

Article

A Novel Fuzzy Entropy-Based Method to Improve the Performance of the Fuzzy C-Means Algorithm

Barbara Cardone ¹ and Ferdinando Di Martino ^{1,2,*}

¹ Department of Architecture, Federico II University of Naples, 80134 Naples, Italy; b.cardone@unina.it

² Interdepartmental Research Center of Research A. Calza Bini, University of Naples Federico II, 80134 Napoli, Italy

* Correspondence: fdimarti@unina.it; Tel.: 0039-081-2538904; Fax.: 0039-081-2538909

Received: 13 March 2020; Accepted: 25 March 2020; Published: 26 March 2020

Abstract: One of the main drawbacks of the well-known Fuzzy C-means clustering algorithm (FCM) is the random initialization of the centers of the clusters as it can significantly affect the performance of the algorithm, thus not guaranteeing an optimal solution and increasing execution times. In this paper we propose a variation of FCM in which the initial optimal cluster centers are obtained by implementing a weighted FCM algorithm in which the weights are assigned by calculating a Shannon Fuzzy Entropy function. The results of the comparison tests applied on various classification datasets of the UCI Machine Learning Repository show that our algorithm improved in all cases relating to the performances of FCM.

Keywords: FCM; weighted FCM; fuzzy entropy; fuzzy entropy function

1. Introduction

The Fuzzy C-means (FCM) algorithm [1,2] is a well-known partitive fuzzy clustering algorithm which adopts the Euclidean metric for calculating distances and detects cluster centers as points. Unlike partitive crisp algorithms like K-means [3], FCM objects are allowed to belong to more than one cluster and handle uncertainty in assigning them to a cluster.

FCM applies an iterative process with the aim of minimizing an objective function in order to detect the cluster centers assigning the membership degrees of the objects to them. As in K-means, the number of clusters is set a priori and initially the membership degrees of the objects to the clusters are randomly assigned. At each iteration the cluster centers and the partition matrix are recalculated, minimizing an objective function. The process ends when the absolute difference between the partition matrix and the one calculated in the previous cycle is less than a prefixed threshold.

The initialization phase significantly affects the algorithm's performances, influencing its accuracy and running time. In fact, based on the initial choice of the partition matrix or clusters, the algorithm can quickly converge towards a global optimum or get trapped in a local optimum.

Many variations of FCM were worked out in order to optimize the initialization phase.

Some authors [4–6] apply the Subtractive Clustering algorithm to find the initial cluster however, this method requires to set a priori the mountain peak and the mountain radius parameters.

In [7], an initialization method based on a density cluster algorithm is proposed.

Recently metaheuristic methods have been proposed in the literature to optimize the initialization of the cluster centers. A Genetic Algorithm (GA) is applied to a kernel FCM algorithm in [8] in order to find the initial cluster centers. In [9], a GA is applied to find the optimal initial cluster centers in image segmentation. In [10], a Particle Swarm Optimization algorithm is applied to find the optimal FCM initial cluster centers for sentiment clustering. Three metaheuristic hybrid FCM

algorithms based, respectively, on Differential Evolution, GA, and Particle Swarm Optimization (PSO) are used in [11] to find the optimal initial cluster centers.

However, the use of metaheuristic algorithms, despite contributing to improve the accuracy of the solutions obtained by running FCM, can produce a heavy increase in computational load.

In this research we propose a hybrid FCM algorithm in which a fuzzy entropy algorithm is applied to initialize the cluster centers.

The aim of this research is to improve FCM in classification problems finding the initial clusters by using a weighted FCM algorithm in which a fuzzy entropy function is calculated for setting the weight's value. This fuzzy entropy function is applied to measuring the fuzziness of an object in assigning it to the clusters.

In the literature some variations of the FCM algorithm that include entropy functions are proposed in order to increase the FCM performances.

