Đại học quốc gia TP.HCM
Trường đại học công nghệ thông tin

Ngành Khoa Học Máy Tính

# FUZZY CLUSTERING

*Học viên:*
Hà Kiệt Hùng - 220101031

*Giảng viên:*

# FUZZY CLUSTERING

May 15, 2024

## Contents

# 1 Introduction

Clustering is a fundamental task in data analysis and machine learning, aiming to partition a dataset into groups or clusters based on the similarity of data points. Traditional clustering algorithms, such as K-means and hierarchical clustering, assign each data point to exactly one cluster, resulting in a hard partitioning of the data. However, in many real-world scenarios, data points may exhibit degrees of membership to multiple clusters, leading to a more nuanced representation of the underlying structure of the data.

Fuzzy clustering, also known as soft clustering, addresses this limitation by allowing data points to belong to multiple clusters simultaneously with varying degrees of membership. This flexibility enables fuzzy clustering algorithms to capture complex patterns in the data that may not be well-suited for traditional hard clustering methods.

At the heart of fuzzy clustering is the concept of fuzzy sets, introduced by Lotfi A. Zadeh in the 1960s. Fuzzy sets generalize classical set theory by allowing elements to have degrees of membership ranging between 0 and 1, rather than strictly belonging or not belonging to a set. In the context of clustering, fuzzy sets are employed to represent the degree to which a data point belongs to each cluster.

The most widely used fuzzy clustering algorithm is the Fuzzy C-means (FCM) algorithm, proposed by Dunn in 1973 and refined by Bezdek in 1981. FCM iteratively assigns data points to clusters based on their distances to cluster centroids, updating the membership degrees at each iteration until convergence. Other popular fuzzy clustering algorithms include the Gustafson-Kessel (GK) algorithm, Possibilistic C-means (PCM), and Fuzzy Subtractive Clustering (FSC), each with its own advantages and applications.

Fuzzy clustering finds applications in various domains, including pattern recognition, image processing, bioinformatics, and data mining. By providing a flexible framework for modeling complex data relationships, fuzzy clustering offers valuable insights into the underlying structure of datasets and supports decision-making in diverse fields.

In this repository, we explore different fuzzy clustering algorithms, their theoretical foundations, implementations, and practical applications. Through detailed explanations, code examples, and demonstrations, we aim to provide a comprehensive understanding of fuzzy clustering and its potential for analyzing and interpreting complex datasets.

# 2 Fundamentals of Fuzzy Clustering

## 2.1 Overview of traditional clustering vs. fuzzy clustering

### 2.1.1 Traditional Clustering

1. Traditional clustering, as know as hard clustering, techniques attempt to segment data by grouping related attributes in uniquely defined clusters. Each data point in the sample space is assigned to only one cluster. In partitioning the data only cluster centers are moved and none of the data points are moved. K-means is one of the hard clustering method. K-means is the simplest and most widely used algorithm in many areas. This algorithm similarly measure is based on Euclidean distance and works only for datasets that consist of numerical attributes.

2. K-means algorithm is as following:

   (a) Input: k: the number of clusters

   (b) Method:

Step 1: Choose k numbers of clusters to be determined.

Step 2: Choose $C_k$ centroids randomly as the initial centers of the clusters.

Step 3: Repeat

   3.1 Assign each object to their closest cluster center using Euclidean distance

   3.2 Compute new cluster center by calculating mean points.

Step 4 Until

   4.1 No change in cluster center OR No object changes its clusters.

3. Method to find the distance is to calculate to sum of the squared difference as follows and it is known as the Euclidean distance (1).

$$d_k = \sum_{j=1}^{n} ||X_j^k - C_j^i||^2 \tag{1}$$

where,

$$
\begin{aligned}
d_k &: \text{distance of the } k^{th} datapoint \\
n &: \text{number of attributes in a cluster} \\
X_j^k &: \text{jth value of the } k^{th} datapoint \\
C_j^i &: \text{jth value of the } i^{th} clustercenter
\end{aligned}
$$

The cluster centers are initially randomly assigned, and each data point $x_i$ is then assigned to the cluster with the minimum distance. After all data points have been assigned, new cluster centers are determined by calculating the weighted average of all data points within each cluster. This process shifts the cluster centers toward the center of the data distribution. Iterations continue until there is no further change in the cluster centers. The $k$-means algorithm is particularly effective for handling crisp data with distinct boundaries.

Note: Add more images here....

4. K-means advantage.

   Firstly, its relatively simple implementation makes it accessible to users across various domains. Moreover, its scalability to large datasets ensures that it can efficiently handle substantial amounts of data, making it suitable for applications with extensive data collections. Additionally, K-means guarantees convergence, meaning that it will eventually reach a stable solution. Furthermore, K-means easily adapts to new examples, making it versatile for dynamic datasets. Lastly, its ability to generalize to clusters of different shapes and sizes, including elliptical clusters, enhances its utility in identifying complex patterns within the data.

