Journal of Modern Technology and Engineering
Vol.9, No.2, 2024, pp.69-93
https://doi.org/10.62476/jmte9269

COMPREHENSIVE ANALYSIS OF FUZZY C - MEANS CLUSTERING AND ITS VARIANTS

Orkun Gürler[1*], **Efendi Nasibov**[2]

[1]The Graduate School of Natural and Applied Sciences, Dokuz Eylül University, İzmir, Turkiye
[2]Department of Computer Science, Dokuz Eylül University, İzmir, Turkiye

Abstract. Fuzzy clustering algorithms are one of the most important techniques for analysing and extracting information from data when working with datasets containing overlapping clusters. Fuzzy clustering provides a more precise representation of complex data structures compared to conventional crisp (hard) clustering approaches. It accomplishes this by allowing data points to be assigned to multiple clusters with different degrees of membership. This paper provides an extensive review of various fuzzy clustering algorithms, such as Fuzzy C - Means (FCM) and its variations including Gustafson - Kessel (GK), Noise Clustering (NC), Possibilistic C - Means (PCM), Possibilistic Fuzzy C - Means (PFCM), Credibilistic Fuzzy C - Means (CFCM), and Kernel Fuzzy C - Means (KFCM). Their underlying mathematical theories, pseudocodes and applicability to different types of data are reviewed. Furthermore, the time complexity of these algorithms are analysed and a detailed comparison is presented to clarify their performance and scalability. By examining both theoretical aspects and empirical results, this study aims to provide a comprehensive knowledge of the trade-offs between computational efficiency and clustering accuracy. This analysis is intended to serve as a resource for researchers and practitioners in selecting appropriate fuzzy clustering techniques for their specific applications.

Keywords: Clustering Analysis, Fuzzy Clustering, Cluster Validity, Pattern Recognition.
AMS Subject Classification: 62H30, 68T10, 03E72, 94D05, 62A86.
Corresponding author: Orkun Gürler, The Graduate School of Natural and Applied Sciences, Dokuz Eylül University, Buca, İzmir, Turkiye, e-mail: *orkun.gurler@ogr.deu.edu.tr*
Received: 25 May 2024; Revised: 18 July 2024; Accepted: 30 July 2024; Published: 30 August 2024.

1 Introduction

Clustering algorithms are designed to partition data into classes, referred to as clusters, based on inherent similarities and differences. The clusters formed by these algorithms are derived from these similarities and differences, and provide valuable insights into the underlying behavior of the data. Consequently, clustering algorithms are widely used across various disciplines, particularly in computer science, to generate meaningful information from data.

Clustering as itself often refers to crisp (hard) clustering. In crisp clustering, each data point belongs to only one cluster, which can lead to the improper formation of overlapping clusters and the extraction of inaccurate information from datasets (Hartigan, 1973; Jain and Dubes, 1988).

With the introduction of Zadeh (1965)'s fuzzy set theory and the continuous development of fuzzy clustering algorithms, more realistic clusters and more accurate insights can be produced

How to cite (APA): Gürler, O., Nasibov E. (2024). Comprehensive analysis of fuzzy C - means clustering and its variants. *Journal of Modern Technology and Engineering*, *9*(2), 69-93 https://doi.org/10.62476/jmte9269

from the data compared to crisp clustering algorithms. The underlying concept of fuzzy clustering is that data points can belong to multiple clusters with different degrees, simultaneously. These membership degrees should range between 0 and 1. Thus, the degree to which a data point belongs to a particular cluster is explicitly indicated.

Although the use of fuzzy set theory in clustering algorithms was first proposed by Bellman et al. (1966), the first fuzzy clustering algorithm was introduced by Ruspini (1969). Dunn (1973) further developed this algorithm by incorporating a regression analysis approach, and it was later generalized by Bezdek (1974), becoming widely known as the Fuzzy C-Means (FCM) algorithm.

While these algorithms generally outperform crisp clustering algorithms, their performance significantly declines in datasets containing noise and outliers. Data with noise and outliers can undesirably shift cluster centers, leading to the generation of inaccurate information. To overcome these challenges and determine more accurate cluster centers, different similarity and dissimilarity metrics have been explored in fuzzy clustering algorithms.

The clusters generated by fuzzy clustering algorithms that use the Euclidean distance function are inherently spherical, which may not accurately capture the behavior of certain types of data. To address this issue, Gustafson and Kessel (1979) proposed using of a generalized distance function, aiming to resolve this limitation. The comparison of various distance functions for FCM algorithm has been addressed by Arora et al. (2019).

FCM algorithm performs significantly worse with noisy data. As a solution, the Noise Clustering algorithm was introduced to the literature by Dave (1991). (Krishnapuram and Keller, 1993) introduced the Possibilistic C-Means (PCM) algorithm, which uses the possibility of the data belonging to clusters as a metric for clustering. Several advanced PCM algorithms have been analyzed in the work of Zhang and Leung (2004). The literature also includes various PCM algorithms that integrate different methodologies to improve clustering performance (Zhang and Chen, 2003b; Schneider, 2000; Xie et al., 2008; Zhang et al., 2017). PCM algorithm aims to mitigate the poor results associated with noisy data. However, as outliers continued to be a problem, Pal et al. (1997) developed a hybrid method known as Fuzzy Possibilistic C-Means (FPCM). This method evaluated both the probability and possibility of data belonging to clusters. Despite this, the FPCM algorithm did not successfully achieve its goal of delivering good results with data with outliers. Consequently, Pal et al. (2005) updated the objective function of FPCM and created the Possibilistic Fuzzy C-Means (PFCM) algorithm. Chintalapudi and Kam (1998a) proposed a constant in the Credibilistic Fuzzy C-Means (CFCM) algorithm that reduces membership values for outlier data when calculating the weights of the data.

While the FCM algorithm can perform well on linearly separable datasets, it may not yield accurate results on non-linear, and high-dimensional datasets. To address this issue, Girolami (2002) introduced the concept of kernel-based clustering. Additionally, kernel-based fuzzy clustering algorithms have been introduced in Zhang and Chen (2003a) and Wu et al. (2003). But, Tsai and Lin (2011) proposed a modified variance function for the Kernel Fuzzy C-Means (KFCM) and presented a comparison with the FCM algorithm.

Similarity and dissimilarity metrics can vary in fuzzy clustering algorithms just as they do in crisp clustering algorithms. The literature contains numerous algorithms that apply different logics and mathematical theories. This article aims to explain in detail how some fundamental algorithms based on different foundations are constructed and the logics they are based on.

Section 2 provides the introduction and detailed examination of aforementioned algorithms. Section 3 introduce cluster validity indices in detail along with their pseudocodes. In Section 4, the results of cluster validity indices produced by these algorithms on both synthetically generated datasets and real-world datasets are compared. Section 5 offers an in-depth analysis of the findings and discusses their applicability. Finally, Section 6 concludes the article with a summary of the main results and offers suggestions for further research.

2 Explanation of Algorithms

Clustering algorithms involve the use of a non-linear mathematical computation. The objective is to minimise the cost function, which includes a quadratic equation. By reducing the objective function, the data's membership to the clusters will improve.

Numerous approaches have been developed for specific reasons to address various difficulties that arise in data clustering. The next subsections address some known algorithms which are primarily associated with the FCM algorithm.

This section will use letters and symbols to describe the algorithms and pseudocodes. Throughout the text, n denotes the number of data, d the dimension of the data (number of features), and c the number of clusters. In addition, $X_{(n\times d)}$ represents the data set, $U_{(c\times n)}$ represents the membership degrees, $V_{(c\times d)}$ represents the cluster centres. In functions and algorithms, u_{ij} denotes the membership degree of data j to cluster i, v_i the vector of the i th cluster centre. Similarly, m denotes the fuzziness, ε the error coefficient, and l_{max} the maximum number of iterations.

The distance of the data x_j from the cluster centre v_i, with A being the norm matrix, is expressed by the equation $D_{ij}^2 = \|x_j - v_i\|_A^2 = (x_j - v_i)'A(x_j - v_i)$. When A is defined as the unit norm matrix, D_{ij}^2 gives the squared Euclidean distance and this means that the clusters are formed spherical. Also, using the covariance matrix will give the Mahalanobis distance function. Unless stated otherwise, it should be understood that the squared Euclidean distance function is used in the algorithm. Additional symbols, that are not included in this paragraph, will be described inside the algorithm where they are utilised.

2.1 Fuzzy C-Means (FCM) Algorithm

FCM clustering algorithm is a popular algorithm used in various applications. FCM is a method used in machine learning and data mining to partition a dataset into clusters, and it is commonly employed in pattern recognition and image segmentation. It is an extension of the K-Means (Hard C-Means) clustering algorithm (Sebestyen, 1962; MacQueen et al., 1967), but with softened assignment of data points to clusters. Influenced by Ruspini (1969) work, Dunn (1973) created what can be considered a specific case of the FCM algorithm, using the Euclidean norm. Later, Bezdek (1973b) generalized this situation, completing the algorithm. In 1984, it was implemented by Bezdek et al. (1984) and became widely recognized under its current name. In FCM, every data point belongs to each cluster, and this membership is expressed through a value called the membership degree, which ranges between 0 and 1. This allows for more flexible and nuanced cluster assignments.