In [12], a fuzzy clustering method combining the possibility of C-means clustering [13] and a fuzzy entropy function is proposed in order to make the algorithm robust to noise. A variation of FCM based on information entropy is proposed in [14] in order to find the optimal number of clusters. The authors introduce a weighting parameter in the objective function to adjust the location of cluster centers and reduce the influence of noise. Two entropy-based FCM variations in which an entropy function is applied for feature extraction are proposed in [15,16]. In [16], the entropy weight method is applied to delete features with less information, then, the K-means and the One-Hot encoding algorithms are used to initialize the membership matrix. A new entropy based FCM algorithm is applied in [17] for the segmentation of brain MR images. The objective function is modified adding a Shannon Entropy component applied to redress the uncertainty among the pixels. In [18], the authors apply the maximum entropy method to find the optimal kernel weights in a kernel FCM algorithm. An entropy-based FCM algorithm is applied in [19] to cluster webpages in a PageRank process. The weight of each cluster is set by using an information entropy method, which is then used to adjust the average weight. An improved entropy-based FCM algorithm is applied in [20] to detect epistasis whereby a cross-entropy method is used to calculate the distances.

In this research, we used the De Luca and Termini fuzzy entropy definition given in [21,22] to measure the fuzziness that models the membership degree of an object to the clusters. We applied a weighted FCM algorithm in the initialization phase to optimize the initial choice of the cluster centers whereby the weight assigned to an object was greater the lower its fuzziness. We tested our algorithm on various classification training sets to measure its performances and compare the results with the ones obtained using FCM.

In section 2 the FCM algorithm is presented and in section 3 the Entropy weighted FCM (EwFCM) algorithm is presented. Section 4 shows the results of our experiments. Final considerations are discussed in section 5.

2. FCM Algorithm

Let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbb{R}^n$ be a set of N objects in the n -dimensional space \mathbb{R}^n where $\mathbf{x}_j = (x_{j1}, \dots, x_{jn})$ and $\mathbf{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_C\} \subset \mathbb{R}^n$ be the set of centers of the C clusters. Let \mathbf{U} be the $C \times N$ partition matrix where u_{ij} is the membership degree of the j th object \mathbf{x}_j to the i th cluster.

The FCM algorithm [1,2] is based on the minimization of the following objective function:

$$J(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m d_{ij}^2 = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|\mathbf{x}_j - \mathbf{v}_i\|^2 \quad (1)$$

Where $d_{ij} = \|\mathbf{x}_j - \mathbf{v}_i\|$ is the Euclidean distance between the center \mathbf{v}_i of the i th cluster and the j th object \mathbf{x}_j and $m \in [1, +\infty)$ is the fuzzifier parameter (a constant which affects the membership values and defines the degree of fuzziness of the partition). For $m = 1$, FCM become a Hard C-means clustering whereby the more m tends towards $+\infty$ the more the fuzziness level of the clusters grows.

Considering the following constraints:

$$\sum_{i=1}^C u_{ij} = 1 \quad \forall j \in \{1, \dots, N\} \tag{2}$$

$$0 < \sum_{j=1}^N u_{ij} < N \quad \forall i \in \{1, \dots, C\} \tag{3}$$

and applying the Lagrange multipliers, we obtain the following solutions for Equation (1):

$$\mathbf{v}_i = \frac{\sum_{j=1}^N u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^N u_{ij}^m} \quad i \in \{1, \dots, C\} \tag{4}$$

and

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{2}{m-1}}} \quad i \in \{1, \dots, C\}, j \in \{1, \dots, N\}. \tag{5}$$

An iterative process is proposed in [2] as follows: Initially the membership degrees are assigned randomly and in each iteration the cluster centers are calculated by Equation (4), then the membership degree components are calculated by Equation (5). The iterative process stops at the *t*th iteration when:

$$|\mathbf{U}^{(t)} - \mathbf{U}^{(t-1)}| < \varepsilon \quad i = 1, \dots, C; j = 1, \dots, N \tag{6}$$

where $\varepsilon > 0$ is a parameter assigned a priori to stop the iteration process and

$$|\mathbf{U}^{(t)} - \mathbf{U}^{(t-1)}| = \max_{\substack{i=1, \dots, C \\ j=1, \dots, N}} \{|u_{ij}^{(t)} - u_{ij}^{(t-1)}|\} \quad i = 1, \dots, C; j = 1, \dots, N. \tag{7}$$

A variation of the FCM algorithm is the weighted FCM (wFCM) algorithm in which a weight defines the influence of the object to the solutions.

