axioms

MDPI

Article

Hybrid Fuzzy C-Means Clustering Algorithm, Improving Solution Quality and Reducing Computational Complexity

Joaquín Pérez-Ortega [1,*], Carlos Fernando Moreno-Calderón [1], Sandra Silvia Roblero-Aguilar [2], Nelva Nely Almanza-Ortega [3], Juan Frausto-Solís [4,*], Rodolfo Pazos-Rangel [4] and Alicia Martínez-Rebollar [1]

[1] Tecnológico Nacional de México/Cenidet, Cuernavaca 62490, Mexico; d19ce059@cenidet.tecnm.mx (C.F.M.-C.); alicia.mr@cenidet.tecnm.mx (A.M.-R.)
[2] Tecnológico Nacional de México/IT Tlalnepantla, Tlalnepantla 54070, Mexico; sandra.ra@tlalnepantla.tecnm.mx
[3] Investigadoras e Investigadores por México/Conahcyt, Ciudad de México 03100, Mexico; nelva.almanza@conahcyt.mx
[4] Tecnológico Nacional de México/IT Cd. Madero, Madero 89440, Mexico; rodolfo.pr@cdmadero.tecnm.mx
* Correspondence: joaquin.po@cenidet.tecnm.mx (J.P.-O.); juan.frausto@itcm.edu.mx (J.F.-S.)

Abstract: Fuzzy C-Means is a clustering algorithm widely used in many applications. However, its computational complexity is very large, which prevents its use for large problem instances. Therefore, a hybrid improvement is proposed for the algorithm, which considerably reduces the number of iterations and, in many cases, improves the solution quality, expressed as the value of the objective function. This improvement integrates two heuristics, one in the initialization phase and the other in the convergence phase or the convergence criterion. This improvement was called HPFCM. A set of experiments was designed to validate this proposal; to this end, four sets of real data were solved from a prestigious repository. The solutions obtained by HPFCM were compared against those of the Fuzzy C-Means algorithm. In the best case, reductions of an average of 97.65% in the number of required iterations and an improvement in quality solution of 82.42% were observed when solving the SPAM dataset. Finally, we consider that the proposed heuristics may inspire improvements in other specific purpose variants of Fuzzy C-Means.

Keywords: computational complexity; convergence; efficiency; Fuzzy C-Means; hybrid clustering algorithm; initialization; solution quality

MSC: 62H30; 68W40; 90C70; 91C20

check for updates

Citation: Pérez-Ortega, J.; Moreno-Calderón, C.F.; Roblero-Aguilar, S.S.; Almanza-Ortega, N.N.; Frausto-Solís, J.; Pazos-Rangel, R.; Martínez-Rebollar, A. Hybrid Fuzzy C-Means Clustering Algorithm, Improving Solution Quality and Reducing Computational Complexity. *Axioms* **2024**, *13*, 592. https://doi.org/10.3390/axioms13090592

Academic Editor: Darjan Karabašević

Received: 1 July 2024
Revised: 24 August 2024
Accepted: 25 August 2024
Published: 30 August 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The problem addressed in this research consists of improving the efficiency of the Fuzzy C-Means algorithm without compromising the solution quality. This algorithm has been used successfully in the solution of instances from different domains, for example, image segmentation [1,2], medical applications [3,4], marketing [5], and electric vehicles [6]. However, its computational cost is very high, which prevents its application to large datasets because of its execution time. It is foreseeable that this limitation will tend to increase because, in recent years, data generation has been growing exponentially. An alternative for overcoming this limitation is improving the efficiency of the algorithm. In this sense, the main contribution of this research is a new hybrid variant, which has shown a significant increase in its efficiency through the reduction in the number of iterations for obtaining a solution. The objective of increasing the efficiency of the algorithm was achieved partially due to the study of the proposals by Ruspini [7], Dunn [8], and Bezdek [9] from the optimization perspective. In this way, it was observed that the algorithm could be initialized using a *seed* membership matrix or utilizing a *seed* centroid set. Traditionally, the membership matrix has been used for initialization; however, some advantages have

been found in using centroids instead, where the main one is that it reduces the number of iterations.

Additionally, from the optimization perspective, it is more natural to define the convergence criterion considering the values of the objective value in each iteration; however, a value of the differences of the membership matrix has traditionally been used as a convergence criterion. Experimentally, benefits have been found by using the differences of the objective function values as a convergence criterion; one of them is that this reduces the number of iterations, and the other is that it allows associating a criterion of solution quality.

This article is organized into six sections: Sections 2 and 3 describe the clustering algorithm and the related work. Section 4 presents the FCM algorithm and the new heuristics called HOFCM and HPFCM. Section 5 shows the results obtained from experimental tests. Finally, conclusions are presented in Section 6.

2. Clustering Algorithms

Clustering algorithms have been classified into hierarchical and partitional [10,11]; the most popular of the former K-means [12,13], while Fuzzy C-Means (FCM) [9] is the most representative of partitional clustering algorithms. One of the differences between FCM and K-means is that atypical data does not significantly affect its results. However, a disadvantage of FCM with respect to K-Means is its computational complexity [14]. Formally, the clustering problem was formulated as follows: let $X = \{x_1, \ldots, x_n\}$ be the set of data of n objects that should be partitioned, where $x_i \in \Re^d$ for $i = 1, \ldots n$, and d is the number of attributes (or dimensions) of each object x_i, and c is the number of clusters in the range $[2, n]$.