In most clustering algorithms and FCM, the primary goal is to minimize the objective function to create a well-defined and dense data domain. The objective function for this optimization problem is described with the equation

$$J = \sum_{j=1}^{n}\sum_{i=1}^{c} u_{ij}^m D_{ij}^2. \tag{1}$$

This function sums the squared differences between data points and cluster centers, so the most compact clusters are found by minimizing the function. The main constraint of this objective function is that the sum of a data point's membership degrees across all clusters must be equal to 1.

$$\sum_{i=1}^{c} u_{ij} = 1 \tag{2}$$

The value known as the fuzzifier, denoted as m, should be greater than 1. In FCM and most clustering algorithms, this value is typically set to $m = 2$ (Bezdek, 1973b).

As mentioned in the begining of this section, the value D_{ij}^2 indicates the Euclidean distance function between the data point x_j and the cluster center v_i. The Euclidean norm used creates spherical clusters. During the determination of membership degrees, inter-cluster information is utilized.

$$u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{D_{ij}^2}{D_{kj}^2} \right)^{\frac{1}{m-1}} \right]^{-1} \tag{3}$$

FCM utilizes a membership degree $u(x, v)$ for each data point x_j and each cluster center v_i. When initializing the algorithm, specified constraints are applied to set up the membership degrees appropriately. Another method for initiating the algorithm is to first assign the cluster centers. Although this method might increase the number of iterations in some cases, if the cluster centers are chosen properly, the iteration speed can significantly decrease.

In every iteration, membership degrees are assigned based on the distances of the data points to the cluster centers. The objective function is calculated by aggregating the distances from data points to the cluster centers, weighted by their membership levels. As the value of the objective function decreases, more compact clusters are achieved. Cluster centers are formed by calculating their weighted averages. After updating the membership degrees, cluster centers are recalculated. This iteration continues until convergence is achieved.

$$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m} \tag{4}$$

The pseudocode of FCM is shown in Algorithm 1.

Algorithm 1: FCM Algorithm

Input: $X, c, m, \varepsilon, l_{\max}$

1 $U = random([c, n])$

2 **for** $l = 1$ **to** $l_{\max}$ **do**

3 $\quad v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}$

4 $\quad u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{D_{ij}^2}{D_{kj}^2} \right)^{\frac{1}{m-1}} \right]^{-1}$

5 $\quad$ **if** $\|V^{l-1} - V^l\| \leq \varepsilon$ **then**

6 $\quad\quad$ Stop

Output: U, V

Generally, the time complexity of algorithms depends on the size of the data. However, the complexity analysis presented in this article has been calculated with high precision, incorporating variables such as the number of data points, the dimensions of the data, the number of clusters, and the number of iterations.

When the time complexity of the algorithm is calculated line by line, the result will be as follows.

Line 1. Generating initial weight matrix would cost $O(n \cdot d \cdot c)$ times. Since this process can be precomputed outside of the algorithm, this line can be disregarded.

Line 3. When the numerator and denominator are calculated separately, the numerator costs $O(n \cdot d \cdot c)$, and the denominator costs $O(n \cdot c)$.

Line 4. If the distance function is calculated for each value, computing the membership degree matrix will cost $O(n \cdot d \cdot c^2)$. However, if the distance matrix is calculated before computing the membership degrees, using linear algebra tools[1], the cost of computing the membership degree matrix will be reduced to $O(n \cdot c^2)$.

Linear algebra tools can also be utilized to calculate the centroid and membership degree matrix, which can significantly reduce the computational cost of the algorithm. In this and subsequent algorithms, it should be assumed that the distance matrix is calculated before being used in the respective functions.

Since these operations will stop once convergence is achieved, the number of iterations l is used in the complexity analysis instead of the maximum number of iterations. When computed within the loop and taking into account l iterations, the overall complexity will be

$$O((c+d) \cdot n \cdot c \cdot l).$$

2.2 Gustafson-Kessel (GK) Algorithm

GK clustering algorithm is a generalization of the FCM algorithm that uses a modified Mahalanobis distance function instead of the Euclidean distance function (Gustafson and Kessel, 1979). This generalization enables the GK algorithm to utilize cluster-specific fuzzy covariance matrices, allowing it to adapt to clusters of varying shapes and orientations. This adaptability results in more flexible and accurate clustering outcomes compared to algorithms that assume spherical clusters. The GK algorithm has been employed across various fields due to its ability to handle complex cluster structures effectively, and it has been adapted to meet the specific needs of different domains (Gath and Geva, 1989; Babuka et al., 2002; Filev and Georgieva, 2010; Bas and Egrioglu, 2022).

The GK algorithm operates by minimizing the objective function in Equation (1) that measures the compactness of clusters, similar to the FCM algorithm. Unlike FCM, the modified Mahalanobis distance function used in the GK algorithm plays a significant role in understanding the structure of the data. This adjustment means that clusters are formed as hyper-ellipsoids rather than spheres. The algorithm takes volume parameters (typically $\rho_i = 1$) as input for each cluster, making it robust against changes in cluster densities and more effective at handling datasets with overlapping clusters.

The modified Mahalanobis distance function is as described with the following equations (Equation (5)).

$$P_i = \frac{\sum_{j=1}^{n} u_{ij}^m (x_j - v_i)(x_j - v_i)^T}{\sum_{j=1}^{n} u_{ij}^m} \tag{5}$$

$$D_{ij}^2 = (x_j - v_i)^T (\rho_i \det(P_i))^{\frac{1}{d}} P_i^{-1} (x_j - v_i)$$

The pseudocode of GK is shown in Algorithm 2.

The time complexity can be analyzed as follows.

Line 3. The calculation of the cluster centers is performed in the same manner as in FCM, and is detailed in its respective section.

- Assuming the distance matrix is generated separately from the membership degree calculation, the fuzzy Mahalanobis distance function will cost $O(n \cdot c \cdot d^2 + c \cdot d^3)$.

[1]The use of linear algebra tools like ATLAS (Whaley et al., 2001) or BLAS (Whaley and Petitet, 2005) -that are also available in Python programming packages such as Numpy Harris et al. (2020)- for matrix operations, will significantly reduce the time complexity.

Algorithm 2: GK Algorithm

Input: $X, c, \rho, m, \varepsilon, l_{\max}$

1 $U = random([c, n])$

2 **for** $l = 1$ **to** $l_{\max}$ **do**

3 $v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}$

4 $u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{D_{ij}^2}{D_{kj}^2} \right)^{\frac{1}{m-1}} \right]^{-1}$

5 **if** $\|V^{l-1} - V^l\| \leq \varepsilon$ **then**

6 Stop

Output: U, V

Line 4. By calculating the distance matrix separately from the membership function, the membership degree matrix will be generateda as in FCM. i.e. $O(n \cdot c^2)$.

With the inclusion of the number of iterations, the time complexity for the GK algorithm will be as follows.

$$O((n + d) \cdot d^2 \cdot c \cdot l)$$

2.3 Noise Clustering (NC) Algorithm

NC algorithm was first introduced by Dave (1991) to address the sensitivity of FCM-based algorithms to noise and outliers. The approach relaxes the constraint on membership degrees, ensuring that the sum of a noise point's memberships across all valid clusters is not forced to equal one. This modification reduces the influence of noise and outliers on clustering results.

$$\begin{aligned} \sum_{i=1}^{c} u_{ij} &= 1 - u_{Nj} \\ \sum_{i=1}^{c} u_{ij} &< 1 \end{aligned} \tag{6}$$

The NC algorithm, which has been applied across various domains (Shen et al., 2005; Rehm et al., 2007; Murali et al., 2018), works by identifying data points that do not conform to the defined clusters. These points are then assigned to a separate cluster known as the "noise cluster". This method aims to enhance the accuracy of clustering in datasets with significant noise.

In this approach, a noise cluster is added alongside the existing clusters, and the distance between each data point and to noise cluster is calculated as follows (Equation (7)), where the value of λ is typically chosen within the range of $[0.005, 0.5]$ (Dave, 1991).

$$\delta^2 = \frac{\lambda}{nc} \sum_{j=1}^{n} \sum_{i=1}^{c} D_{ij}^2 \tag{7}$$

This approach permits noise points to have very low membership values in the valid clusters. As a result, the objective function in the NC method will be defined as shown in Equation (8).

$$J = \sum_{j=1}^{n} \sum_{i=1}^{c} u_{ij}^m D_{ij}^2 + \sum_{j=1}^{n} \delta^2 \left(1 - \sum_{k=1}^{c} u_{kj} \right)^m \tag{8}$$

The membership degree update equation for this formulation is obtained by differentiating the NC objective function with respect to u_{ij}.

$$u_{ij} = \left[\left(\frac{D_{ij}^2}{\delta^2} \right)^{\frac{1}{m-1}} + \sum_{k=1}^{c} \left(\frac{D_{ij}^2}{D_{kj}^2} \right)^{\frac{1}{m-1}} \right]^{-1} \tag{9}$$

The first term in the denominator of the membership degree calculation function will become large for outliers, resulting in small membership degrees for outliers across all valid clusters.