The objective function in wFCM is the given by:

$$J_w(\mathbf{U}, \mathbf{V}) = \sum_{i=1}^C \sum_{j=1}^N w_j u_{ij}^m d_{ij}^2 = \sum_{i=1}^C \sum_{j=1}^N w_j u_{ij}^m \|\mathbf{x}_j - \mathbf{v}_i\|^2. \tag{8}$$

The partition matrix \mathbf{U} and the cluster centers \mathbf{V} are obtained minimizing this objective function by using the Lagrange multipliers. The solution for the components u_{ij} of the partition matrix are given by Equation (5). The cluster centers are given by:

$$\mathbf{v}_i = \frac{\sum_{j=1}^N w_j u_{ij}^m \mathbf{x}_j}{\sum_{j=1}^N w_j u_{ij}^m} \tag{9}$$

in which the weight w_j provides the degree of influence of the *j*th object to find the cluster centers.

A set of wFCM-based cluster methods was proposed by some researchers (see, for example, [23–26]) in order to reduce the number of objects in massive datasets assigning to an object a weight based on the density of near data [23] and to encode pixel’s local information in image segmentation activities [24,25]. In [26] some variations of wFCM are proposed in order to handle very large data.

Formally, at any cycle the weight w_j assigned to the object \mathbf{x}_j , $j = 1, \dots, N$ is calculated by using a weight function $w(\mathbf{x}_j)$.

The pseudocodes of the FCM (Algorithm 1) and wFCM algorithms (Algorithm 2) are shown below.

Algorithm 1 FCM

-
1. Set m, ε, C
 2. Initialize randomly the partition matrix \mathbf{U}
 3. **Repeat**
 4. Calculate \mathbf{v}_i , $i = 1, \dots, C$ by using Equation (4)
 5. Calculate u_{ij} , $i = 1, \dots, C; j = 1, \dots, N$ by using Equation (5)
 6. **Until** $|\mathbf{U}^{(t)} - \mathbf{U}^{(t-1)}| > \varepsilon$
-

Algorithm 2 wFCM

-
1. Set m, ϵ, c
 2. Initialize randomly the partition matrix U
 3. **Repeat**
 4. Calculate $w_j, j = 1, \dots, N$ by using a weight function $w(x_i)$
 5. Calculate $v_i, i = 1, \dots, C$ by using Equation (9)
 6. Calculate $u_{ij}, i = 1, \dots, C; j = 1, \dots, N$ by using Equation (5)
 7. **Until** $|U^{(t)} - U^{(t-1)}| > \epsilon$
-

FCM can be considered a special case of wFCM where $w_j = 1, j = 1, \dots, N$.

3. The Entropy Weighted FCM Algorithm

De Luca and Termini [21,22] introduce the concept of fuzzy entropy as a measure of the degree of fuzziness of a fuzzy set and they define the properties of any fuzzy entropy.

Let $X = \{(x_s, u_s), s = 1, \dots, S\}$ be a fuzzy set with u_s membership degree of x_s to X . The fuzzy entropy of X is given by:

$$H(X) = K \sum_{s=1}^S h(u_s) \tag{10}$$

where K is a constant positive and $h: [0,1] \rightarrow [0,1]$ is a continuous function monotonically increasing in $[0, 1/2]$ and monotonically decreasing in $[1/2, 1]$, with $h(0) = h(1) = 0$ and $h(u_s) = h(1 - u_s)$.