For the first time, the fuzzy clustering problem was formulated as an optimization problem that minimizes an objective function [7]. Later, this formulation was improved by Dunn [8] and Bezdek [9]. Next, Bezdek's formulation is presented [9]. Equation (1) shows the objective function J_m. The decision variables are matrix U of membership degrees and the set of centroids V.

$$P: \text{minimize } J_m(U, V) = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{m}||x_i - v_j||^2 \quad (1)$$

Subject to

$$u_{ij} \in [0,1] \qquad 1 \le i \le n, \quad 1 \le j \le c \quad (2)$$

$$\sum_{j=1}^{c} u_{ij} = 1 \qquad 1 \le i \le n \quad (3)$$

$$u_{ij} = 1 \qquad \text{if } x_i = v_j \quad (4)$$

$$0 < \sum_{i=1}^{n} u_{ij} < n \qquad 1 < j \le c \quad (5)$$

Next, the notation necessary for the article is defined:
i: index of object;
j: index of cluster;
m: weighting exponent or fuzzy factor, $m > 1$;
c: number of clusters in X;
n: number of objects in X;
ε: threshold value;
t: iteration count;
$X = \{x_1, \ldots, x_n\}$: set of n objects to be partitioned according to a similarity criterion;
$U = \{u_{ij}\}$: membership degree of object i to cluster j;
$V = \{v_1, \ldots, v_c\}$: set of centroids, where v_j is the centroid of cluster j;
T: limit number of iterations;

ΔU: convergence criterion (the difference of the membership degrees values in two consecutive iterations)

ΔV: convergence criterion (the difference of the centroids' values in two consecutive iterations);

$\Delta_{old}J_m$: convergence criterion (the difference of the objective function values in two consecutive iterations);

$\Delta_{new}J_m$: new convergence criterion (the difference of the objective function values, in two consecutive iterations, expressed as a percentage of its value in the next to the last one).

$||x_i - v_j||^2$: distance from object x_i to centroid v_j using the Euclidean distance.

The solution method mentioned by Bezdek in [9] consists of performing the calculation of Equations (6) and (7) alternately until satisfying a convergence criterion.

$$v_j = \frac{\sum_{i=1}^{n} (u_{ij})^m x_i}{\sum_{i=1}^{n} (u_{ij})^m} \qquad 1 \leq j \leq c \tag{6}$$

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{||x_i - v_j||}{||x_i - v_k||} \right)^{\frac{2}{(m-1)}}} \qquad 1 \leq i \leq n, 1 \leq j \leq c \tag{7}$$

Equations (8) and (9) are two convergence criteria [9] applied when a stop condition is met. Traditionally, the threshold values are indicated with the symbol ε.

$$\Delta U^{(t)} = \max\left(abs\left({u_{ij}}^{(t-1)} - {u_{ij}}^{(t)} \right) \right) \tag{8}$$

$$\Delta V^{(t)} = \max\left(abs\left({v_{ij}}^{(t-1)} - {v_{ij}}^{(t)} \right) \right) \tag{9}$$

Algorithm FCM has four phases: initialization, centroid calculation, classification, and convergence [15]. For the initialization phase, the following parameters should be defined: the number of clusters c, the value of the weighting exponent m, the matrix of membership degrees U, the value of the convergence threshold ε, and the limit number of iterations.

Since the main contributions of this article are improvements in the initialization and convergence phases, the following subsections will describe the initialization and convergence approaches by other authors.

3. Related Work

This section presents the most relevant works related to this research. First, studies focusing on the algorithm's initialization method are examined. Next, investigations centered on the convergence method are reviewed.

3.1. Initialization Methods

The general method for solving the fuzzy clustering problem consists of initializing the algorithm using the membership matrix or a set of centroids. Specifically, Bezdek proposed a method for initializing the membership matrix [16]. Many of the implementations of FCM and its variants continued to use the aforesaid method. In contrast, this article proposes centroids in the initialization phase because several advantages were experimentally observed in the algorithm behavior. In order to give an idea of the different initialization methods proposed by other authors, Table 1 is presented; its structure is the following: column one shows the publication reference, column two indicates if the initialization was performed by calculating the membership matrix using a heuristic method, column three specifies if a set of randomly generated centroids was used in the initialization, and column four indicates if the initialization was performed using centroids generated using a heuristic method.

Table 1 shows that, in the pioneering research works, FCM initialization was performed using the membership degrees matrix [7,9,16]. Only a few variants have used a sample of randomly selected objects as the set of initial centroids [17,18]. In [19], the proposed approach consisted of dividing the objects' space into a grid and finding high-density regions for selecting the objects that would be the initial centroids. In [20,21], the FCM++ algorithm was proposed, which used the K++ heuristics [21] to select the objects that would perform the initial centroid role. Later, an improvement of K++ was proposed in [22]. In [23], the final centroids generated by the K-Means algorithm were used, which, through a transformation process called One-Hot encoding, generated the initial membership matrix for FCM. Finally, in [24], the HOFCM algorithm was proposed, which uses as initial centroids those to which the OK-Means algorithm converges [25]. It is worth mentioning that the HOFCM algorithm, in its experimental phase, was compared to three algorithms [16,20,23] and outperformed them in all the test cases, which involved large datasets. HOFCM reduced the execution time by up to 93% and improved the solution quality by up to 68.12% in the best case.

Table 1. Different methods for initializing the Fuzzy C-Means algorithm.

Reference	*U* Heuristic	*V* Random	*V* Heuristic
[7]	Ruspini		
[8]	Dunn		
[9,15,16]	Bezdek		
[17,18]		Method 1	
[19]			Method 2
[20,21]			K++
[20,21]			Variant K++
[23]			K-Means
[24]	OKMeans++		

3.2. Convergence Methods

In the convergence phase of FCM, several stop criteria have been used. Figure 1 shows four convergence criteria published in [18]: (1) the largest difference of the membership degree values in two consecutive iterations, (2) the largest distance between centroid positions in two consecutive iterations, (3) the difference of the objective function in two successive iterations, and (4) the limit of the number of iterations. Some implementations use a combination of a limit in the number of iterations and the largest difference of the membership degree values in two consecutive iterations [16,26]. We remark that the most used criterion is the first one. Table 2 presents different convergence methods.

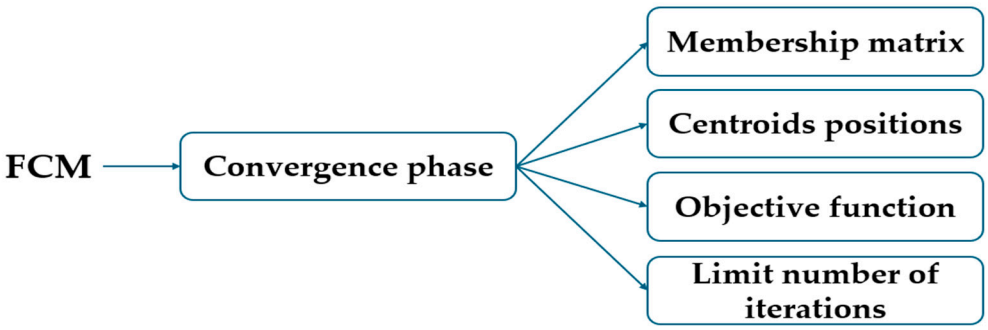


Figure 1. Convergence or stop criteria of the Fuzzy C-Means algorithm.