5. K-means advantage.
   In real-world scenarios, data seldom exhibit clear-cut groupings. Instead, clusters often possess indistinct boundaries, blending into the data space and frequently overlapping neighboring clusters. This phenomenon arises due to the inherent complexities of natural data, which are characterized by several limitations:

1. **Unclear Knowledge**: The data may contain elements with uncertain or problematical attributes.
2. **Ambiguity**: Data points may lack definitiveness or determination, leading to vague cluster boundaries.
3. **Doubtfulness**: Certain information about the data may be lacking, resulting in uncertainties within the clusters.
4. **Interpretational Ambiguity**: Data may offer multiple interpretations, leading to ambiguous cluster definitions.
5. **Instability**: Data characteristics may vary over time, making cluster boundaries less steady.
6. **Susceptibility to Change**: Data may not be inherently dependable or reliable, introducing fluctuations in cluster formations.

These inherent limitations underscore the challenge of accurately delineating clusters in real-world datasets.

### 2.1.2 Fuzzy Clustering

1. To lay the groundwork for our exploration of 'Fuzzy Clustering', we will first introduce the concept of 'Fuzzification', the process of converting crisp or uncertain data into fuzzy representations.

   Fuzzification: It is the method of transforming a crisp quantity(set) into a fuzzy quantity(set). This can be achieved by identifying the various known crisp and deterministic quantities as completely nondeterministic and quite uncertain in nature. This uncertainty may have emerged because of vagueness and imprecision which then lead the variables to be represented by a membership function as they can be fuzzy in nature.

   For example, when I say the temperature is 45° Celsius the viewer converts the crisp input value into a linguistic variable like favorable temperature for the human body, hot or cold.

2. Fuzzy clustering (soft clustering) means that an object is possible to be in several clusters. Fuzzy C-Means classified as soft clustering method. Fuzzy c-means clustering involes two processes: the calculation of cluster centers and the assignment of points to these centers using a form of Euclidean distance.

The algorithm shares many similarities with K-means clustering. However, it distinguishes itself by assigning membership values to data items within a range of 0 to 1. So it incorporates fuzzy set's concepts of partial membership and forms overlapping clusters to support it. The algorithm needs a fuzzification parameter m in the range [1,n] which determines the degree of fuzziness in the clusters. When m reaches the value of 1 the algorithm works like a crisp partitioning algorithm and for larger values of m the overlapping of clusters is tend to be more. The algorithm calculates the membership value $\mu$ with the formula:

$$\mu_j(x_i) = \frac{(\frac{1}{d_j})^{\frac{1}{m-1}}}{\sum_{k=1}^{k}(\frac{1}{d_k})^{\frac{1}{m-1}}} \tag{2}$$

where:

- $\mu_j(x_i)$: is the membership of $x_i$ in the $j$th cluster,
- $d_{ji}$: is the distance of $x_i$ in cluster $c_j$,
- $m$: is the fuzzification parameter,
- $p$: is the number of specified clusters,
- $d_{ki}$: is the distance of $x_i$ in cluster $C_k$.

The new cluster centers are calculated with these membership values using the exp. 4

$$c_j = \frac{\sum_i [\mu_i(x_i)]^m x_i}{\sum_i [\mu_j(x_i)]^m} \tag{3}$$

Where:

$C_j$ : the center of the$j^{th}$cluster
$x_i$ : is the $i^{th}$ data point
$u_j$ : is the function which returns the membership
$m$ : is the fuzzification parameter

3. Fuzzy c-means algorithm is as following:

**Algorithm 1:** Fuzzy C-Means Algorithm

**Input:** Number of clusters $p$, fuzzification parameter $m$, cluster centers $C_j$

**Output:** Estimated cluster centers $C_j$

Initialize $C_j$ randomly;

**repeat**

  **for** $i = 1$ **to** $n$ **do**

    Update $\mu_j(x_i)$ applying equation (3);

  **end**

  **for** $j = 1$ **to** $p$ **do**

    Update $C_j$ with equation (4) using current $\mu_j(x_i)$;

  **end**

  Check for convergence ($C_j$ estimate stabilize);

**until** $C_j$ *estimate stabilize*;

4. **Limitations of the algorithm.**
   The main drawbacks are due to the restriction that the sum of membership values of a data point xi in all the clusters must be equal to one as in expression (4). This restriction tends to give high membership values for the outlier points. So the algorithm has difficulty in handling outlier points. Secondly the membership of a data point in a cluster depends directly on the membership values of other cluster centers and this sometimes happens to produce undesirable results.

$$\sum_{j=1}^{p} \mu_j(x_i) = 1 \tag{4}$$

5. **Limitations of the algorithm.**
   Adaptive fuzzy clustering algorithm is similar to c-means algorithm in many ways and it supports the concept of partial memberships for data points in clusters. The main difference is that it removes the restrictions imposed in c-means algorithm through expression (4). The algorithm calculates fuzzy membership values for a data points through a new method as given in exp. 5

$$c_j = \frac{\sum_i [\mu_j(x_i)]^m x_i}{\sum_i [\mu_j(x_i)]^m} \tag{5}$$

   Where:

- $C_j$: Center of the $j^{th}$ cluster
- $x_i$: $i^{th}$ data point
- $\mu_j$: Function returning the membership
- $m$: Fuzzification parameter