and the best known unsupervised learning algorithms. It solves the well-known clustering problem. The K-Means algorithm aims to partition a set of objects, based on their attributes/features, into k clusters, where k is a pre-defined constant. The algorithm defines k centroids, one for each cluster. The centroid of a cluster is formed in such a way that it be closely related, in terms of similarity, to all objects in that cluster.

Technically, what k-means is interested in, is the variance. It minimizes the sum of variance by putting each object to the cluster such that the variance is minimized. Coincidentally, sum of squared deviations, one objects contribution to the total variance, over all dimensions is exactly the definition of squared euclidean distance.

In Mahout implementation of k-mean, Each object will be represented as vector in space. First, k points will be selected by the algorithm randomly and treated as centers, every object closest to each center are clustered.

K-means clustering job requires input vector directory, output clusters directory, distance measure, maximum number of iterations to be performed and an integer value representing the number of clusters the input data is to be divided into.

Alpaydin symbolizes this algorithm like below where m is sequence of means, x' is sequence of samples, and b is sequence of estimated labels

Initialize $\boldsymbol{m}_i$, $i = 1,\ldots,k$, for example, to k random $\boldsymbol{x}^t$

Repeat

For all $\boldsymbol{x}^t \in X$

$b_i^t \leftarrow 1$ if $\| \boldsymbol{x}^t - \boldsymbol{m}_i \| = \min_j \| \boldsymbol{x}^t - \boldsymbol{m}_j \|$

$b_i^t \leftarrow 0$ otherwise

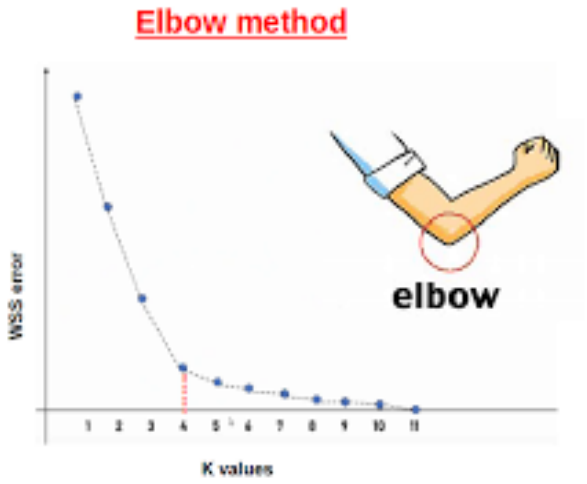
For all $\boldsymbol{m}_i$, $i = 1,\ldots,k$

$\boldsymbol{m}_i \leftarrow \sum_t b_i^t \, x^t \; / \sum_t b_i^t$

Until $\boldsymbol{m}_i$ converge

4. Elbow Method

The elbow method is based on the observation that increasing the number of clusters can help to reduce the sum of within-cluster variance of each cluster. This is because having more clusters allows one to capture finer groups of data objects that are more similar to each other. To define the optimal clusters, Firstly, we use the clustering algorithm for various values of k. This is done by ranging k from 1 to 10 clusters. Then we calculate the total intra-cluster sum of square. Then, we proceed to plot intra-cluster sum of square based on the number of clusters. The plot denotes the approximate number of clusters required in our model. The optimum clusters can be found from the graph where there is a bend in the graph.



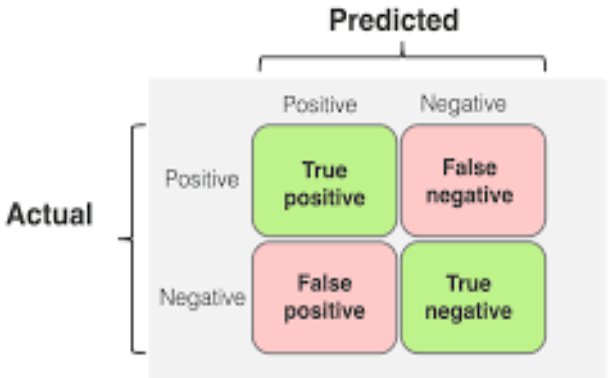
5. Confusion Matrix

The confusion matrix is a widely used tool for evaluating the performance of classification models in supervised learning. However, its application to k-means clustering, an unsupervised learning algorithm, requires some adaptation. In supervised learning, a confusion matrix is constructed using true labels and predicted labels. In contrast, k-means clustering deals with unlabeled data, so true labels are not inherently available. To use a confusion matrix for k-means clustering, we must first map the clusters to known true labels, if such labels are available for evaluation.

K-means clustering partitions a dataset into KKK clusters, where each data point belongs to the cluster with the nearest centroid. After clustering, if we have the true labels of the data points, we can assign each cluster to the true label that is most frequent within that cluster. This process involves examining the cluster assignments and comparing them to the true labels to find the most common label in each cluster. This forms the basis for constructing a confusion matrix in the context of clustering.