A simple example of fuzzy entropy function is given by:

$$h(u_s) = \begin{cases} 2u_s & \text{if } u_s < \frac{1}{2} \\ 2(1 - u_s) & \text{if } u_s \geq \frac{1}{2} \end{cases} \tag{11}$$

in which $h(0) = h(1) = 0$ and $h(0.5) = 1$.

De Luca and Termini in [21] propose the following fuzzy entropy function as a measure of fuzziness of a membership degree:

$$h(u_s) = \begin{cases} 0 & \text{if } u_s = 0 \\ -u_s \ln(u_s) - (1-u_s) \ln(1 - u_s) & \text{if } 0 < u_s < 1 \\ 0 & \text{if } u_s = 1 \end{cases} \tag{12}$$

We use Equation (12) to measure the fuzziness in the assigning the object $x_j = (x_{j1}, \dots, x_{jm})$ to the i th cluster, given by:

$$h(u_{ij}) = \begin{cases} 0 & \text{if } u_{ij} = 0 \\ -u_{ij} \ln(u_{ij}) - (1-u_{ij}) \ln(1 - u_{ij}) & \text{if } 0 < u_{ij} < 1 \\ 0 & \text{if } u_{ij} = 1 \end{cases} \tag{13}$$

where u_{ij} is the membership degree of the j th object to the i th cluster.

The mean fuzziness assigned to x_j is given by:

$$H(x_j) = \frac{1}{c} \sum_{i=1}^c h(u_{ij}). \tag{14}$$

$H(x_i)$ is calculated to assign the weight w_j to the j th object in Equation (8). The constant K has been set to the value $1/C$ in order to normalize the mean fuzziness $H(x_i)$.

Due to the existence of the constraint in Equation (2), the maximum fuzziness value $h_{max} \leq 1$ is obtained when the membership degree of the object to each cluster is equal to $1/C$, while the minimum value is 0, and is obtained when the object belongs with a membership degree equal to 1 to a cluster and membership degrees equal to 0 to all others.

Since the relevance of the object must be greater the lower its fuzziness, we assign the following weight value:

$$w_j = 1 - H(x_j). \quad (15)$$

This formula for w_j is due to the choice of a linear dependence of the weight on the fuzziness. Many other choices are possible, as $w_j = 1/H(x_j)$ however, it has the risk of assigning very high weight values to objects with low or evanescent fuzziness among which outliers or noisy objects can be confused since FCM is sensitive to the presence of noise and outliers in the data, the risk of increasing the influence of noise and outliers in the detection of the final cluster may not be negligible.

We define the clustering mean fuzziness \bar{H} given by:

$$\bar{H} = \frac{1}{N} \sum_{j=1}^N H(x_j). \quad (16)$$

\bar{H} take values in $[0, h_{max}]$ and is initially high, but its value is reduced along the iterations, in which it is given more importance to objects closer to the cluster centers.

The initial cluster centers are found when absolute difference between the mean fuzziness calculated in the current cycle and the one calculated in the previous cycle is below a fixed threshold η or when the number of iterations is equal to a maximal number of iterations i_{max} .

The EwFCM algorithm (Algorithm 3) is schematized below:

Algorithm 3 EwFCM

1. Set m, ε, c
 2. Initialize randomly the partition matrix \mathbf{U}
 3. $n_{iter} := 1$ // number of iterations
 4. **Repeat**
 5. Calculate $w_j, j = 1, \dots, N$ using Equation (15)
 6. Calculate $v_i, i = 1, \dots, C$ by using Equation (9)
 7. Calculate $u_{ij}, i = 1, \dots, C; j = 1, \dots, N$ by using Equation (5)
 8. **Until** $|\bar{H}^{(t)} - \bar{H}^{(t-1)}| > \eta$ OR $n_{iter} = i_{max}$
 9. **Repeat**
 10. Calculate $v_i, i = 1, \dots, C$ by using Equation (4)
 11. Calculate $u_{ij}, i = 1, \dots, C; j = 1, \dots, N$ by using Equation (5)
 12. $n_{iter} := n_{iter} + 1$
 13. **Until** $|\mathbf{U}^{(t)} - \mathbf{U}^{(t-1)}| > \varepsilon$
-