Table 2. Different methods for the convergence phase in the Fuzzy C-Means algorithm.

Reference	ΔU	ΔV	J_m	T
[7]	✓			
[8]		✓		
[9]	✓	✓		
[16,26]	✓			✓
[17,27–31]			✓	
[18]			✓	

The structure of Table 2 is the following: column one shows the publication reference, column two indicates if the difference of the membership degrees values in two consecutive iterations (ΔU) was used, column three specifies if the difference of centroid values in two consecutive iterations (ΔV) was utilized, column four indicates if the difference of the objective function (J_m) values in two consecutive iterations was used, and the last column specifies if a limit number of iterations (T) was used as stop criterion.

Table 2 shows that, at the beginning of the FCM algorithm, ΔU and ΔV were used as convergence criteria [7–9]. A combination of ΔU and T was used in [16,26]. Additionally, in the specialized literature, different improvements of the algorithm have been published, which use the objective function as the stop criterion [17,27–31]. It is worth mentioning that none of the improvements mentioned in Table 2, in the experimental part, report an improvement in the solution quality, expressed as the objective function value.

Hybrid methods [24] can generate better results because they can overcome the combined algorithms' disadvantages. Therefore, a hybrid improvement for the FCM algorithm is proposed, which significantly reduces the number of iterations and, in many cases, improves the solution quality in terms of the objective function value. Our proposal consists of combining the approaches of two different algorithms. From the first one, it adopts its initialization strategy [24]. From the second one, it adopts its convergence strategy [18].

4. Materials and Methods

The new algorithm that we propose in this article is inspired by three algorithms: FCM [16], HOFCM [24], and P-FCM [18]. These algorithms will be described in the following sections, emphasizing the section related to this research.

4.1. FCM Algorithm

The first formal approach for the fuzzy clustering problem as an optimization problem was proposed by Ruspini in [7]. In 1973, Dunn [8] extended the meaning of hard clustering to preliminary concepts of fuzzy means. Then, Bezdek [9] generalized Dunn's approach for obtaining an infinite family of Fuzzy C-Means algorithms.

The pseudocode of the standard FCM algorithm [16] is shown in the Table 3.

Table 3. FCM algorithm.

Input: X, c, m, ε, T
Output: V, U
Initialization:
  t: = 0;
  Calculate membership matrix $U^{(t)}$: using Bezdek's method;
Calculate centroids:
  Calculate the centroids using Equation (6);
Calculate membership matrix:
  Update and calculate the membership matrix using Equation (7);
Convergence:
  If $\Delta U^{(t)} < \varepsilon$ or $t = T$:
    Stop the algorithm;
  Otherwise:

Table 3. *Cont.*

12	$U^{(t)}$: = $U^{(t+1)}$ and t: = t + 1;
13	Go to **Calculate centroids;**
14	**End of algorithm**

The FCM algorithm has the following input parameters: the dataset X, the number of clusters c, the weighting exponent m, the threshold value ε, and the limit number of iterations T. The output parameters or decision variables are the matrix that contains the membership degrees U and the set of centroids V. In the initialization phase, the iteration counter t is set to 0, the threshold value ε is set to 0.01, and the initial matrix of membership degrees is generated using Bezdek's random method [16]. The cycle starts at line 4, where the centroids are calculated, and at line 6, the membership matrix calculation is performed using Equations (6) and (7), respectively. Notice that line 9 defines a convergence criterion constituted by two stop conditions, such that the algorithm stops when anyone is satisfied.

4.2. HOFCM Algorithm

The HOFCM algorithm [24] introduces an improvement in the initialization phase of the standard FCM algorithm. The pseudocode of the HOFCM algorithm is shown in the Table 4.

Table 4. HOFCM algorithm.

Input: X, c, m, ε, T	
Output: V, U	
1	**Initialization:**
2	ϵok: = threshold value for determining O-K-Means convergence;
3	Assign the value for c;
4	Function K++ (X, c):
5	Return V';
6	Function O-K-Means (X, V', ϵok, c):
7	Return V'';
8	Function S (V''):
9	Return U;
10	Determine the value of the threshold ε to determine the convergence of the algorithm;
11	Assign the value for m;
12	t: = 0;
13	**Calculate centroids:**
14	Calculate the centroids using Equation (6);
15	**Classification:**
16	Update and calculate the membership matrix using Equation (7);
17	**Convergence:**
18	**If** $\Delta U^{(t)} < \varepsilon$ or $t = T$:
19	Stop the algorithm;
20	**Otherwise**:
21	$U^{(t)}$: = $U^{(t+1)}$ and t: = t + 1;
22	Go to **Calculate centroids**

The HOFCM algorithm consists of four phases. The first phase is the initialization phase, during which three functions are created: (1) The K++ function, which takes the dataset X and the number of clusters as input parameters; this function returns a set of centroids denoted by the variable V'. (2) The O-K-Means function, which receives dataset X, the set of centroids V' generated by K++, and the threshold value ϵok as input parameters to determine the algorithm's convergence and the number of clusters; at the end of the convergence phase, it returns optimized centroids defined by the variable V''. (3) The transformation function s, which transforms the final centroids from the O-K-Means algorithm—selected based on the lowest value of the objective function—into the initial optimized values of the membership matrix; these values serve as input parameters for the

FCM algorithm. The second, third, and fourth phases of the HOFCM algorithm, called calculate centroids, classification, and convergence, respectively, are typical of the standard FCM algorithm.

4.3. P-FCM Algorithm

The P-FCM algorithm [18] is a heuristic that modifies the convergence phase of the FCM algorithm. The convergence strategy evaluates the difference of the objective function in two consecutive iterations, expressed as a percentage; when the value is less than a defined threshold, the algorithm converges. The pseudocode of the P-FCM algorithm is shown in Table 5.

