INTRODUCTION TO FUZZY CLUSTERING – SOFT CLUSTERING

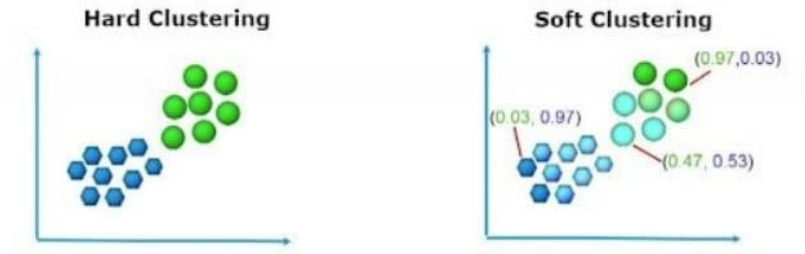
Author: Nguyen Thanh Luan

1. What is clustering?

Clustering is an unsupervised machine learning algorithm aimed at dividing a dataset into groups or clusters, where the points within the same cluster have similar characteristics and are distinct from points in other clusters. Essentially, clustering algorithms are categorized into two main types: soft clustering and hard clustering.

+ Hard clustering: The algorithm assigns each data point to a single cluster, such as the K-means algorithm.

+ Soft clustering: Instead of assigning each point to one cluster, the algorithm calculates the probability of each data point belonging to different clusters. This means that a point can belong to multiple clusters with varying probabilities. C-means is the algorithm introduced in this article.



a. K-means algorithm

As introduced, K-means is a popular hard clustering algorithm. The algorithm works with the goal of dividing data into k clusters so that data points within the same cluster have similar characteristics. The steps of the algorithm are as follows:

Step 1: Choose the number of clusters (k) to create.

Step 2: Initialize k initial points as the centers of the clusters.

Step 3: Repeat the following steps until the stopping criterion is met:

+ Assign each data point to the cluster with the closest center.

+ Update the center of each cluster by calculating the average of all data points assigned to that cluster

+ Check if the assignment of data points to clusters has changed compared to the previous iteration. If there is no significant change or no change at all, stop the algorithm

Step 4: The algorithm terminates when the stopping criterion is met or the maximum number of iterations is reached.

Limitations of the K-means algorithm: It is sensitive to the initial points, not flexible regarding cluster shapes (K-means tends to assume clusters are convex and of similar sizes, which is not generalizable to real-world data), accuracy is limited by the stopping rule, it is heavily affected by noise, and is not suitable for non-linear data.

b. C-means algorithm

Unlike K-means, C-means (Fuzzy C-means, FCM) allows a data point to belong to multiple clusters with varying degrees of membership (fuzziness). This approach enables the exploration of more complex data clusters in a more flexible manner. The steps of the algorithm are as follows:

Step 1: Choose k initial points as cluster centers and set the fuzziness degree m (m = 2).

Step 2: Initialize the weight matrix using the partition matrix mechanism to assign membership values.

Step 3: Repeat the following steps until the stopping criterion is met:

+ Calculate the center of each cluster by using the weight matrix **W**

+ Calculate the degree of membership of each data point to each cluster using the weight matrix and the distance from the point to the cluster center, with the fuzziness degree **m** defined.

+ Update the weight matrix **W** by using the degree of membership of each point to each cluster, and apply the fuzzy formula to calculate the new weights.

Step 4: The algorithm terminates when the stopping criterion is met or the maximum number of iterations is reached.

Limitations of the C-means algorithm*:* It is sensitive to changes in the number of clusters (although the number of clusters can be adjusted by tweaking the fuzziness degree, choosing an inappropriate initial number of clusters may lead to suboptimal clustering), complex computations, poor differentiation between overlapping clusters, difficulty in determining the fuzziness parameter **m**. Like K-means, FCM is heavily influenced by outliers and does not handle non-normally distributed data well (because FCM assumes a normal distribution).

2. Compare two algorithms K-means and C-means

Criteria	K-means	C-means
Cluster type	Hard clustering	Soft clustering
Accuracy	Prone to high accuracy	Self improvement
Initialization	More sensitive	More stable

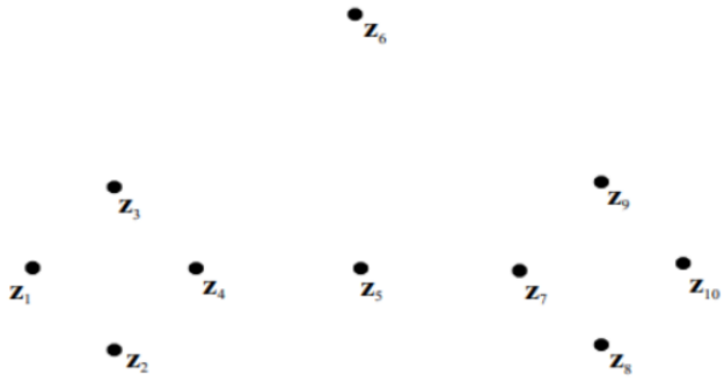
Cluster Shape	Simple and uniform in size (not flexible)	Can accommodate non-convex shapes and varying sizes (flexible)
Noise and outlier	Sensitive	Better
Computation Cost	Faster	Slower
Complexity	Simple	More complex
Priority	Tends to focus on clusters with more data points	Solves the issue of prioritizing clusters with more data points

Although both algorithms differ, C-means is essentially considered a variant of K-means with improvements based on the fuzzy parameter m in the calculation formula. Therefore, the loss function for both algorithms can be reduced to the following common formula:

$$J(Y, M) = \sum_{i=1}^{N} \sum_{j=1}^{K} w_{ij}^{m} \left\| x_i - c_j \right\|^2$$

Note: if the parameter m = 1, the C-means algorithm will return to K-means

3. How does C-means reduce the 'priority' in K-means?



+ Kmeans:

c_0	1	1	1	1	1	1	0	0	0	0
c_1	0	0	0	0	0	0	1	1	1	1

+ Cmeans:

c_0	1	1	1	0.8	0.5	0.5	0.2	0	0	0
c_1	0	0	0	0.2	0.5	0.5	0.8	1	1	1

The correct answer is the ***fuzziness*** parameter. In the case above, with data points 5 and 6, which are located in between and have equal distances to the centers of the clusters, K-means will create a bias by assigning these two points to the cluster with more points (let's assume cluster 0). On the other hand, C-means will assess that these two points have equal membership in both clusters.