We tested the EwFCM algorithm on two well-known datasets of the UCI Machine Learning Repository (<http://archive.ics.uci.edu/ml/datasets.html>), Iris and Wine, comparing the performance with those obtained by executing the FCM algorithm. In order to make complete classification performance assessments we measured the accuracy, precision, recall (or sensitivity), and F1 score indexes where F1 score is the weighted average of precision and recall.

The four indexes are given by [27,28]:

$$\text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (17)$$

$$\text{precision} = \frac{TP}{TP+FP} \times 100 \quad (18)$$

$$\text{recall} = \frac{TP}{TP+FN} \times 100 \quad (19)$$

$$\text{F1 score} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \times 100 \quad (20)$$

where:

- TP (True Positive) is the number of data correctly assigned to the class;
- TN (True Negative) is the number of objects correctly not assigned to the class;
- FP (False Positive) is the number of objects wrongly assigned to the class;
- FN (False Negative) is the number of objects wrongly not assigned to the class.

To measure the four indices, each fuzzy cluster is labeled with the name of the class in the training dataset to which the largest number of objects assigned to the cluster belong.

4. Simulation Results

We compared the performance of FCM and EwFCM on over 40 datasets of the UCI machine learning repository. The fuzzifier parameter m was set to 2 and the end iteration parameter ε was set to 0.01. Furthermore, by means of suitable calibration performed on various datasets by varying the value of the parameters η and i_{max} , the entropy threshold parameter was set to 0.05 and the maximum number of cycles was set to 50. For our experiments, we used an Intel core I5 3.2 GHz processor. The FCM algorithm was executed 10 times and the performance results obtained by running EwFCM were compared with the best performance results obtained by running FCM.

For brevity we present the complete results obtained for two datasets: Iris and Wine. The Iris dataset was composed of 150 objects with four numerical features describing three types of iris flowers: Iris setosa, Iris virginica, and Iris versicolor. Although Iris setosa is linearly separable from the other two classes, Iris virginica and Iris versicolor are not linearly separable from each other.

We executed the two algorithms setting the number of clusters C to 3.

In Tables 1 and 2 show the results obtained by executing, respectively, FCM and EwFCM. The four metrics are calculated for each class. The last column shows, for each metric, the average value among those calculated for each class.

Table 1. Dataset Iris: Fuzzy C-means clustering algorithm (FCM) results.

Index	Iris Setosa	Iris Versicolor	Iris Virginica	Mean
Accuracy	100.00%	92.67%	92.67%	95.11%
Precision	100.00%	86.79%	91.49%	92.76%
Recall	100.00%	92.00%	86.00%	92.67%
F1 Score	100.00%	89.32%	89.66%	92.66%

Table 2. Dataset Iris: EwFCM results.

Index	Iris Setosa	Iris Versicolor	Iris Virginica	Mean
Accuracy	100.00%	94.67%	94.67%	96.44%
Precision	100.00%	90.38%	93.75%	94.71%
Recall	100.00%	94.00%	90.00%	94.67%
F1 Score	100.00%	92.16%	91.84%	94.66%

As the two tables show, the mean values of all the performance metrics obtained by running the EwFCM algorithm were higher than those obtained by running the FCM algorithm. In particular, all the metrics calculated by using EwFCM for the classes Iris versicolor and Iris virginica were better than the correspondent ones obtained by using the FCM algorithm.

Table 3 shows the results of the comparison of the EwFCM with other FCM variations aimed at optimal search of the initial cluster centers. Table 3 shows the mean accuracy, precision, and recall obtained by using FCM, EwFCM, Density-based FCM [7] Kernel-based FCM [8], and PSO FCM [10].