Table 5. P-FCM algorithm.

Input: X, V, c, m, ε	
Output: V, U	
1	**Initialization:**
2	t: = 0;
3	$U^{(t)}$: = $\{u_{11}, \ldots, u_{ij}\}$; is randomly generated;
4	**Calculate centroids:**
5	Calculate the centroids using Equation (6);
6	**Calculate membership matrix:**
7	Update and calculate the membership matrix using Equation (7);
8	**Convergence:**
9	**If** $\Delta_{new}{J_m}^{(t)} \leq \varepsilon$:
10	Stop the algorithm;
11	**Otherwise**:
12	$U^{(t)}$: = $U^{(t+1)}$ and t: = t + 1;
13	Go to **Classification**
14	**End of algorithm**

Line 8 shows the convergence phase, which includes the stop indicator of the P-FCM algorithm. In this phase, the value of indicator $\Delta_{new}J_m$ is calculated, which expresses as a percentage the difference of the objective function in two consecutive iterations (see Equation (10)). The algorithm stops when this value is less than or equal to the convergence threshold ε.

$$\Delta_{new}{J_m}^{(t)} = 100 * \frac{\left({J_m}^{(t-1)} - {J_m}^{(t)}\right)}{{J_m}^{(t-1)}} \tag{10}$$

4.4. HPFCM Algorithm

It is known that the efficiency of the FCM algorithm depends, among other factors, on the initialization and convergence strategies. In this research, a new algorithm, HPFCM, is proposed. This algorithm combines two different heuristics, which considerably reduces the number of iterations and improves the solution quality in many cases. From the first heuristics, it adopts its initialization strategy [24]. From the second, it adopts its convergence strategy [18].

The pseudocode that incorporates the proposed heuristics, called HPFCM, is shown in Table 6.

Algorithm HPFCM, like FCM, consists of four phases. In the initialization phase, a strategy borrowed from the HOFCM heuristics [23], three functions are executed: (1) Function K++ has as input parameters the dataset X and the number of clusters; additionally, this function has as output parameter the set of centroids, denoted by variable V'. (2) Function O-K-Means has as input parameters the dataset X, the set of centroids V', the value of the convergence threshold ϵok, and the number of clusters; furthermore, this function has as output parameter the set of centroids, symbolized by variable V''. (3) Function S has as input parameter the set of centroids, signified by variable V''; in addition, this function has as output parameter a membership matrix, calculated from V''. The second and

third phases (called centroid calculation and classification, respectively) are phases from the standard FCM. The convergence phase is inspired by the P-FCM heuristics [18].

Table 6. HPFCM algorithm.

Input: X, c, m, ε, T
Output: V, U
Initialization:
  ϵok: = threshold value for determining O-K-Means convergence;
  Assign the value for c;
  Function K++ (X, c):
    Return V';
  Function O-K-Means (X, V', ϵok, c):
    Return V'';
  Function S (V''):
    Return U;
  Determine the value of the threshold ε to determine the convergence of the algorithm;
  Assign the value for m;
  t: = 0;
Calculate centroids:
  Calculate the centroids using Equation (6);
Classification:
  Update and calculate the membership matrix using Equation (7);
Convergence:
  If $\Delta_{new} J_m{}^{(t)} \leq \varepsilon$:
    Stop the algorithm;
  Otherwise:
    $U^{(t)} := U^{(t+1)}$ and $t := t + 1$;
  Go to **Classification**
End of algorithm

5. Results

Four experiments were carried out to evaluate our proposal, where four real datasets were solved. Each dataset was solved using the FCM, HOFCM, and HPFCM algorithms. The standard FCM algorithm was implemented so that other results could be compared against those from this algorithm; the HOFCM algorithm was implemented because it includes an improvement change in the initialization phase of FCM. Several executions with different configuration parameters, such as different threshold values, were performed. Altogether, 100 executions of the algorithms were performed.

5.1. Experiments Setup

The three were implemented in C language, and the GCC 7.4.0 compiler was used. The experiments were executed on an Acer Nitro 5 laptop with Intel Core i7-11800H processor at 2.30 GHz, 16 GB of RAM memory, 512 GB SSD and 1 TB HDD, and Windows 11 operating system.

5.1.1. Description of Datasets

The datasets were obtained from the UCI Machine Learning Repository [32]. Table 7 shows the characteristics of the real datasets. The structure of Table 7 is as follows: column one shows the dataset name, column two presents the number of objects, column three shows the number of attributes, and column four presents the product of columns two and three.

The first two datasets were selected because they have been used as test datasets in publications on clustering algorithms. The subsequent two datasets were chosen because other authors have used them for experimenting with large datasets and on parallel and distributed hardware.

Table 7. Description of real datasets.

X	n	d	$n \times d$
WDBC	569	30	17,070
ABALONE	4177	7	29,239
SPAM	4601	57	262,257
URBAN	360,177	2	720,354

5.1.2. Parameter Configurations for Algorithm Execution

Test cases were designed for evaluating different values of the objective function and the number of iterations carried out by the algorithms. The parameters for executing the algorithms were the following:

FCM: $c = 50$, $m = 2$, $T = 500$, and $\varepsilon = 0.01$;

HOFCM: $c = 50$, $m = 2$, $\varepsilon ok = 0.72$, and $\varepsilon = 0.01$;

HPFCM 0.9: $c = 50$, $m = 2$, $\epsilon ok = 0.72$, and $\varepsilon = 0.9$;

HPFCM 0.22: $c = 50$, $m = 2$, $\varepsilon ok = 0.72$, and $\varepsilon = 0.22$;

HPFCM 0.06: $c = 50$, $m = 2$, $\varepsilon ok = 0.72$, and $\varepsilon = 0.06$.