The pseudocode of NC is shown in Algorithm 3.

Algorithm 3: NC Algorithm

Input: $X, c, \lambda, m, \varepsilon, l_{\max}$

1 $U = random([c, n])$

2 **for** $l = 1$ **to** $l_{\max}$ **do**

3 $v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}$

4 $\delta^2 = \frac{\lambda}{nc} \sum_{j=1}^{n} \sum_{i=1}^{c} D_{ij}^2$

5 $u_{ij} = \left[\left(\frac{D_{ij}^2}{\delta^2} \right)^{\frac{1}{m-1}} + \sum_{k=1}^{c} \left(\frac{D_{ij}^2}{D_{kj}^2} \right)^{\frac{1}{m-1}} \right]^{-1}$

6 **if** $\|V^{l-1} - V^l\| \leq \varepsilon$ **then**

7 Stop

Output: U, V

The time complexity for Noise Clustering is similar to that of FCM.

Line 3. The calculation of the cluster centers does not differ, i.e. $O(n \cdot d \cdot c)$.

Line 4. Noise cluster distance calculation costs $O(n \cdot c)$

Line 5. By calculating the distance matrix separately from the membership function, the membership degree matrix will be generated again with complexity $O(n \cdot c^2)$ as in FCM.

With the inclusion of the number of iterations, the time complexity for the NC algorithm will be as follows.

$$O((c + d) \cdot n \cdot c \cdot l)$$

2.4 Possibilistic C-Means (PCM) Algorithm

The PCM algorithm was introduced by Krishnapuram and Keller (1993) to overcome the sensitivity of the traditional FCM algorithm to noise and outliers. Unlike FCM, which produces probabilistic memberships for data points, PCM generates possibility (typicality) degrees. This approach removes the constraint that the membership values must sum to one for each data point, enabling more effective handling of overlapping clusters.

$$0 < \sum_{i=1}^{c} t_{ij} < N \tag{10}$$

Here, t_{ij} is the possibility of data point x belonging to the cluster i, and $T = [t_{ij}]_{c \times n}$ is called the typicality matrix.

PCM updates the objective function of FCM to more effectively handle overlapping clusters.

$$J = \sum_{j=1}^{n}\sum_{i=1}^{c} t_{ij}^{m} D_{ij}^{2} + \sum_{i=1}^{c} b_i \sum_{j=1}^{n} (1 - t_{ij})^{m} \tag{11}$$

The update of cluster centers is the same as in the FCM algorithm, as shown in the corresponding Equation (4).

The algorithm can be initialized using either the typicality matrix or the cluster center matrix; however, Krishnapuram and Keller (1993) and Barni et al. (1996) recommended initializing the algorithm with the results obtained from the FCM algorithm.

As with other FCM-based algorithms, taking the derivative of the objective function with respect to, in this manner, t_{ij} and setting it equal to zero will yield the typicality matrix.

$$t_{ij} = \left[1 + \left(\frac{D_{ij}^{2}}{b_i} \right)^{\frac{1}{m-1}} \right]^{-1} \tag{12}$$

The value b_i represents the distance at which a data point's membership value equals 0.5. As a result, b_i should be selected based on the desired bandwidth of the possibility distribution for each cluster. If the clusters are expected to be similar, this value can be set to a similar or identical value across all clusters.

Ideally, b_i should correspond to the overall size and shape of cluster v_i.

$$b_i = \lambda \frac{\sum_{j=1}^{n} u_{ij}^{m} D_{ij}^{2}}{\sum_{j=1}^{n} u_{ij}^{m}} \tag{13}$$

In practice, Equation (13) has proven to be an effective method for determining b_i (Krishnapuram and Keller, 1993). This approach ensures that b_i is proportional to the average fuzzy intra-cluster distance of cluster v_i. Typically, the constant λ is set to 1.

The pseudocode of PCM is shown in Algorithm 4.

Algorithm 4: PCM Algorithm

Input: $X, c, \lambda, m, \varepsilon, l_{\max}$

$[U, V] = fcm(X, c, m, \varepsilon, l_{\max})$

$b_i = \lambda \frac{\sum_{j=1}^{n} u_{ij}^{m} D_{ij}^{2}}{\sum_{j=1}^{n} u_{ij}^{m}}$

for $l = 1$ **to** $l_{\max}$ **do**

  $t_{ij} = \left[1 + \left(\frac{D_{ij}^{2}}{b_i} \right)^{\frac{1}{m-1}} \right]^{-1}$

  $v_i = \frac{\sum_{j=1}^{n} t_{ij}^{m} x_j}{\sum_{j=1}^{n} t_{ij}^{m}}$

  if $\|V^{l-1} - V^{l}\| \leq \varepsilon$ **then**

    Stop

Output: T, V

When calculating the time complexity of the PCM algorithm, which is initialized using the FCM algorithm as recommended, the resulting time complexity will be lower than that of

FCM. Therefore, the time complexity of PCM is upperbounded with FCM's time complexity $O((c+d)\cdot n\cdot c\cdot l)$. However, excluding the time complexity of the FCM algorithm, the time complexities for each step are as follows.

Line 2. The calculation of the bandwith algorithm costs $O(n\cdot c)$.

Line 4. Typicality matrix requires $O(n\cdot c)$

Line 5. The calculation of the cluster centers is the same as in FCM, with a time complexity of $O(n\cdot d\cdot c)$.

Including the number of iterations, without initializing function, the total time complexity would be

$$O(n\cdot d\cdot c\cdot l).$$

2.5 Possibilistic Fuzzy C-Means (PFCM) Algorithm

The PFCM algorithm is a hybrid clustering method that combines the advantages of the FCM and PCM algorithms. Pal et al. developed PFCM to address the limitations of FCM, such as its sensitivity to noise and outliers, and PCM, which can struggle to identify accurate cluster prototypes in the presence of overlapping clusters (Pal et al., 2005). PFCM merges the probabilistic membership model of FCM with the possibilistic typicality model of PCM, creating a more robust and versatile clustering approach.

Although the hybrid algorithm combining FCM and PCM was first introduced by Pal et al. (1997) with the name FPCM (Fuzzy Possibilistic C-Means), the PFCM algorithm was designed to resolve specific issues identified in it.

While PFCM is particularly effective in handling noisy data and cases where clusters are not clearly separable, it is crucial to carefully select the parameters c_f, c_p, and b. Incorrect parameter selection can significantly degrade the clustering performance.

The PFCM algorithm aims to minimize a composite objective function that incorporates both fuzzy membership and possibilistic typicality. A significant issue in the original FPCM algorithm is the constraint that requires the sum of the typicality values for all data points within a cluster to equal 1. In PFCM, this constraint on typicality values has been relaxed, while the constraint on membership values has been retained. This adjustment results in the objective function presented in Equation (14).

$$J=\sum_{j=1}^{n}\sum_{i=1}^{c}(c_f u_{ij}^m+c_p t_{ij}^\eta)D_{ij}^2+\sum_{i=1}^{c}b_i\sum_{j=1}^{n}(1-t_{ij})^\eta \tag{14}$$

The parameter $m>1$ represents the fuzziness coefficient, and $\eta>1$ indicates the degree of typicality. The constants c_f and c_p determine the relative importance of fuzzy membership and typicality values within the objective function. The parameter b_i, as described in the PCM algorithm, is applied similarly in the PFCM algorithm.

By solving the reduced objective function using the Lagrange multiplier method, the following equation (Equation (15)) is derived. The PFCM algorithm calculates both the fuzzy membership degree and the possibilistic typicality degree for the data.

The update of the membership degree follows the weight update function used in the FCM algorithm. However, in updating the typicality degree, following the relaxation of the FPCM constraint, the constant c_p is incorporated into the function.

$$t_{ij}=\left[1+\left(\frac{c_p}{b_i}D_{ij}^2\right)^{\frac{1}{\eta-1}}\right]^{-1} \tag{15}$$

In contrast to the FCM and PCM algorithms, the PFCM algorithm determines cluster centers by averaging the weighted values, where the weights are derived from both the membership degrees and typicality degrees, each scaled by their respective constants.

$$v_i = \frac{\sum_{j=1}^{n} (c_f u_{ij}^m + c_p t_{ij}^\eta) x_j}{\sum_{j=1}^{n} (c_f u_{ij}^m + c_p t_{ij}^\eta)} \tag{16}$$

The pseudocode of PFCM is shown in Algorithm 5.