Table 3. Dataset Iris: Comparison with other FCM-based algorithms.

Index	FCM	EwFCM	Density	Kernel	PSO
Accuracy	95.11%	96.44%	96.05%	96.00%	96.44%
Precision	92.76%	94.71%	94.19%	94.07%	94.67%
Recall	92.67%	94.67%	94.15%	94.04%	94.67%
F1 score	92.66%	94.66%	94.12%	94.00%	94.67%

These results show that the best results were obtained by using EwFCM and PSO. The values of accuracy, precision, recall, and F1 score obtained running the two algorithms were very similar.

To evaluate the computational cost, we measured the number of iterations and running time. Table 4 shows the number of iterations and running time measured using FCM, EwFCM Density FCM, Kernel FCM, and PSO FCM. The number of iteration and the running time calculated using FCM were the minimum and average values obtained by executing the algorithm 10 times.

Table 4. Dataset Iris: Number of iterations and running time comparison.

Method	Iterations	Running Time (s)
FCM Min	16	0.13
FCM Mean	16.7	0.14
EwFCM	8	0.11
Density FCM	9	0.11
Kernel FCM	10	0.12
PSO FCM	8	0.13

These results show that EwFCM had a better computational cost than FCM. While the number of iterations obtained by running EwFCM was approximately half that obtained by running the FCM algorithm, the running time was slightly less. This happens in consideration of the time taken in the initialization phase by the EwFCM algorithm. EwFCM and PSO FCM provided the least number of iterations but the running time in PSO FCM was slightly longer than in EwFCM, as PSO FCM took longer than EwFCM to complete the initialization phase.

Now we present the results of the comparison tests performed on the wine dataset.

The wine dataset was given by 178 objects with 13 numerical features representing the chemical composition of wine. The dataset contains data of chemical composition of Italian wine derived from three different crops: It is partitioned in three classes whose belong 59, 71, and 48 objects, respectively. We executed the two algorithms setting the number of clusters C to 3.

Tables 5 and 6 show the resultant four indicators obtained by using FCM and EwFCM, respectively.

The four metrics are calculated for each class and the last column shows, for each metric, the average value among those calculated for each class.

Table 5. Dataset Wine: FCM results.

Index	Class 1	Class 2	Class 3	Mean
Accuracy	91.33%	87.79%	89.35%	89.49%
Precision	83.33%	87.88%	82.61%	84.61%
Recall	93.22%	81.69%	79.17%	84.69%
F1 Score	88.00%	84.67%	80.85%	84.51%

Table 6. Dataset Wine: EwFCM results.

Index	Class 1	Class 2	Class 3	Mean
Accuracy	95.43%	92.53%	92.53%	93.50%
Precision	90.48%	93.94%	85.71%	90.04%
Recall	96.61%	87.32%	87.50%	90.48%
F1 Score	93.44%	90.51%	86.60%	90.18%

These results confirm the ones obtained by applying FCM and EwFCM on the Iris dataset: The values of all the performance indicators obtained by running the EwFCM algorithm were higher than those obtained by running the FCM algorithm. This trend was obtained for all training sets used in our experiments.

Table 7 shows the mean accuracy, precision, and recall obtained for the dataset wine by using FCM, EwFCM, Density-based FCM, Kernel-based FCM, and PSO FCM.

Table 7. Dataset Wine: Comparison with other FCM-based algorithms.

Index	FCM	EwFCM	Density	Kernel	PSO
Accuracy	89.49%	93.50%	93.10%	92.79%	93.81%
Precision	84.61%	90.04%	89.47%	89.16%	90.61%
Recall	84.69%	90.48%	89.78%	89.51%	90.84%
F1 Score	84.51%	90.18%	89.54%	89.24%	90.66%

For the results of the tests performed on the Iris dataset and Wine dataset, the optimal values of mean accuracy, precision, recall, and F1-score were obtained using the algorithms EwFCM and PSO FCM.