For FCM the values of the initial membership matrix were obtained using Bezdek's method. For HPFCM, the values of the threshold value ε were three for each experiment: $\varepsilon = 0.9$, $\varepsilon = 0.22$ and $\varepsilon = 0.06$. These values were proposed in [18] from a series of experiments with FCM. The value for $\varepsilon = 0.9$ is based on the Pareto Principle and constitutes a balance between the convenient values of the objective function and the number of iterations. The values $\varepsilon = 0.22$ and $\varepsilon = 0.06$ are associated with objective function values, which are 98% and 99% similar (i.e., equivalent to differences of 2% and 1%, respectively) to the objective function value when the algorithm has a convergence threshold of 0.00001.

5.2. Experimental Results

Each experiment consisted of executing FCM, HOFCM, and HPFCM five times because this is a representative sample in previous publications. In experiments I, II, III, and IV, the *WDBC*, *ABALONE*, *SPAM*, and *URBAN* datasets were solved respectively.

Tables 8–11 show the results from the five executions of each algorithm and configuration for each dataset. The structure of the tables is explained next. Column one indicates the execution number. Column two shows the results from FCM, which are subdivided into the total of iterations and the final value of the objective function. Column three presents the results from HOFCM, which are subdivided like those of column two. Columns four, five, and six show the results from HPFCM for $\varepsilon = 09$, $\varepsilon = 0.22$, and $\varepsilon = 0.06$, respectively, and additionally, these results are subdivided similarly as those of column two.

Table 8. Results from experiment I, dataset *WDBC*.

E	FCM		HOFCM		HPFCM 0.9		HPFCM 0.22		HPFCM 0.06	
	t	OF	t	OF	t	OF	t	OF	t	OF
1	107	749,256.98	102	462,581.65	4	481,344.46	10	470,369.65	17	466,814.16
2	92	764,026.31	54	465,843.52	4	486,051.21	8	480,181.55	14	476,482.08
3	81	831,151.69	144	454,661.28	4	498,014.16	11	485,386.93	32	467,050.78
4	107	771,827.15	25	454,679.38	4	469,648.82	11	455,658.05	13	455,164.07
5	91	745,695.01	54	458,615.38	4	475,056.66	9	462,925.54	12	461,856.56
Average	**95.60**	**772,391.43**	**75.80**	**459,276.24**	**4.00**	**482,023.06**	**9.80**	**470,904.34**	**17.60**	**465,473.53**

Table 9. Results from experiment II, dataset *ABALONE*.

E	FCM		HOFCM		HPFCM 0.9		HPFCM 0.22		HPFCM 0.06	
	t	OF	*t*	OF	*t*	OF	*t*	OF	*t*	OF
1	198	4.89	158	4.96	4	5.29	11	5.14	27	5.03
2	289	4.84	177	4.82	4	5.06	7	5.03	27	4.90
3	157	4.84	128	4.92	4	5.06	6	5.04	15	4.99
4	187	4.90	115	4.92	4	5.08	9	5.00	16	4.97
5	317	4.84	164	4.91	5	5.19	12	5.08	30	4.99
Average	**229.60**	**4.86**	**148.40**	**4.91**	**4.20**	**5.14**	**9.00**	**5.06**	**23.00**	**4.98**

Table 10. Results from experiment III, dataset *SPAM*.

E	FCM		HOFCM		HPFCM 0.9		HPFCM 0.22		HPFCM 0.06	
	t	OF	*t*	OF	t	OF	t	OF	t	OF
1	327	15,512,347.05	66	2,594,609.86	5	2,743,216.21	7	2,729,787.58	17	2,672,582.12
2	246	15,346,906.74	47	2,567,788.11	4	2,641,992.34	9	2,600,034.52	20	2,576,467.98
3	433	14,389,142.01	19	2,738,174.19	4	2,800,149.19	10	2,746,694.76	13	2,741,455.53
4	326	15,663,485.28	35	2,563,461.92	7	2,658,812.92	10	2,619,088.31	12	2,615,393.81
5	459	15,254,377.61	31	2,623,242.90	4	2,700,152.82	6	2,688,707.75	19	2,638,046.71
Average	**358.20**	**15,233,251.74**	**39.60**	**2,617,455.40**	**4.80**	**2,708,864.70**	**8.40**	**2,676,862.58**	**16.20**	**2,648,789.23**

Table 11. Results from experiment IV, dataset *URBAN*.

E	FCM		HOFCM		HPFCM 0.9		HPFCM 0.22		HPFCM 0.06	
	t	OF	*t*	OF	*t*	OF	*t*	OF	*t*	OF
1	78	6386.83	32	4629.85	4	4867.73	6	4845.09	22	4636.03
2	95	5819.01	33	4681.14	4	5269.01	6	5238.33	31	4681.70
3	137	6304.24	33	4858.13	3	5038.73	6	4999.36	19	4868.28
4	102	5468.40	30	4562.42	4	4734.64	6	4715.66	17	4565.79
5	105	5974.76	36	4694.16	6	4816.80	8	4795.37	17	4701.21
Average	**103.40**	**5990.65**	**32.80**	**4685.14**	**4.20**	**4945.38**	**6.40**	**4918.76**	**21.20**	**4690.60**

Four graphs were generated, one for each experiment, where the x-axis and y-axis represent the number of iterations and the objective function value. Each point on the graph represents the solution obtained by one of the three algorithms. The values on x and y were normalized in the range [0, 1] using the max-min method to observe the solution of each algorithm and compare them.

5.3. Analisis of Results

The average values of the HPFCM and FCM algorithms were compared based on the results obtained. For quantifying the differences between the results of the algorithms, two indices, α and β, were defined, where α expresses the reduction in the number of iterations of HPFCM with respect to FCM as a percentage of the FCM value (see details in Equation (11)). Index β expresses the reduction of the objective function of HPFCM with respect to FCM as a percentage (see Equation (12)). Notably, the FCM algorithm converged in all experimental cases when the stop criteria $\Delta U(t) < \varepsilon$ was met.

$$\alpha = 100 - \left(100 * \frac{average_t_HPFCM}{average_t_FCM}\right) \tag{11}$$

$$\beta = 100 - \left(100 * \frac{average_OF_HPFCM}{average_OF_FCM}\right) \tag{12}$$

5.3.1. Analysis of the Results of Experiment I

Based on the experimental results shown in Table 8 and Figure 2, the following observations were obtained:

