Hello Everyone,

I hope you all are fine and doing well.

Fuzzy C-Means (FCM) Clustering: Mathematical Formulation with Example

模糊 C 均值 (FCM) 聚類：數學公式與範例

Fuzzy C-Means (FCM) is a clustering algorithm that allows each data point to belong to multiple clusters with varying degrees of membership. The key idea behind FCM is to minimize an objective function that measures both the distance between data points and cluster centroids and the degree of membership of each data point to each cluster.

模糊 C 均值 (FCM) 是一種聚類演算法，允許每個資料點屬於具有不同隸屬程度的多個聚類。 FCM 背後的關鍵思想是最小化目標函數，該函數既測量資料點與簇質心之間的距離，也測量每個資料點與每個簇的隸屬度。

Mathematical Formulation of Fuzzy C-Means (FCM)

1. Definitions and Setup

Let:

1) $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n data points, where each data point x_i is a vector in R^m (i.e., each point has m features).

2) $C = \{c_1, c_2, \ldots, c_k\}$ be the set of cluster centers (centroids), where k is the number of clusters.

模糊 C 均值 (FCM) 的數學公式

1. 定義和設置

讓：

1) $X = \{x_1, x_2, \ldots, x_n\}$ 是一組 n 資料點，其中每個資料點 x_i是 R^m 中的向量（即每個點都有 m 特徵）。

2) $C = \{c_1, c_2, \ldots, c_k\}$ 是聚類中心（質心）的集合，其中 k 是聚類數量。

3) $U = [u_{ij}]$ be the fuzzy membership matrix, where u_{ij} represents the degree of membership of data point x_i in cluster j. These values satisfy:

$$0 \le u_{ij} \le 1$$

And

$$\sum_{j=1}^{k} u_{ij} = 1 \; for\ each\ i$$

meaning each data point has membership values across all clusters summing to 1.

3) U=[u_{ij}] 是模糊隸屬度矩陣，其中u_{ij}表示資料點x_i在簇 j 中的隸屬度。
這些值滿足： 0≤u_{ij}≤1 和 $\sum_{j=1}^{k} u_{ij} = 1 \; for\ each\ i$
這意味著每個資料點在所有群集中的成員資格值總和為 1。

2. Objective Function

The objective of the FCM algorithm is to minimize the following **cost function** (or **objective function**):

2. 目標函數

FCM演算法的目標是最小化以下**成本函數**（或**目標函數**）：

$$J_m(U,C) = \sum_{i=1}^{n}\sum_{j=1}^{k} u_{ij}^{m}\left\|x_i - c_j\right\|^2$$

where:

1) u_{ij} is the membership value of data point x_i in cluster c_j.

2) m is the **fuzzification parameter** (usually $m > 1$, typically $m = 2$).

在哪裡：

1) u_{ij}是資料點x_i在叢集c_j中的隸屬度值。

2) m是**模糊化參數**（通常m>1，通常m=2）

3) $\|x_i - c_j\|^2$ is the squared Euclidean distance between data point x_i and cluster center c_j:

$$\|x_i - c_j\|^2 = (x_i - c_j)^T (x_i - c_j)$$

The objective function aims to minimize the weighted sum of squared distances between each data point and the cluster centers, weighted by the degree of membership. The **fuzzification parameter** m controls the degree of fuzziness of the clustering. A larger m makes the membership values more evenly distributed, while a smaller m leads to a more crisp (less fuzzy) clustering.

3) $\|x_i - c_j\|^2$ 是資料點x_i與聚類中心c_j之間的歐氏距離平方：

$$\|x_i - c_j\|^2 = (x_i - c_j)^T (x_i - c_j)$$

目標函數旨在最小化每個資料點與聚類中心之間的加權平方和（按隸屬度加權）。**模糊化參數** m 控制聚類的模糊程度。較大的 m 使隸屬度值分佈更均勻，而較小的 m 則導致更清晰（不太模糊）的聚集。

3. Updating the Cluster Centers

The cluster centers c_j are updated iteratively based on the membership values. The updated cluster center c_j is calculated as the **weighted average** of all data points, where the weights are the membership values raised to the power m:

$$c_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m}$$

This formula ensures that the new cluster center c_j is closer to the data points with higher membership in cluster j.

3. 更新集群中心 聚類中心 c_j 根據隸屬值迭代更新。更新後的聚類中心 c_j 計算為所有資料點的加權平均值，其中權重是會員值的 m 次方：

$$c_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m}$$

此公式確保新的聚類中心c_j更接近聚類 j 中具有較高隸屬度的資料點。

4. Updating the Membership Matrix

The membership values u_{ij} are updated based on the distance between each data point x_i and the cluster centers. The membership value of point x_i in cluster j is computed as:

4. 更新會員矩陣 隸屬度值u_{ij}根據每個資料點x_i與聚類中心之間的距離進行更新。簇 j 中點 x_i的隸屬度值計算如下：

$$u_{ij} = \frac{1}{\sum_{l=1}^{k} \left(\frac{\|x_i - c_j\|}{\|x_i - c_l\|} \right)^{\frac{2}{m-1}}}$$

where:

1) $\|x_i - c_j\|$ is the Euclidean distance between data point x_i and cluster center c_j.

2) $\|x_i - c_l\|$ is the Euclidean distance between data point x_i and any other cluster center c_l.