Algorithm 5: PFCM Algorithm

Input: $X, c, m, \eta, c_f, c_p, \lambda, \varepsilon, l_{\max}$

$[U, V] = fcm(X, c, m, \varepsilon, l_{\max})$

$b_i = \lambda \frac{\sum_{j=1}^{n} u_{ij}^m D_{ij}^2}{\sum_{j=1}^{n} u_{ij}^m}$

for $l = 1$ **to** $l_{\max}$ **do**

  $t_{ij} = \left[1 + \left(\frac{c_p}{b_i} D_{ij}^2\right)^{\frac{1}{\eta-1}}\right]^{-1}$

  $u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{D_{ij}^2}{D_{kj}^2}\right)^{\frac{1}{m-1}}\right]^{-1}$

  $v_i = \frac{\sum_{j=1}^{n} (c_f u_{ij}^m + c_p t_{ij}^\eta) x_j}{\sum_{j=1}^{n} (c_f u_{ij}^m + c_p t_{ij}^\eta)}$

  if $\|V^{l-1} - V^l\| \leq \varepsilon$ **then**

    Stop

Output: U, V

To calculate and understand the time complexity for the PFCM algorithm, one can refer to the time complexity analyses of the FCM and PCM algorithms which will similarly cost

$$O((c + d) \cdot n \cdot c \cdot l).$$

2.6 Credibilistic Fuzzy C-Means (CFCM) Algorithm

The CFCM algorithm is an extension of the traditional FCM clustering algorithm, designed to enhance clustering performance in uncertain and imprecise datasets. It incorporates elements of both fuzzy logic and credibilistic theory.

Introduced by Chintalapudi and Kam (1998a), the CFCM algorithm is a hybrid clustering method, which addresses the degree of typicality of data points within clusters (Chintalapudi and Kam, 1998b; Zadeh, 1965, 1978). This integration allows the CFCM algorithm to provide a more adaptable and flexible structure for clustering, utilizing a credibilistic approach to assign reliability degrees.

While the CFCM algorithm is resilient to the influence of outliers and noisy data, the inclusion of the credibilistic term increases computational complexity. This can significantly slow down computation, particularly when dealing with large datasets. Nevertheless, the CFCM algorithm offers valuable insights that can aid in better interpretation and understanding of data.

Although the objective function in the CFCM algorithm is the same as the one used in the FCM algorithm, the fuzzy constraint in FCM has been modified in the CFCM algorithm, as represented by Equation (17). In the CFCM algorithm, the sum of the weights of the data points across clusters is expected to be equal to the reliability degree ψ_j of the data point j.

$$\sum_{i=1}^{c} u_{ij} = \psi_j \tag{17}$$

According to Chintalapudi and Kam, the reliability degree is associated with the isolation of a vector (data point) in the feature space. Based on this approach, the algorithm utilizes the k-nearest neighbors method. The selection of the number of nearest neighbors κ is guided by Equation (18). The integer κ represents the minimum number of data points required in an isolated group for it to be considered a cluster. The parameter λ is relatively insensitive to clustering results, with a recommended value of $\lambda = 0.5$.

$$\kappa = ceil(\lambda(n/c)), \qquad \lambda \in (0,1) \tag{18}$$

Let Y represent the vector of the κ nearest neighbors of x_j.

$$Y_j^{\kappa} = \{y_j^1, \cdots, y_j^{\kappa}\} = \{y_j^z\}_{z=1}^{\kappa}, \qquad y_j^z \in X \tag{19}$$

The term μ_j denotes the average distance from x_j to its nearest neighbors.

$$\mu_j = \frac{\sum_{z=1}^{\kappa} \|y_j^z - x_j\|^2}{\kappa}$$

For an outlier data x, most (or all) of its κ nearest neighbors will be 'distant' from data x. Conversely, a non-outlier data point is typically surrounded by many other objects, meaning its κ nearest neighbors will be 'close' to it. As a result, the value of μ_j will be high for outliers and low for non-outliers.

In Equation (20), the μ values are normalized within the range of $(0,1)$ to generate the reliability degrees of the data points. If a data point's reliability degree is 0 ($\psi_j = 0$), it indicates that this point is the 'least typical' vector in the dataset, regardless of whether it is an outlier.

$$\psi_j = 1 - \frac{(\mu_j - \min(\mu_1, \cdots, \mu_n))}{\max(\mu_1, \cdots, \mu_n) - \min(\mu_1, \cdots, \mu_n)} \tag{20}$$

The weight calculation, updated by incorporating the reliability degree is given by

$$u_{ij} = \psi_j \left[\sum_{k=1}^{c} \left(\frac{\|x_j - v_i\|^2}{\|x_j - v_k\|^2} \right)^{\frac{1}{m-1}} \right]^{-1}. \tag{21}$$

No modification has been proposed for the calculation of cluster centers; the function remains the same as in the FCM algorithm.

The pseudocode of CFCM is shown in Algorithm 6.

Note that, the ψ function can be precomputed using the kNN (k-Nearest Neighbors) algorithm, eliminating the need for its calculation within the main algorithm. This suggests that the inclusion of ψ does not significantly impact the overall time complexity of the algorithm. However, it's important to note that the time complexity of the KNN algorithm is $O(n \cdot d \cdot c)$, while the time complexity for computing ψ is $O(n)$. In conclusion, the time complexity of the CFCM algorithm remains the same as that of the FCM algorithm, i.e.

$$O((c+d) \cdot n \cdot c \cdot l).$$

Algorithm 6: CFCM Algorithm

Input: $X, c, \lambda, m, \varepsilon, l_{\max}$

$U = random([c, n])$

$\kappa = ceil(\lambda(n/c))$

$Y_j^{\kappa} = \{y_j^1, \cdots, y_j^{\kappa}\}$

$\mu_j = \frac{\sum_{z=1}^{\kappa} \|y_j^z - x_j\|^2}{\kappa}$

$\psi_j = 1 - \frac{(\mu_j - \min(\mu_1, \cdots, \mu_n))}{\max(\mu_1, \cdots, \mu_n) - \min(\mu_1, \cdots, \mu_n)}$

for $l = 1$ **to** $l_{\max}$ **do**

$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}$

$u_{ij} = \psi_j \left[\sum_{k=1}^{c} \left(\frac{D_{ij}^2}{D_{kj}^2} \right)^{\frac{1}{m-1}} \right]^{-1}$

if $\|V^{l-1} - V^l\| \leq \varepsilon$ **then**

Stop

Output: U, V

2.7 Kernel Fuzzy C-Means (KFCM) Algorithm

The KFCM algorithm incorporates RBF (Radial Basis Function) kernel methods to enhance performance in complex and non-linear datasets. The core idea behind KFCM is the use of a kernel function that transforms the original data points into a new feature space.

Following the introduction of Girolami (2002) idea, Tsai and Lin (2011) attempted to achieve more robust clustering by updating the variance calculation used in the RBF kernel distance function. This demonstrates KFCM's flexibility in selecting different kernel functions. In this section, we utilize the KFCM algorithm developed by Tsai and Lin (2011).

However, KFCM also has certain limitations. This approach requires the computation and storage of the kernel matrix, which can be computationally intensive, particularly for large datasets. Additionally, selecting an appropriate kernel function and its parameters often involves a lengthy trial-and-error process. The literature also includes numerous applications using kernel functions and fuzzy clustering algorithms developed based on these functions (Chen and Zhang, 2004; Dhillon et al., 2004, 2007; Graves and Pedrycz, 2010; Rezaee et al., 2021; Elshenawy et al., 2022).

To obtain the FCM objective function in a high-dimensional feature space, a kernel-based distance function is added. Among the commonly used kernels —Polynomial, Sigmoid, and Gaussian— Tsai and Lin preferred the Gaussian (RBF) kernel. Instead of explicitly calculating the (Φ) value shown in the KFCM objective function in Equation (22), a mapping to the feature space is performed. During this mapping process, the RBF kernel distance function ($K(x, y)$) is again utilized.

$$\begin{gathered} J = \sum_{j=1}^{n} \sum_{i=1}^{c} u_{ij}^m \|\Phi(x_j) - \Phi(v_i)\|^2 \\ \|\Phi(x_j) - \Phi(v_i)\|^2 = K(x_j, x_j) + K(v_i, v_i) - 2K(x_j, v_i) \\ K(x, y) = \exp\left(-\frac{\|x - y\|_A^2}{\sigma^2} \right) \end{gathered} \tag{22}$$

Instead of using the classical sample variance value, Tsai and Lin proposed a modified variance value to better adjust the bandwidth of the kernel distance. The modified variance value

(σ^2) in the kernel function is recommended to be calculated as shown in Equation (23) (Tsai and Lin, 2011).

$$\begin{aligned} \bar{v} &= \frac{1}{n}\sum_{j=1}^{n} x_j \\ \bar{d} &= \frac{1}{n}\sum_{j=1}^{n}\sqrt{\|x_j-\bar{v}\|^2} \\ \sigma^2 &= \frac{1}{n-1}\sum_{j=1}^{n}\left(\sqrt{\|x_j-\bar{v}\|^2}-\bar{d}\right)^2 \end{aligned} \tag{23}$$

Accordingly, the update functions for cluster centers and membership degrees are as presented in Equation (24).

$$\begin{aligned} v_i &= \frac{\sum_{j=1}^{n} u_{ij}^m K(x_j,v_i)x_j}{\sum_{j=1}^{n} u_{ij}^m K(x_j,v_i)} \\ u_{ij} &= \left[\sum_{k=1}^{c}\left(\frac{1-K(x_j,v_i)}{1-K(x_j,v_k)}\right)^{\frac{1}{m-1}}\right]^{-1} \end{aligned} \tag{24}$$

To reduce the number of iterations, the FCM centroid function is used to initialize the cluster centers in the KFCM algorithm.