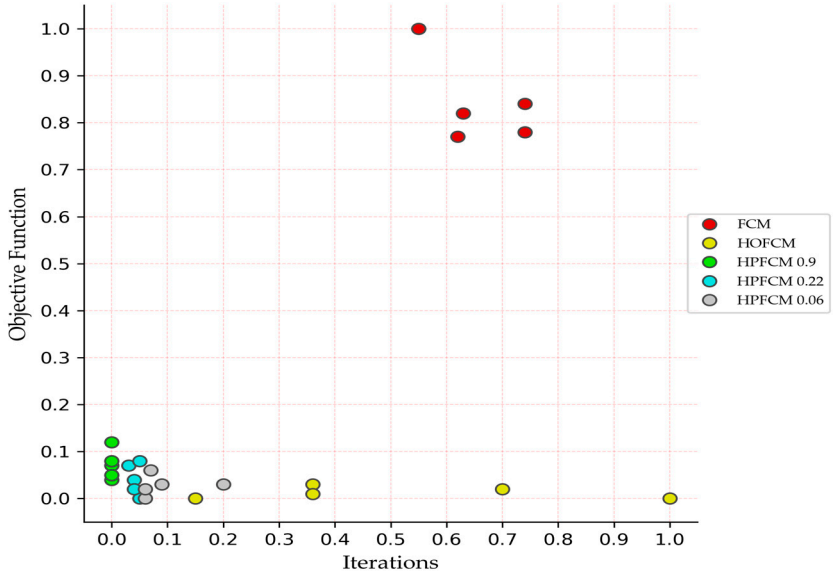


Figure 2. Distribution of normalized solutions for dataset *WDBC*.

(a) FCM obtained an average of 772,391.43 for the objective function, whereas HPFCM 0.06 obtained 465,473.53 (see Table 8).

(b) As an be observed in Figure 2, the FCM results (depicted by red circles) have large values of the objective function and number of iterations; in contrast, the HOFCM and HPFCM results (depicted by yellow, green, blue, and gray circles) are near the graph origin. Because the distance among the groups of results is large, it was proved that HPFCM had better performance than FCM. In the best case, it was observed that HPFCM 0.9 obtained a value of $\alpha = 95.82$ and HPFCM 0.06 a value of $\beta = 39.73$.

5.3.2. Analysis of the Results of Experiment II

Based on the experimental results shown in Table 9 and Figure 3, the following observations were determined:

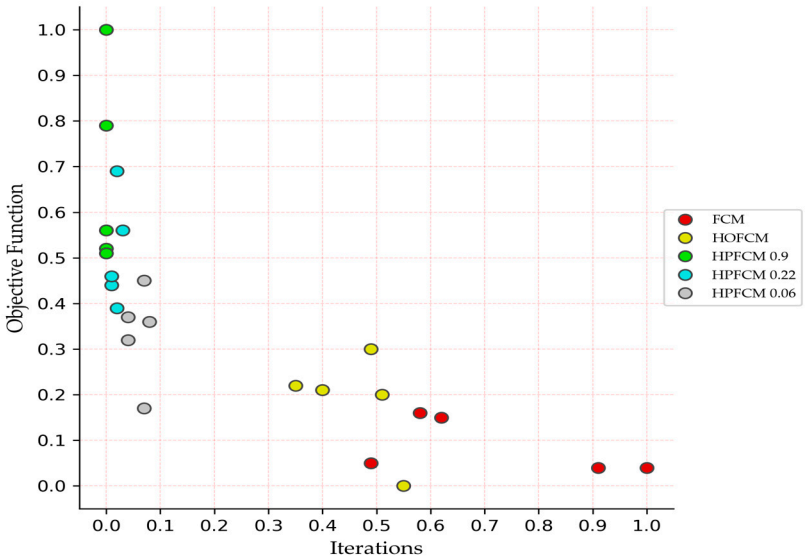


Figure 3. Distribution of normalized solutions for dataset *ABALONE*.

Table 9 shows that FCM obtained the best average value of the objective function. However, the number of iterations was higher with respect to HOFCM. In the best case, HPFCM 0.9 obtained a value $\alpha = 98.17$ and a quality reduction $\beta = 5.76$.

5.3.3. Analysis of the Results of Experiment III

Based on the experimental results shown in Table 10 and Figure 4, the following observations were obtained:

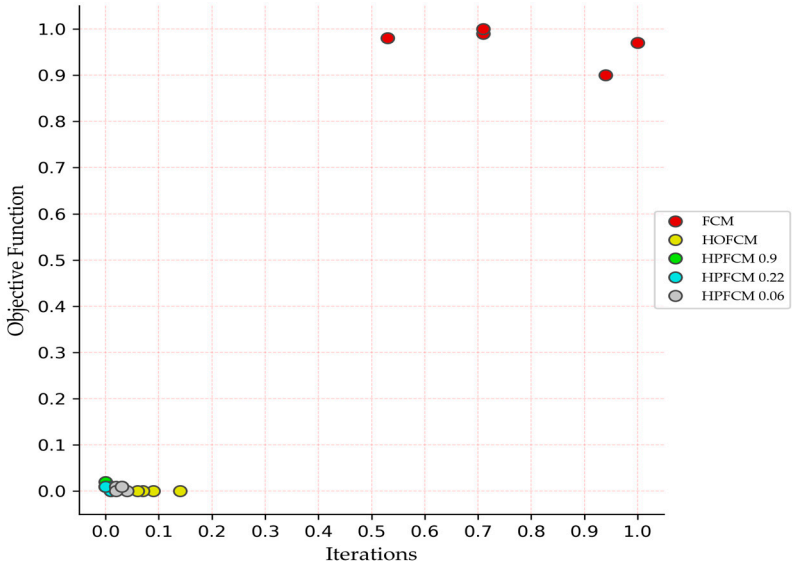


Figure 4. Distribution of normalized solutions for dataset *SPAM*.

Figure 4 shows the results for FCM (depicted by red circles), whereas the values for HOFCM and HPFCM (depicted by yellow, green, blue, and gray circles) are near the graph origin. In the best case, HPFCM 0.9 obtained a value $\alpha = 95.82$, and HPFCM 0.06 attained a value of $\beta = 39.73$. According to these observations, HPFCM performed better than FCM.