The formula ensures that data points are more likely to belong to clusters whose centers are closer to the point, with the degree of membership depending on the relative distances to all cluster centers.

在哪裡：
1) $||x_i - c_j||$ 是資料點x_i和聚類中心c_j之間的歐氏距離。
2) $||x_i - c_l||$ 是資料點x_i與任何其他聚類中心c_l之間的歐氏距離。
此公式確保資料點更有可能屬於中心距離該點較近的簇，其隸屬程度取決於與所有簇中心的相對距離。

5. Stopping Condition

The algorithm iterates between updating the membership matrix and the cluster centers until the objective function $J_m(U, C)$ converges, i.e., until the changes in the membership values and cluster centers fall below a predefined threshold. Alternatively, the algorithm can stop after a fixed number of iterations.

5. 停止條件 此演算法在更新隸屬矩陣和聚類中心之間進行迭代，直到目標函數 $J_m(U,C)$ 收斂，即直到隸屬值和聚類中心的變化低於預先定義的閾值。或者，演算法可以在固定次數的迭代後停止。

Example of Fuzzy C-Means (FCM)

模糊 C 均值 (FCM) 範例

Let's go through a small example to demonstrate how FCM works.

Data Points

Suppose we have the following 2D data points:

$$x_1 = (1,2)$$
$$x_2 = (2,3)$$
$$x_3 = (5,8)$$
$$x_4 = (8,8)$$

We want to cluster these points into $k = 2$ clusters.

讓我們透過一個小例子來示範 FCM 的工作原理。

資料點 假設我們有以下二維資料點：

$$x_1 = (1,2)$$
$$x_2 = (2,3)$$
$$x_3 = (5,8)$$
$$x_4 = (8,8)$$

我們希望將這些點聚類成 k=2 個簇。

Step 1: Initialize the Membership Matrix and Cluster Centers

Let's initialize the fuzzy membership matrix U randomly (in practice, this is done more systematically):

$$U = \begin{bmatrix} 0.6 & 0.4 \\ 0.7 & 0.3 \\ 0.2 & 0.8 \\ 0.1 & 0.9 \end{bmatrix}$$

Let's also initialize the cluster centers randomly:

$$c_1 = (2,2)$$
$$c_2 = (6,6)$$

Step 2: Update the Cluster Centers

Using the formula for updating cluster centers:

$$c1 = \frac{\big(0.6 * (1,2)\big) + \big(0.7 * (2,3)\big) + \big(0.2 * (5,8)\big) + \big(0.1 * (8,8)\big)}{0.6 + 0.7 + 0.2 + 0.1}$$

$$= \frac{(0.6,1.2) + (1.4,2.1) + \big((1.0,1.6)\big) + \big((0.8,0.8)\big)}{1.6}$$

$$= \frac{(3.8,5.7)}{1.6} = (2.375,3.563)$$

Next, we compute c_2:

$$c1 = \frac{\big(0.4 * (1,2)\big) + \big(0.3 * (2,3)\big) + \big(0.8 * (5,8)\big) + \big(0.9 * (8,8)\big)}{0.4 + 0.3 + 0.8 + 0.9}$$

$$= \frac{(0.4,0.8) + (0.6,0.9) + (4.0,6.4) + (7.2,7.2)}{2.4}$$

$$= \frac{(12.2,15.3)}{2.4} = (5.083,6.375)$$

Step 3: Update the Membership Matrix

Now, we update the membership values for each point based on the distances to the new cluster centers. Let's calculate the distance of each point to both cluster centers and update the memberships.

For simplicity, assume we do the calculations for the first point $x_1 = (1,2)$:

1)The distance from x_1 to $c_1 = (2.375,3.563)$ is:

$$\|x_1 - c_1\| = \left((1 - 2.375)^2 + (2 - 3.563)^2\right)^{\frac{1}{2}}$$
$$= (1.890625 + 2.462369)^{\frac{1}{2}} = (4.353)^{\frac{1}{2}} = 2.09$$

2)The distance from x_1 to $c_2 = (5.083,6.375)$ is:

$$\|x_1 - c_2\| = \left((1 - 5.083)^2 + (2 - 6.375)^2\right)^{\frac{1}{2}}$$
$$= (16.640689 + 19.616625)^{\frac{1}{2}} = (36.257)^{\frac{1}{2}} = 6.02$$

Now, we compute the updated membership values for x_1 using the formula:

$$u_{11} = \frac{1}{\left(\frac{2.09}{6.02}\right)^{\frac{2}{m-1}} + \left(\frac{6.02}{2.09}\right)^{\frac{2}{m-1}}}$$

We continue similarly for the other data points, iterating the process to update both cluster centers and membership matrix.

Step 4: Iterate Until Convergence

Repeat the steps until the cluster centers and membership values converge to stable values. This means that the changes in $J_m(U, C)$, the objective function, are smaller than a specified threshold, or a maximum number of iterations is reached.

Conclusion

Fuzzy C-Means clustering is a powerful tool that assigns membership values to data points, allowing them to belong to multiple clusters with varying degrees of membership. The algorithm minimizes a cost function based on the distance between points and their cluster centers, and iteratively updates the membership matrix and cluster centers until convergence. This allows fuzzy clustering to handle situations where the boundaries between clusters are not well-defined, making it suitable for a wide range of real-world applications.