The pseudocode of KFCM is shown in Algorithm 7.

Algorithm 7: KFCM Algorithm

Input: $X, c, m, \varepsilon, l_{\max}$

$U = random([c,n])$

$\bar{v} = \frac{1}{n}\sum_{j=1}^{n} x_j$

$\bar{d} = \frac{1}{n}\sum_{j=1}^{n}\sqrt{\|x_j-\bar{v}\|^2}$

$\sigma^2 = \frac{1}{n-1}\sum_{j=1}^{n}\left(\sqrt{\|x_j-\bar{v}\|^2}-\bar{d}\right)^2$

$v_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m}$

for $l = 1$ **to** $l_{\max}$ **do**

  $u_{ij} = \left[\sum_{k=1}^{c}\left(\frac{1-K(x_j,v_i)}{1-K(x_j,v_k)}\right)^{\frac{1}{m-1}}\right]^{-1}$

  $v_i = \frac{\sum_{j=1}^{n} u_{ij}^m K(x_j,v_i)x_j}{\sum_{j=1}^{n} u_{ij}^m K(x_j,v_i)}$

  if $\|V^{l-1} - V^l\| \le \varepsilon$ **then**

    Stop

Output: U, V

Notice that, the modified sample variance can be precomputed. As with the algorithms discussed in this article, the distance function can be precomputed before applying the desired

function, such as the squared Euclidean distance function within a kernel function. However, calculating the exponential of each data point's distance to a cluster will take slightly longer.

After incorporating the exponential calculation, the time complexity for the KFCM algorithm can be expressed as follows.

$$O((c + 2^n \cdot d) \cdot n \cdot c \cdot l)$$

3 Cluster Validity

Clustering validity indices are used to assess the quality and validity of clustering algorithm results. These indices help evaluate how well the clustering structure reflects the natural grouping in the data. For fuzzy clustering, traditional crisp clustering validity indices are not directly applicable due to the probabilistic nature of memberships. Therefore, specific validity indices have been developed for fuzzy clustering algorithms, each focusing on different aspects of clustering quality. See the most prominent indices in the literature in (Bezdek and Pal, 1998; Pakhira et al., 2004; Wu and Yang, 2005; Wang and Zhang, 2007).

Indices such as the Partition Coefficient, Partition Entropy, Xie-Beni Index, and Fuzzy Silhouette Index provide quantitative measures of clustering quality, aiding researchers and practitioners in assessing the effectiveness of the clustering algorithm's results. Each index offers unique insights into different aspects of the clustering structure, making them indispensable for a comprehensive understanding of the underlying patterns in the data. Using these indices together enables a robust evaluation and facilitates the determination of the most appropriate clustering configuration.

Note that, in the CFCM algorithm, when calculating the reliability degree, min-max normalization results in membership degrees that can take on a value of 0. As a result, the calculation of Partition Entropy and Fuzzy Silhouette indices yields invalid results when encountering a 0 value. Therefore, the CFCM algorithm does not have values listed in the Partition Entropy and Fuzzy Silhouette index tables.

3.1 Partition Coefficient

The Partition Coefficient (PC) index, introduced by Bezdek (1973a), is one of the simplest and earliest clustering validity indices designed specifically for fuzzy clustering. It measures the degree of fuzziness in the clustering result and is used as an indicator of cluster compactness.

$$\mathrm{PC} = \frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{c} u_{ij}^2 \tag{25}$$

A PC value approaching 1 indicates that the clustering result resembles a crisp partitioning, where most data points have high membership values (close to 1) for a specific cluster and low membership values (close to 0) for all other clusters. This suggests the presence of distinct clusters with minimal uncertainty.

3.2 Partition Entropy

The Partition Entropy (PE) index, also designed by Bezdek (1975), is used to measure the fuzziness in the clustering result. It observes the entropy, or uncertainty, associated with the membership values of data points.

$$\mathrm{PE} = -\frac{1}{n} \sum_{j=1}^{n} \sum_{i=1}^{c} u_{ij} \log_a(u_{ij}) \tag{26}$$

A PE value approaching 0 indicates that the entropy is decreasing. This suggests that data points have high membership values for one cluster while having low membership values for other clusters. Consequently, this indicates that the clusters are well-defined with little to no overlap between them.

Although the PE index is one of the most commonly used indices for comparing different fuzzy clustering results, it is influenced by the number of clusters, similar to the PC. As a result, while PE provides valuable insights into the fuzziness of membership values in the data, it does not address the separation between clusters or the compactness of the clusters themselves.

3.3 Fuzzy Silhouette

The Fuzzy Silhouette Index (FS) is an extension of the traditional silhouette index(Rousseeuw, 1987), adapted for use with fuzzy clustering. Multiple adaptations of the FS index can be found in the literature (Pakhira et al., 2004; Campello and Hruschka, 2006; Rawashdeh and Ralescu, 2012; Bezdek et al., 2016). However, this paper discusses the generalized intra-inter silhouette index and, following Bezdek's example (Bezdek et al., 2016), refers to it as the "fuzzy silhouette" out of respect for the original authors.

The FS provides a measure of how similar each data point is to its own cluster compared to other clusters, while taking into account the membership degrees inherent in fuzzy clustering. This index helps evaluate both the compactness of clusters and their separateness (or overlap).

The distance function used for this index can vary. In the context of the paper, the value d represents the distance between data point j and data point k, rather than the distance between data point j and cluster center i. The D value denotes the $(n \times n)$ distance matrix, which represents the pairwise squared Euclidean distances between all data points.

$$D = \{d_{jk} | j, k = 1, \cdots, n; j \neq k\}$$

The value a_j represents the fuzzy intra-cluster distance, which is the minimum average distance between data point j and the other data points within the same cluster, weighted by their membership values.

$$a_j = \min_{1 \leq i \leq c} \left\{ \frac{\sum_{k=1}^{n} (u_{ij} \wedge u_{ik}) d_{jk}}{\sum_{k=1}^{n} (u_{ij} \wedge u_{ik})} \right\} \tag{27}$$

The value b_j represents the fuzzy inter-cluster distance, which is the minimum average distance between data point j and the data points in other clusters, weighted by their membership values.

$$b_j = \min_{1 \leq r < s \leq c} \left\{ \frac{\sum_{k=1}^{n} (u_{rj} \wedge u_{sk}) \vee (u_{sj} \wedge u_{rk}) d_{jk}}{\sum_{k=1}^{n} (u_{rj} \wedge u_{sk}) \vee (u_{sj} \wedge u_{rk})} \right\} \tag{28}$$

These matrix values are used to generalize the traditional silhouette index that is calculated as follows (Equation (29)).

$$s_j = \frac{b_j - a_j}{\max(a_j, b_j)} \tag{29}$$

The provided formulas can be generalized to fuzzy partitions by simply replacing the Boolean operators used in the traditional silhouette index with their fuzzy equivalents, the fuzzy min-

max operators. By doing so, the a_j and b_j values become applicable to fuzzy partitions, allowing for the calculation of fuzzy silhouette scores. Accordingly, s_j becomes a fuzzy silhouette value.

$$\mathrm{FS} = \frac{1}{n}\sum_{j=1}^{n} s_j \tag{30}$$

The FS index takes values in the range of $[-1, 1]$. A value close to 1 indicates that the data point is well-clustered. A value close to 0 suggests that the data point is on or near the boundary between clusters. However, if the FS value approaches -1, it indicates that the data point has been misclassified into the wrong cluster.

3.4 Xie-Beni

The Xie-Beni Index (XB) (Xie and Beni, 1991) combines the characteristics of cluster compactness and separation into a single measure, allowing for the evaluation of the overall quality of a clustering solution. In its calculation, it suggests dividing the classic FCM objective function by the minimum distance between cluster centers.

$$\mathrm{XB} = \frac{\sum_{j=1}^{n}\sum_{i=1}^{c} u_{ij}^{m}\|x_j - v_i\|^2}{n \min\limits_{1\le i\ne k\le c}\{\|v_k - v_i\|^2\}} \tag{31}$$

The use of the FCM objective function, which includes the Euclidean distance as a distance metric, can influence the XB index, thereby affecting the evaluation of clustering quality. As a result, despite its widespread use in the literature, the XB index, like the FCM algorithm, is sensitive to outliers.

Employing each technique's specific objective function could be beneficial for comparing results across different datasets. However, this would make comparisons between different techniques potentially biased. In this paper, the FCM objective function was used for the XB index across all techniques and datasets to maintain consistency.

4 Computational Experiments

This article includes an experiment that evaluates the efficency of the algorithms that are described. Each of these algorithms rely on distinct mathematical foundations and are designed to handle specific types of data. Therefore, the performance of algorithms on on different datasets to produce valid results. Various cluster validity indices are used to measure the efficency and quality of clustering results. These indices help determine the optimal number of clusters, assess the compactness and separation of clusters, and ensure that the clustering results are meaningful and useful. In order to ensure a reliable comparison, the algorithms were tested on various datasets and cluster validity indices within the same computational environment.

This section provides a description of the computational environment, algorithm parameters, and datasets that used and shows performance.