5.3.4. Analysis of the Results of Experiment IV

Based on the experimental results that are shown in Table 11 and Figure 5, the following observations were determined:

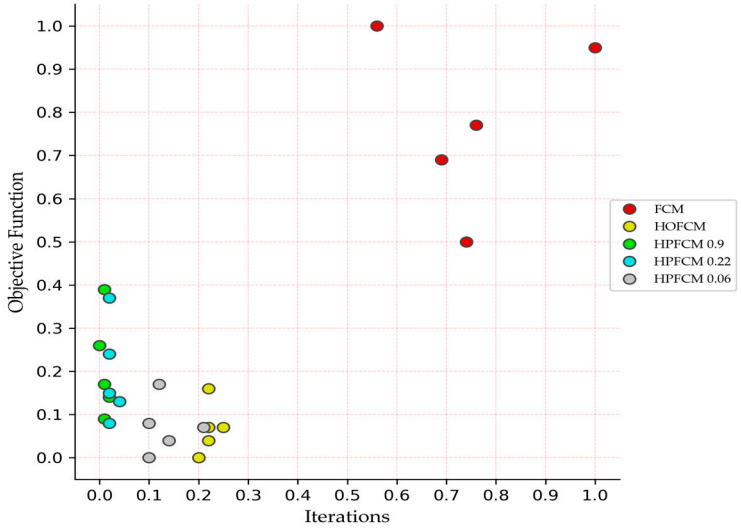


Figure 5. Distribution of normalized solutions for dataset *URBAN*.

FCM obtained an average value of 5990.65 for the objective function and 103.4 iterations on average, whereas HPFCM obtained small values for the objective function and number of iterations. Figure 5 shows that the FCM results (depicted by red circles) for the objective function and number of iterations were the largest, whereas the results for HOFCM and HPFCM (depicted by yellow, green, blue, and gray circles) are close to the graph origin. In the best case, HPFCM 0.9 attained a value of $\alpha = 95.94$, and HPFCM 0.06 obtained a value of $\beta = 21.7$.

6. Conclusions

This article shows that it is possible to increase the efficiency and solution quality of the FCM algorithm using a hybrid approach, which integrates two heuristics: one improves the initialization phase of the algorithm, and the other improves the convergence phase.

To evaluate the HPFCM approach, we performed experiments where *FCM*, *HOFCM*, and *HPFCM* algorithms solved the *WDBC*, *ABALONE*, *SPAM*, and *URBAN* real datasets. Based on the obtained results, it was observed that the HPFCM algorithm generally improved the quality of the solution and needed fewer iterations than the FCM algorithm. The results from experiments II and III are remarkable: in experiment III, the best average results were obtained, and in experiment II, the worst results were attained. In experiment III, when solving the *SPAM* dataset with HPFCM and a value $\varepsilon = 0.22$ as convergence threshold, the following results were obtained: an average improvement of 82.43% in solution quality and an average reduction of 97.65% in the number of iterations with respect to FCM.

In experiment II, when solving the *ABALONE* dataset with HPFCM and a value $\varepsilon = 0.9$ as convergence threshold, the following results were obtained: the algorithm HPFCM obtained a 5.67% loss of quality, which is considered little significant, but remarkably, it achieved a 98.17% reduction in the number of iterations.

It is worth mentioning that the HPFCM algorithm contributes original elements by integrating two heuristics, which have improved the FCM algorithm in the initialization and convergence phases, where the last one has been little studied. Another contribution of our approach is the threshold values used as a convergence criterion. By leveraging such threshold values, users may make informed decisions and prioritize either the execution time or the solution quality. Finally, according to the literature surveyed, we did not find another FCM variant that improves the solution quality, expressed as the value of the objective function as the algorithm converges.

In summary, our proposal contributes to the initialization and convergence phases of the FCM algorithm and has no conflict with improvements in other phases of the algorithm, making possible its integration into other variants of FCM.

Author Contributions: Conceptualization, J.P.-O., C.F.M.-C. and S.S.R.-A.; methodology, J.P.-O., S.S.R.-A. and C.F.M.-C.; software, C.F.M.-C. and S.S.R.-A.; validation, J.P.-O., C.F.M.-C., S.S.R.-A., N.N.A.-O. and A.M.-R.; formal analysis, J.P.-O., A.M.-R. and R.P.-R.; investigation, J.P.-O., C.F.M.-C. and S.S.R.-A.; resources, S.S.R.-A., N.N.A.-O., A.M.-R. and J.F.-S.; data curation, S.S.R.-A., N.N.A.-O., R.P.-R. and C.F.M.-C.; writing—original draft preparation, J.P.-O., C.F.M.-C. and S.S.R.-A.; writing—review and editing, J.P.-O., C.F.M.-C., S.S.R.-A., N.N.A.-O., J.F.-S. and R.P.-R.; visualization, C.F.M.-C.; supervision, J.P.-O.; project administration, J.P.-O.; funding acquisition, J.P.-O., J.F.-S. and A.M.-R. All authors have read and agreed to the published version of the manuscript.

Funding: The Student Carlos Fernando Moreno Calderón acknowledges his scholarship (grantee No. 1000864) to the Consejo Nacional de Humanidades, Ciencia y Tecnología, Mexico.