Table 8 shows the computational cost results obtained by running all algorithms.

Table 8. Dataset Wine: Number of iterations and running time comparison.

FCM	Iterations	Running Time (s)
FCM Min	17	0.145
FCM Mean	18.2	0.148
EwFCM	10	0.137
Density FCM	12	0.142
Kernel FCM	12	0.143
PSO FCM	9	0.145

Similarly to the previous test as well as in this case, the least number of iterations was obtained running EwFCM and PSO FCM. It was almost half than that used by FCM and the running time in EwFCM was slightly lower than in FCM, Density FCM, Kernel FCM, and PSO FCM.

Table 9 shows, for each metrics, the mean values of the difference between the measure obtained applying, respectively, EwFCM, Density FCM, Kernel FCM, and PSO FCM and the one obtained by applying FCM.

Table 9. Mean difference of the measured values of the four performance indicators obtained with respect to FCM.

Index	EwFCM	Density	Kernel	PSO
Accuracy	2.91%	2.50%	2.45%	2.90%
Precision	3.78%	3.32%	3.28%	3.77%
Recall	3.95%	3.42%	3.36%	3.97%
F1 Score	3.86%	3.37%	3.32%	3.85%

These results show that the EwFCM, Density FCM, Kernel FCM, and PSO FCM algorithms, applied to classification datasets of different size and number of features, produced a better performance than the FCM algorithm in obtaining accuracy, precision, recall, and F1-score measurements with an average of more than 2% than that obtained by running the FCM algorithm. In particular, EwFCM and PSO FCM had the best performance with the mean difference of the accuracy being near 3% and the mean difference of the precision, recall, and F1 score metrics being near 4%.

Table 10 summarizes the computational cost results obtained for all datasets by running, respectively, EwFCM, Density FCM, Kernel FCM, and PSO FCM. The minimum, mean, and maximum values of the ratio are shown between the number of iterations and what was obtained by running FCM as well as the ratio between the running time and that obtained by running FCM. Here we have considered, for each experiment, the minimum number of iterations and the shortest running time obtained by executing the FCM algorithm ten times.

Table 10. Synthesis of number of iterations and running time comparison.

Method	Measure	Iterations	Running Time (s)
EwFCM	Min Ratio	0.41	0.81
	Mean Ratio	0.51	0.93

	Max Ratio	0.58	0.97
	Min Ratio	0.48	0.90
Density FCM	Mean Ratio	0.55	1.02
	Max Ratio	0.64	1.07
	Min Ratio	0.49	0.88
Kernel FCM	Mean Ratio	0.56	1.02
	Max Ratio	0.68	1.08
	Min Ratio	0.40	0.91
PSO FCM	Mean Ratio	0.51	1.04
	Max Ratio	0.59	1.11

The results in Table 10 show that the number of iterations in EwFCM and PSO FCM was about half that of FCM. Furthermore, EwFCM had a slightly shorter running time than Density FCM, Kernel FCM, and PSO FCM.

5. Conclusions

We presented a variation of the FCM in which a using a fuzzy entropy function was used in the initialization phase to optimize the assignment of cluster centers. This allowed us to overcome the FCM algorithm problem of detecting local solutions by randomly assigning the initial positions of the clusters. To set the initial cluster centers we used the FCM weighted algorithm in which the weight assigned to each object was greater the smaller its fuzziness. The results of the comparative tests performed on over 40 datasets in the UCI machine learning repository showed that the performance of the proposed algorithm was better than that obtained by applying the FCM algorithm. The mean difference of the accuracy was about 3% and the mean difference of the precision, recall, and F1 score metrics was near 4%. The results of comparisons with other FCM-based algorithms aimed to optimize the initialization phase showed that EwFCM and PSO FCM provided the best performances in terms of accuracy, prediction, recall, and F1-score. Furthermore, the number of iterations in EwFCM and PSO FCM was about half that in FCM. Finally, EwFCM had a slightly shorter running time than Density FCM, Kernel FCM, and PSO FCM.