4.1 Computational Environment

The algorithms were implemented in the Visual Studio Code development environment using the Python programming language (v. 3.12.4) and its libraries Numpy(v. 2.0.1)Harris et al. (2020), Scikit-Learn (v. 1.5.1)Pedregosa et al. (2011), and Matplotlib(v. 3.9.1)Hunter (2007). The experiments were run on a 64-bit Arch Linux operating system installed on a machine built with a Zen4-based 16-core AMD Ryzen$^{\mathrm{TM}}$ 9 7945HX processor using CISC architecture.

4.2 Algorithm Parameters

The parameters for the algorithms were defined once and used consistently across all experiments. For all algorithms, set the fuzziness coefficient $m = 2$, the tolerance $\varepsilon = 0.0001$, and the maximum number of iterations, served as a stopping criterion, $l_{max} = 10000$. Given that the experimental datasets were not large and tests concluded with low iterations, this value was deemed sufficient. However, for real and large datasets, this maximum iteration value may be too high and should be adjusted by the user based on the specific dataset and algorithm being used.

In the GK algorithm, set $\rho = 1_{[1\times c]}$, for the NC algorithm $\lambda = 0.25$, for the PCM algorithm $\lambda = 1$, for the PFCM algorithm $c_f, c_p = 0.5$ and $\lambda = 1$, and finally, for the CFCM algorithm $\lambda = 0.5$. These values were applied consistently across all datasets.

4.3 Datasets

Current clustering techniques are in the developmental stage of research, and the known approaches cannot simultaneously address all of the challenges of the datasets detailed in Nagy (1968). Datasets similar to those described in Nagy (1968) (Figure 1) were created manually using the Numpy and Scikit-Learn packages.

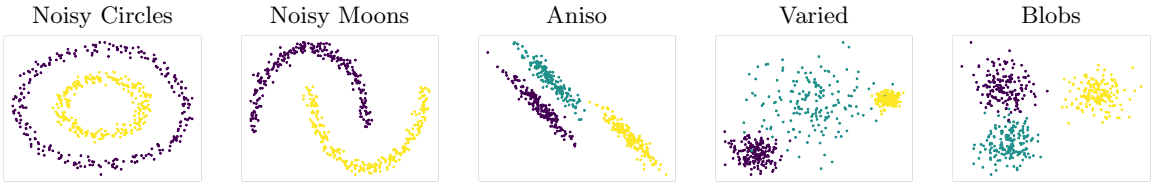


Figure 1: The datasets manually generated based on Nagy (1968) work.

In addition to these, commonly used datasets from MIT (Figure 2), such as Iris (Fisher, 1988), Wine (Aeberhard and Forina, 1991), and Breast Cancer (Wolberg et al., 1995), were also utilized.

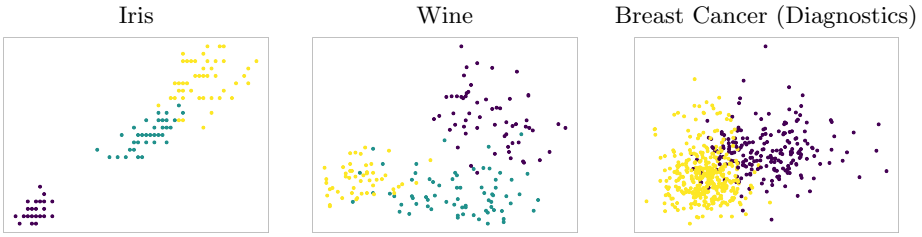


Figure 2: MIT Datasets

Comparing the experiments on both real and synthetic data is beneficial for diversifying the types of data used. This approach allows for an impartial comparison of the strengths and weaknesses of the techniques presented in the article. Details regarding the datasets utilized are provided in Table 1.

4.4 Performance Analysis

During the experiment, every algorithm is executed 100 times for each dataset, and the average of the validity results can be found in the cluster validity index tables, that are Table 2,3,4, and 5.

Table 1: Data, feature and cluster sizes of the datasets

	n	d	c
Noisy Circles	500	2	2
Noisy Moons	500	2	2
Aniso	500	2	3
Varied	500	2	3
Blobs	500	2	3
Iris	150	4	3
Wine	178	13	3
Breast Cancer	569	30	2

Table 2: Partition Coefficient Index (↑)

	N. Circles	N. Moons	Aniso	Varied	Blobs	Iris	Wine	B. Cancer
FCM	0.6946	0.7750	0.7485	0.8417	0.8267	0.7834	0.7909	0.8969
GK	0.5030	0.5152	0.5178	0.5341	0.5155	0.3333	0.3333	0.5000
NC	0.2513	0.3250	0.5529	0.6913	0.6092	0.6007	0.6317	0.5371
PCM	0.4644	0.3774	0.4658	0.6971	0.4142	0.4909	0.7681	0.9844
PFCM	0.7659	0.6049	0.6955	0.6711	0.5932	0.6930	0.7462	1.0765
CFCM	0.4155	0.5673	0.6693	0.7407	0.7119	0.5921	0.7527	0.8735
KFCM	0.5384	0.5573	0.4710	0.4063	0.4076	0.4257	0.6306	0.7174

Table 3: Partition Entropy Index (↓)

	N. Circles	N. Moons	Aniso	Varied	Blobs	Iris	Wine	B. Cancer
FCM	0.4651	0.3685	0.4513	0.3030	0.3440	0.3955	0.3804	0.1809
GK	0.6901	0.6779	0.8451	0.8217	0.8548	1.0986	1.0986	0.6931
NC	0.4996	0.4775	0.4856	0.3360	0.4073	0.4509	0.4055	0.3581
PCM	0.6522	0.6343	0.7020	0.4331	0.5990	0.6281	0.6482	0.3591
PFCM	0.5657	0.5751	0.7239	0.6178	0.6694	0.6520	0.6981	0.3702
CFCM	-	-	-	-	-	-	-	-
KFCM	0.6477	0.6251	0.8951	0.9914	0.9906	0.9662	0.6502	0.4281

Table 4: Fuzzy Silhouette Index (↑)

	N. Circles	N. Moons	Aniso	Varied	Blobs	Iris	Wine	B. Cancer
FCM	0.1823	0.2943	0.4242	0.4984	0.4897	0.4536	0.4526	0.4003
GK	0.0233	0.0495	0.2106	0.3047	0.2727	0	0	0
NC	0.2124	0.3045	0.4397	0.5076	0.4962	0.4552	0.4678	0.2777
PCM	0.0002	0.0006	0.2934	0.4415	0.4303	0.3455	0.0969	0.1398
PFCM	0.0620	0.2032	0.3145	0.4390	0.3887	0.3652	0.3372	0.2100
CFCM	-	-	-	-	-	-	-	-
KFCM	0.0351	0.0561	0.0594	0.0509	0.0517	0.0551	0.2364	0.1377

Table 5: Xie-Beni Index (↓)

	N. Circles	N. Moons	Aniso	Varied	Blobs	Iris	Wine	B. Cancer
FCM	0.3401	0.1226	0.1709	0.0645	0.0790	0.1369	0.1257	0.0597
GK	692	4.5553	0.8600	0.1867	0.2884	1.44e10	2.28e16	1.68e16
NC	0.0529	0.0344	0.0976	0.0351	0.0456	0.0977	0.0749	0.0710
PCM	3.54e5	3.72e4	978	26	0.0466	6.78e4	14	19
PFCM	1.5268	0.1816	0.4401	0.0936	0.0837	0.3882	0.2794	0.9777
CFCM	0.1775	0.0816	0.1441	0.0403	0.0588	0.1738	0.1059	0.0521
KFCM	0.6169	0.5721	1.8270	2.3149	3.6404	2.2192	0.5574	1.4598