Data Availability Statement: The real datasets used were obtained from the UCI machine learning repository https://archive.ics.uci.edu/ml/index.php (accessed on 15 January 2024).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ren, T.; Wang, H.; Feng, H.; Xu, C.; Liu, G.; Ding, P. Study on the improved fuzzy clustering algorithm and its application in brain image segmentation. *Appl. Soft Comput.* **2019**, *81*, 105503. [CrossRef]
2. Cardone, B.; Di Martino, F.; Miraglia, V. A Novel Fuzzy-Based Remote Sensing Image Segmentation Method. *Sensors* **2023**, *23*, 9641. [CrossRef]
3. Alashwal, H.; El Halaby, M.; Crouse, J.J.; Abdalla, A.; Moustafa, A.A. The Application of Unsupervised Clustering Methods to Alzheimer's Disease. *Front. Comput. Neurosci.* **2019**, *13*, 31. [CrossRef] [PubMed]
4. Bhimavarapu, U.; Chintalapudi, N.; Battineni, G. Brain Tumor Detection and Categorization with Segmentation of Improved Unsupervised Clustering Approach and Machine Learning Classifier. *Bioengineering* **2024**, *11*, 266. [CrossRef]
5. Yoseph, F.; Ahamed Hassain Malim, N.H.; Heikkilä, M.; Brezulianu, A.; Geman, O.; Paskhal Rostam, N.A. The impact of big data market segmentation using data mining and clustering techniques. *J. Intell. Fuzzy Syst.* **2020**, *38*, 6159–6173. [CrossRef]
6. Nazari, M.; Hussain, A.; Musilek, P. Applications of Clustering Methods for Different Aspects of Electric Vehicles. *Electronics* **2023**, *12*, 790. [CrossRef]
7. Ruspini, E.H. A new approach to clustering. *Inf. Control* **1969**, *15*, 22–32. [CrossRef]
8. Dunn, J.C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybern.* **1973**, *3*, 32–57. [CrossRef]
9. Bezdek, J.C. Fuzzy Mathematics in Pattern Classification. Ph.D. Thesis, Cornell University, Ithaca, NY, USA, 1973.
10. Bonilla, J.; Vélez, D.; Montero, J.; Rodríguez, J.T. Fuzzy Clustering Methods with Rényi Relative Entropy and Cluster Size. *Mathematics* **2021**, *9*, 1423. [CrossRef]
11. Hashemi, S.E.; Gholian-Jouybari, F.; Hajiaghaei-Keshteli, M. A fuzzy C-means algorithm for optimizing data clustering. *Expert Syst. Appl.* **2023**, *227*, 120377. [CrossRef]
12. Bezdek, J.C. *Elementary Cluster Analysis: Four Basic Methods That (Usually) Work*; River Publishers: Gistrup, Denmark, 2022.
13. MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 21 June–18 July 1965.
14. Ghosh, S.; Kumar, S. Comparative Analysis of K-Means and Fuzzy C-Means Algorithms. *Int. J. Adv. Comput. Sci. Appl.* **2013**, *4*, 35–39. [CrossRef]
15. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Plenum Press: New York, NY, USA, 1981.
16. Bezdek, J.C.; Ehrlich, R.; Full, W.E. FCM: The Fuzzy C-Means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203. [CrossRef]
17. Hashemzadeh, M.; Oskouei, A.G.; Farajzadeh, N. New fuzzy C-means clustering method based on feature-weight and cluster-weight learning. *Appl. Soft Comput.* **2019**, *78*, 324–345. [CrossRef]
18. Pérez-Ortega, J.; Moreno-Calderón, C.F.; Roblero-Aguilar, S.S.; Almanza-Ortega, N.N.; Frausto-Solís, J.; Pazos-Rangel, R.; Rodríguez-Lelis, J.M. A New Criterion for Improving Convergence of Fuzzy C-Means Clustering. *Axioms* **2024**, *13*, 35. [CrossRef]
19. Zou, K.; Wang, Z.; Hu, M. An new initialization method for fuzzy c-means algorithm. *Fuzzy Optim. Decis. Mak.* **2008**, *7*, 409–416. [CrossRef]
20. Stetco, A.; Zeng, X.J.; Keane, J. Fuzzy C-means++: Fuzzy C-means with effective seeding initialization. *Expert Syst. Appl.* **2015**, *42*, 7541–7548. [CrossRef]
21. Arthur, D.; Vassilvitskii, S. k-means++: The Advantages of Careful Seeding. In Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, USA, 7–9 January 2007.
22. Liu, Q.; Liu, J.; Li, M.; Zhou, Y. Approximation algorithms for fuzzy C-means problem based on seeding method. *Theor. Comput. Sci.* **2021**, *885*, 146–158. [CrossRef]
23. Wu, Z.; Chen, G.; Yao, J. The Stock Classification Based on Entropy Weight Method and Improved Fuzzy C-means Algorithm. In Proceedings of the 4th International Conference on Big Data and Computing, Guangzhou, China, 10–12 May 2019.
24. Pérez-Ortega, J.; Roblero-Aguilar, S.S.; Almanza-Ortega, N.N.; Solís, J.F.; Zavala-Díaz, C.; Hernández, Y.; Landero-Nájera, V. Hybrid Fuzzy C-Means Clustering Algorithm Oriented to Big Data Realms. *Axioms* **2022**, *11*, 377. [CrossRef]
25. Pérez, J.; Almanza, N.N.; Romero, D. Balancing effort and benefit of K-means clustering algorithms in Big Data realms. *PLoS ONE* **2018**, *13*, e0201874. [CrossRef]
26. Cannon, R.L.; Dave, J.V.; Bezdek, J.C. Efficient implementation of the Fuzzy C-Means clustering algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 248–255. [CrossRef]
27. Wang, X.; Wang, Y.; Wang, L. Improving Fuzzy C-Means clustering based on feature-weight learning. *Pattern Recognit. Lett.* **2004**, *25*, 1123–1132. [CrossRef]
28. Wan, R.; Yan, X.; Su, X. A Weighted Fuzzy Clustering Algorithm for Data Stream. In Proceedings of the 2008 ISECS International Colloquium on Computing, Communication, Control, and Management, Guangzhou, China, 3–4 August 2008.
29. Xue, Z.A.; Cen, F.; Wei, L.P. A Weighting Fuzzy Clustering Algorithm Based on Euclidean Distance. In Proceedings of the 2008 Fifth International Conference on Fuzzy Systems and Knowledge Discovery, Jinan, China, 18–20 October 2008.
30. Pimentel, B.A.; de Souza, R.M.C.R. Multivariate Fuzzy C-Means algorithms with weighting. *Neurocomputing* **2016**, *174*, 946–965. [CrossRef]

31. Du, X. A robust and high-dimensional clustering algorithm based on feature weight and entropy. *Entropy* **2023**, *25*, 510. [CrossRef] [PubMed]
32. UCI Machine Learning Repository, University of California. Available online: https://archive.ics.uci.edu/ml/index.php (accessed on 15 January 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.