To construct the confusion matrix, we define the rows to represent the true labels and the columns to represent the cluster assignments (predicted labels). Each cell in the matrix indicates the number of data points that belong to the corresponding true label and predicted cluster.

The resulting matrix allows us to evaluate the clustering performance using various metrics such as accuracy, precision, recall, and the F1-score. Accuracy measures the proportion of correctly clustered points, while precision and recall provide insights into the clustering quality for each true label. The F1-score, the harmonic mean of precision and recall, offers a balanced measure of clustering performance.



6. Implementation

K-means clustering is a popular unsupervised learning algorithm used to partition a dataset into K distinct, non-overlapping clusters. The algorithm aims to minimize the variance within each cluster. Here's a concise implementation and explanation of the k-means clustering algorithm:

Steps of K-means Clustering

1. **Initialization**: Choose K initial centroids randomly from the dataset.
2. **Assignment**: Assign each data point to the nearest centroid based on the Euclidean distance.
3. **Update**: Recalculate the centroids as the mean of all data points assigned to each cluster.
4. **Repeat**: Repeat the assignment and update steps until the centroids no longer change significantly or a maximum number of iterations is reached.

7. Python Code for Implementation:

```
import numpy as np
from sklearn.datasets import make_blobs
import matplotlib.pyplot as plt
# Generate sample data
X, y = make_blobs(n_samples=300, centers=4, random_state=42)
# K-means algorithm
def kmeans(X, K, max_iters=100):
    # Initialize centroids
    centroids = X[np.random.choice(X.shape[0], K, replace=False)]
    for _ in range(max_iters):
        # Assign clusters
        distances = np.linalg.norm(X[:, np.newaxis] - centroids, axis=2)
        labels = np.argmin(distances, axis=1)
        # Update centroids
        new_centroids = np.array([X[labels == k].mean(axis=0) for k in range(K)])
        # Check for convergence
        if np.all(centroids == new_centroids):
            break
        centroids = new_centroids
    return labels, centroids
# Apply k-means
K = 4
labels, centroids = kmeans(X, K)
# Plot the results
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=200, alpha=0.75)
plt.show()
```

8. Evaluation & Result

Evaluating k-means clustering involves quantitative metrics to determine how well the algorithm has grouped the data. Common metrics include:

1. **Within-Cluster Sum of Squares (WCSS)**: This measures the total variance within each cluster. A lower WCSS indicates tighter clusters.

$$WCSS = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

where C_k is the k-th cluster, x_i is a data point in cluster k, and μ_k is the centroid of cluster k.

2. **Silhouette Score**: This assesses how similar a data point is to its own cluster compared to other clusters. It ranges from -1 to 1, with higher values indicating better-defined clusters.

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

where a(i) is the average distance between a point and other points in the same cluster, and b(i) is the minimum average distance from the point to points in a different cluster.

3. **Elbow Method**: This involves plotting the WCSS against the number of clusters K. The optimal K is typically at the "elbow" point, where the rate of decrease sharply slows.

The result of k-means clustering is a set of K clusters, each represented by a centroid, with every data point assigned to the nearest cluster. Evaluating the quality and effectiveness of these clusters involves several metrics and techniques.

After running the k-means algorithm, the primary outputs are:

Cluster Assignments: Each data point is assigned a label corresponding to the nearest centroid.

Centroids: These are the mean positions of all the points in each cluster and represent the center of each cluster.

The visual representation often includes a scatter plot where points are colored based on their cluster assignments, and centroids are marked distinctly, often in a different color or with a larger marker.

9. Conclusion

K-means clustering is a foundational technique in unsupervised learning used to partition a dataset into distinct clusters. The algorithm works iteratively to minimize the variance within clusters by assigning each data point to the nearest centroid and updating centroids to be the mean of the assigned points. This iterative process continues until the centroids stabilize or a maximum number of iterations is reached.

The result of k-means clustering is a set of K clusters, each represented by a centroid, and each data point is assigned to the closest cluster. These clusters can be visualized using scatter plots, where points are colored based on their cluster assignments, and centroids are marked distinctly. This visual representation helps in understanding the clustering structure and evaluating its effectiveness.

In addition to quantitative metrics, qualitative analysis through visualization is crucial. Scatter plots and domain-specific knowledge provide insights into the compactness and separation of clusters, ensuring that the results are meaningful and interpretable.

In conclusion, k-means clustering is a powerful tool for uncovering hidden patterns in data by grouping similar data points into clusters. Its effectiveness is determined through a combination of quantitative metrics and qualitative analysis, making it a versatile method for various applications in data analysis, customer segmentation, image recognition, and more. By understanding its principles and evaluation methods, one can effectively utilize k-means clustering to derive meaningful insights from complex datasets.