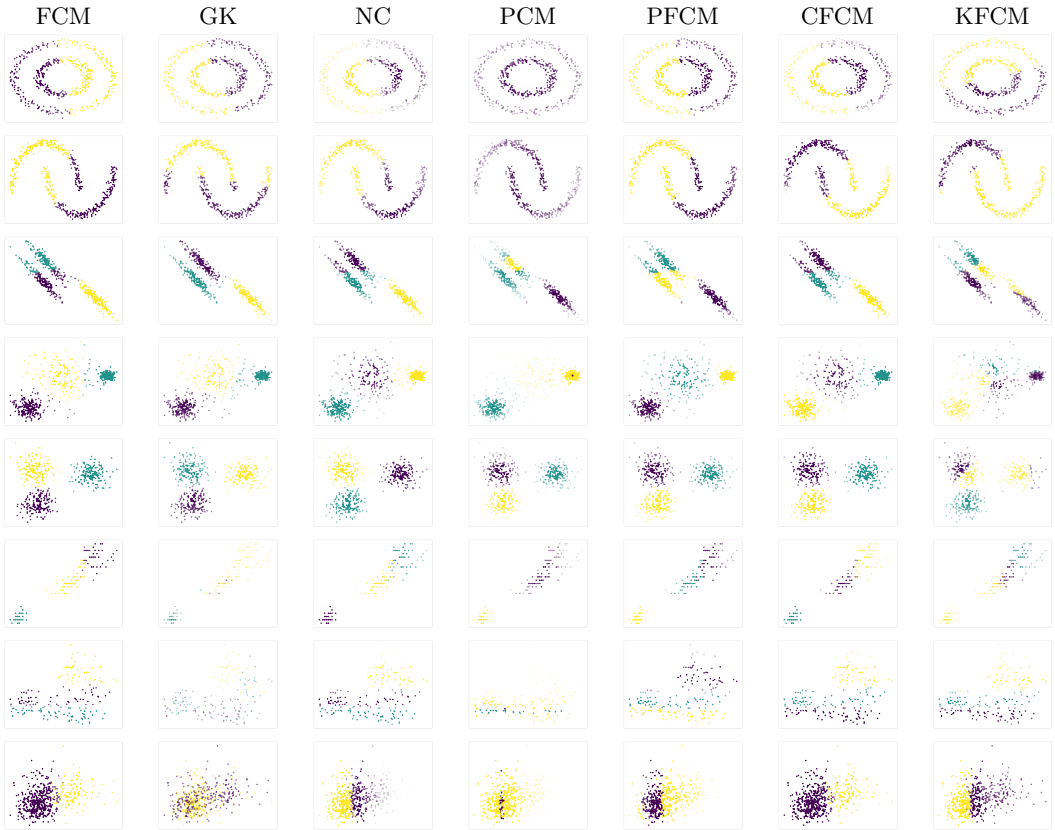
The execution times for the experiments, which were run on the machine detailed earlier, are provided in Table 6. The reported times are in seconds. Each algorithm was executed 100 times, and the average execution times were recorded.

Table 6: Performance Times (sec)

	N. Circles	N. Moons	Aniso	Varied	Blobs	Iris	Wine	B. Cancer
FCM	0.4961	0.0202	0.0570	0.0471	0.0378	0.0194	0.0534	0.0594
GK	5.1231	2.6193	2.7427	1.5053	2.4449	0.1843	0.1762	0.4426
NC	0.3659	0.0793	0.0901	0.0714	0.0633	0.0307	0.0797	0.1940
PCM	0.5260	0.1461	0.2325	0.1131	0.0861	0.0378	0.1633	0.1181
PFCM	0.5899	0.0460	0.1208	0.1055	0.0677	0.0392	0.1070	0.2280
CFCM	0.1525	0.0429	0.0762	0.0640	0.0580	0.0223	0.0495	0.2723
KFCM	1.4765	8.8256	1.1679	1.2252	1.5315	0.3671	0.7208	1.0025

The visual clustering results of the algorithms across different datasets are provided in Figure 3. These visualizations illustrate how each algorithm clusters the data, allowing for a comparative analysis of their performance on various datasets. Different colors represent different clusters and the opacity indicates the membership degree, such that the dots are more opaque if their degree is high and more transparent if otherwise.

Figure 3: Visualization of results



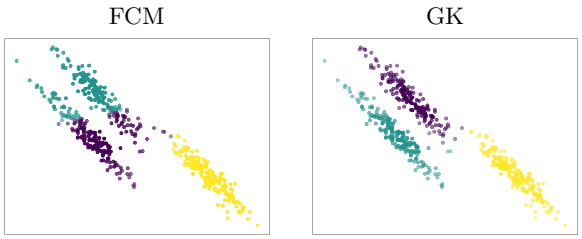
5 Results and Discussion

When examining the tables, it is evident which algorithm yields better results based on the clustering validity index values. However, when considering the compatibility of fuzzy clustering

algorithms with specific datasets, even various validity indices may not accurately reflect an algorithm's performance. One contributing factor to this is the choice of parameters.

The selection of parameters must be appropriate for both the desired algorithm and the dataset; otherwise, the results may not meet expectations. For example, when reviewing all the validity indices, it can be observed that the algorithm providing the best results for the Aniso dataset varies across different indices. This indicates that the "best" algorithm might differ depending on the specific index used, highlighting the importance of careful parameter selection and the potential for variability in algorithm performance across different datasets.

Figure 4: FCM and GK algorithms visual results for Aniso dataset



However, when examining Figure 4, it can be visually observed that the GK algorithm, despite achieving nearly the worst results across all indices, still manages to produce the desired cluster alignment. This highlights that while validity indices are crucial for evaluating the performance of a fuzzy clustering algorithm, the importance of appropriate parameter selection and cluster compatibility should not be overlooked. The visual outcomes suggest that even with poor index scores, an algorithm like GK can still deliver acceptable clustering results when the parameters and data characteristics align well.

Nonetheless, there are some clear conclusions that can be drawn about the algorithms. First, the first five datasets —Noisy Circles, Noisy Moons, Aniso, Varied, and Blobs— are manually generated and feature distributions that are challenging to cluster. The first two datasets, in particular, have structures that are difficult to detect using C-means type clustering algorithms. C-means type algorithms typically rely on the Euclidean distance function, which tends to produce spherical clusters. As a result, for the first two datasets, where the clusters overlap, the clustering results are not realistic. This is because the inherent circular or crescent-shaped structures in these datasets are not well-suited to the assumptions made by C-means algorithms, leading to poor clustering performance. There are algorithms in the literature that are better suited for clustering data with these types of structures. These algorithms are specifically designed to handle complex, non-spherical, or overlapping cluster shapes, providing more accurate and realistic clustering results for such challenging datasets (Dave, 1990; Ester et al., 1996; Nasibov and Ulutagay, 2007; Balakumaran et al., 2010; Nasibov et al., 2015; Bie et al., 2016; Schubert et al., 2017).

In addition, the last three datasets —Iris, Wine, and Breast Cancer— are real-world datasets, and analyzing their results provides a more effective comparison of the performance of clustering functions. These three datasets exhibit various characteristics desirable in the experimental phase, such as noise, outliers, high dimensionality, and large size. The results are consistent across both clustering validity indices and graphical representations.

As expected, the NC algorithm, which is designed to perform well with noisy data, produced the best results on the Breast Cancer dataset, which has a noisy structure. Additionally, the KFCM algorithm performed well due to its effectiveness with high-dimensional data, although it fell behind PCM and PFCM due to its sensitivity to noise.

When examining the performance times, a clear pattern emerges, as previously mentioned. Due to the complexity of calculating the distance function, the KFCM and GK algorithms took significantly longer to execute compared to other algorithms, as seen in the performance time

table.

Lastly, one of the most important points to be drawn from these visual results is not just the accuracy of the fuzzy clustering algorithm in assigning data to the correct group, but also the variations in the information that can be derived from the dataset. As observed, most C-means algorithms generally have a high accuracy rate in cluster assignment. However, it is crucial to consider how strongly the data points belong to their respective clusters. By utilizing different algorithms on the same dataset, diverse insights can be generated, highlighting the different interpretations and information that each algorithm can extract.

6 Conclusion

In this study, a comprehensive comparison of seven different fuzzy clustering algorithms, each based on different mathematical foundations, was performed using eight datasets and four cluster validity indices.

The findings show the performance of various algorithms across diverse datasets, identifying their respective strengths and limitations. The experiments conducted on the eight datasets of varying structures facilitated an assessment of the algorithms' adaptability to different dataset types.

The analysis demonstrated that the structural characteristics of datasets significantly influence the effectiveness of clustering algorithms. The four cluster validity indices employed in this study were integral to the performance assessment of the algorithms. The results indicate that each index exhibits varying levels of sensitivity depending on the specific algorithm and dataset involved.

This study highlights the critical importance of selecting appropriate fuzzy clustering algorithms and cluster validity indices. Future research should consider the inclusion of additional datasets and validity indices to achieve a more comprehensive understanding of algorithmic performance. This comparative analysis provides valuable guidance for selecting suitable fuzzy clustering algorithms and cluster validity indices in data mining and machine learning applications.

References

Aeberhard, S. and Forina, M. (1991). Wine. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5PC7J.

Arora, J., Khatter, K., and Tushir, M. (2019). Fuzzy c-means clustering strategies: A review of distance measures. *Software Engineering: Proceedings of CSI 2015*, pages 153–162.

Babuka, R., Van der Veen, P., and Kaymak, U. (2002). Improved covariance estimation for gustafson-kessel clustering. In *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291)*, volume 2, pages 1081–1085. IEEE.

Balakumaran, T., Vennila, I., and Shankar, C. G. (2010). Detection of microcalcification in mammograms using wavelet transform and fuzzy shell clustering. *CoRR*, abs/1002.2182.

Barni, M., Cappellini, V., and Mecocci, A. (1996). Comments on" a possibilistic approach to clustering". *IEEE Transactions on Fuzzy Systems*, 4(3):393–396.

Bas, E. and Egrioglu, E. (2022). A fuzzy regression functions approach based on gustafson-kessel clustering algorithm. *Information Sciences*, 592:206–214.

Bellman, R., Kalaba, R., and Zadeh, L. (1966). Abstraction and pattern classification. *Journal of Mathematical Analysis and Applications*, 13(1):1–7.

Bezdek, J. C. (1973a). Cluster validity with fuzzy sets. *Journal of Cybernetics*, 3(3):58–73.

Bezdek, J. C. (1973b). *Fuzzy-Mathematics in Pattern classification.*. Cornell University.

Bezdek, J. C. (1974). Numerical taxonomy with fuzzy sets. *Journal of mathematical biology*, 1(1):57–71.

Bezdek, J. C. (1975). Mathematical models for systematics and taxonomy. In *Proceedings of the 8th International Conference on Numerical Taxonomy, 1975*, pages 143–166. Freeman.

Bezdek, J. C., Ehrlich, R., and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & geosciences*, 10(2-3):191–203.

Bezdek, J. C., Moshtaghi, M., Runkler, T., and Leckie, C. (2016). The generalized c index for internal fuzzy cluster validity. *IEEE Transactions on Fuzzy Systems*, 24(6):1500–1512.

Bezdek, J. C. and Pal, N. R. (1998). Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(3):301–315.

Bie, R., Mehmood, R., Ruan, S., Sun, Y., and Dawood, H. (2016). Adaptive fuzzy clustering by fast search and find of density peaks. *Personal and Ubiquitous Computing*, 20:785–793.

Campello, R. J. and Hruschka, E. R. (2006). A fuzzy extension of the silhouette width criterion for cluster analysis. *Fuzzy Sets and Systems*, 157(21):2858–2875.

Chen, S. and Zhang, D. (2004). Robust image segmentation using fcm with spatial constraints based on new kernel-induced distance measure. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(4):1907–1916.

Chintalapudi, K. K. and Kam, M. (1998a). The credibilistic fuzzy c means clustering algorithm. In *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*, volume 2, pages 2034–2039. IEEE.

Chintalapudi, K. K. and Kam, M. (1998b). A noise-resistant fuzzy c means algorithm for clustering. In *1998 IEEE international conference on fuzzy systems proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98CH36228)*, volume 2, pages 1458–1463. IEEE.

Dave, R. N. (1990). Fuzzy shell-clustering and applications to circle detection in digital images. *International Journal Of General System*, 16(4):343–355.

Dave, R. N. (1991). Characterization and detection of noise in clustering. *Pattern Recognition Letters*, 12(11):657–664.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556.

Dhillon, I. S., Guan, Y., and Kulis, B. (2007). Weighted graph cuts without eigenvectors a multilevel approach. *IEEE transactions on pattern analysis and machine intelligence*, 29(11):1944–1957.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57.

Elshenawy, L. M., Chakour, C., and Mahmoud, T. A. (2022). Fault detection and diagnosis strategy based on k-nearest neighbors and fuzzy c-means clustering algorithm for industrial processes. *Journal of the Franklin institute*, 359(13):7115–7139.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

Filev, D. and Georgieva, O. (2010). *An Extended Version of the Gustafson-Kessel Algorithm for Evolving Data Stream Clustering*, chapter 12, pages 273–299. John Wiley & Sons, Ltd.

Fisher, R. A. (1988). Iris. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C56C76.

Gath, I. and Geva, A. B. (1989). Unsupervised optimal fuzzy clustering. *IEEE Transactions on pattern analysis and machine intelligence*, 11(7):773–780.

Girolami, M. (2002). Mercer kernel-based clustering in feature space. *IEEE transactions on neural networks*, 13(3):780–784.

Graves, D. and Pedrycz, W. (2010). Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study. *Fuzzy sets and systems*, 161(4):522–543.

Gustafson, D. E. and Kessel, W. C. (1979). Fuzzy clustering with a fuzzy covariance matrix. In *1978 IEEE conference on decision and control including the 17th symposium on adaptive processes*, pages 761–766. IEEE.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

Hartigan, J. A. (1973). Clustering. *Annual review of biophysics and bioengineering*, 2(1):81–102.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc.

Krishnapuram, R. and Keller, J. M. (1993). A possibilistic approach to clustering. *IEEE transactions on fuzzy systems*, 1(2):98–110.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, pages 281–297. Oakland, CA, USA.

Murali, A., Bhargava, A., and Wright, E. S. (2018). Idtaxa: a novel approach for accurate taxonomic classification of microbiome sequences. *Microbiome*, 6:1–14.

Nagy, G. (1968). State of the art in pattern recognition. *Proceedings of the IEEE*, 56(5):836–863.

Nasibov, E., Atilgan, C., Berberler, M. E., and Nasiboglu, R. (2015). Fuzzy joint points based clustering algorithms for large data sets. *Fuzzy sets and Systems*, 270:111–126.

Nasibov, E. N. and Ulutagay, G. (2007). A new unsupervised approach for fuzzy clustering. *Fuzzy Sets and Systems*, 158(19):2118–2133.

Pakhira, M. K., Bandyopadhyay, S., and Maulik, U. (2004). Validity index for crisp and fuzzy clusters. *Pattern recognition*, 37(3):487–501.

Pal, N. R., Pal, K., and Bezdek, J. C. (1997). A mixed c-means clustering model. In *Proceedings of 6th international fuzzy systems conference*, volume 1, pages 11–21. IEEE.

Pal, N. R., Pal, K., Keller, J. M., and Bezdek, J. C. (2005). A possibilistic fuzzy c-means clustering algorithm. *IEEE transactions on fuzzy systems*, 13(4):517–530.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Rawashdeh, M. and Ralescu, A. (2012). Center-wise intra-inter silhouettes. In *Scalable Uncertainty Management: 6th International Conference, SUM 2012, Marburg, Germany, September 17-19, 2012. Proceedings 6*, pages 406–419. Springer.

Rehm, F., Klawonn, F., and Kruse, R. (2007). A novel approach to noise clustering for outlier detection. *Soft Computing*, 11:489–494.

Rezaee, M. J., Eshkevari, M., Saberi, M., and Hussain, O. (2021). Gbk-means clustering algorithm: An improvement to the k-means algorithm based on the bargaining game. *Knowledge-Based Systems*, 213:106672.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Ruspini, E. H. (1969). A new approach to clustering. *Information and Control*, 15(1):22–32.

Schneider, A. (2000). Weighted possibilistic c-means clustering algorithms. In *Ninth IEEE International Conference on Fuzzy Systems. FUZZ-IEEE 2000 (Cat. No. 00CH37063)*, volume 1, pages 176–180. IEEE.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X. (2017). Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21.

Sebestyen, G. S. (1962). *Decision-making processes in pattern recognition (ACM monograph series)*. Macmillan Publishing Co., Inc.

Shen, S., Sandham, W., Granat, M., and Sterr, A. (2005). Mri fuzzy segmentation of brain tissue using neighborhood attraction with neural-network optimization. *IEEE transactions on information technology in biomedicine*, 9(3):459–467.

Tsai, D.-M. and Lin, C.-C. (2011). Fuzzy c-means based clustering for linearly and nonlinearly separable data. *Pattern recognition*, 44(8):1750–1760.

Wang, W. and Zhang, Y. (2007). On fuzzy cluster validity indices. *Fuzzy sets and systems*, 158(19):2095–2117.

Whaley, R. C. and Petitet, A. (2005). Minimizing development and maintenance costs in supporting persistently optimized BLAS. *Software: Practice and Experience*, 35(2):101–121. `http://www.cs.utsa.edu/~whaley/papers/spercw04.ps`.

Whaley, R. C., Petitet, A., and Dongarra, J. J. (2001). Automated empirical optimization of software and the ATLAS project. *Parallel Computing*, 27(1–2):3–35. Also available as University of Tennessee LAPACK Working Note #147, UT-CS-00-448, 2000 (`www.netlib.org/lapack/lawns/lawn147.ps`).

Wolberg, W., Mangasarian, O., Street, N., and Street, W. (1995). Breast Cancer Wisconsin (Diagnostic). UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5DW2B.

Wu, K.-L. and Yang, M.-S. (2005). A cluster validity index for fuzzy clustering. *pattern recognition letters*, 26(9):1275–1291.

Wu, Z.-d., Xie, W.-x., and Yu, J.-p. (2003). Fuzzy c-means clustering algorithm based on kernel method. In *Proceedings Fifth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2003*, pages 49–54. IEEE.

Xie, X. L. and Beni, G. (1991). A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(08):841–847.

Xie, Z., Wang, S., and Chung, F.-L. (2008). An enhanced possibilistic c-means clustering algorithm epcm. *Soft Computing*, 12:593–611.

Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8(3):338–353.

Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy sets and systems*, 1(1):3–28.

Zhang, D.-Q. and Chen, S.-C. (2003a). Clustering incomplete data using kernel-based fuzzy c-means algorithm. *Neural processing letters*, 18:155–162.

Zhang, D.-Q. and Chen, S.-C. (2003b). Kernel-based fuzzy and possibilistic c-means clustering. In *Proceedings of the International Conference Artificial Neural Network*, volume 122, pages 122–125.

Zhang, J.-S. and Leung, Y.-W. (2004). Improved possibilistic c-means clustering algorithms. *IEEE transactions on fuzzy systems*, 12(2):209–217.

Zhang, Q., Yang, L. T., Chen, Z., and Li, P. (2017). Pphopcm: Privacy-preserving high-order possibilistic c-means algorithm for big data clustering with cloud computing. *IEEE Transactions on Big Data*, 8(1):25–34.