Further tests are needed in the future. We intend to improve the proposed algorithm by exploring its use on massive datasets and testing the use of robust weight functions with respect to the presence of noise and outliers in the data. We also intend to carry out performance comparison tests of EwFCM with respect to other well-known machine learning clustering algorithms.

Author Contributions: Conceptualization, B.C. and F.D.M.; methodology, B.C. and F.D.M.; software, B.C. and F.D.M.; validation, B.C. and F.D.M.; formal analysis, B.C. and F.D.M.; investigation, B.C. and F.D.M.; resources, B.C. and F.D.M.; data curation, B.C. and F.D.M.; writing—original draft preparation, B.C. and F.D.M.; writing—review and editing, B.C. and F.D.M.; visualization, B.C. and F.D.M.; supervision, B.C. and F.D.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest

References

1. Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms*; Kluwer Academic Publishers: Norwell, MA, USA, 1981; p. 256.
2. Bezdek, J.C.; Ehrlich, R.; Full, W. FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **1984**, *10*, 191–203.
3. MacQueen, J.B. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*; University of California Press: Berkeley, CA, USA, 1967; Volume 1, pp. 281–297.

4. Liu, W.-Y.; Xiao, C.H.-J.; Wang, B.-W.; Shi, Y.; Fang, S.-F. Study on combining subtractive clustering with fuzzy c-means clustering. In Proceedings of the 2003 Winter Simulation Conference, Xi'an, China, 5–5 November 2003; pp. 2659–2662.
5. Chen, J.; Qin, Z.; Jia, J. A Weighted Mean Subtractive Clustering Algorithm. *Inf. Technol. J.* **2008**, *7*, 356–360.
6. Yang, Q.; Zhang, D.; Tian, F. A initialization method for Fuzzy C-means algorithm using Subtractive Clustering. In Proceedings of the Third International Conference on Intelligent Networks and Intelligent Systems, Shenyang, China, 1–3 November 2010 Volume10, pp. 393–396.
7. Zou, K.-Q.; Wang, Z.-P.; Pei, S.-J.; Hu, M. An New Initialization Method for Fuzzy c-Means Algorithm Based on Density. In *Advances in Multimedia, Software Engineering and Computing Vol.2*; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2009; Volume 54, pp. 547–553.
8. Ding, Y.; Fu, X. Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm. *Neurocomputing* **2016**, *188*, 233–238.
9. Li, D.; Han, Z.; Zhao, J. A Novel Level Set Method with Improved Fuzzy C-Means Based on Genetic Algorithm for Image Segmentation. In Proceedings of the 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM); Chengdu, China, 10–11 August 2017; pp. 151–157.
10. Siringoringo, R.; Jamaluddin, J. Initializing the Fuzzy C-Means Cluster Center with Particle Swarm Optimization for Sentiment Clustering. In *Proceedings 1st International Conference of SNIKOM 2018*; IOP Publishing: Bristol, UK, 2019; Volume 1361, p. 012002.
11. Franco, D.G.D.B.; Steiner, M.T.A. Clustering of solar energy facilities using a hybrid fuzzy c-means algorithm initialized by metaheuristics. *J. Clean. Prod.* **2018**, *191*, 445–457.
12. Jung, F.H.; Hung, W.X.; Ping, M.H.; Bin, W. Fuzzy Entropy Clustering Using Possibilistic Approach. *Procedia Eng. – Advanced Control Engineering Inf. Sci.* **2011**, *15*, 1993–1997.
13. Krishnapuram, R.; Keller, J. The possibilistic C-means algorithm: insights and recommendations. *IEEE Trans. Fuzzy Syst.* **1996**, *4*, 385–393.
14. Su, X.; Wang, X.; Wang, Z.; Xiao, Y. A New Fuzzy Clustering Algorithm Based on Entropy Weighting. *J. Comput. Inf. Syst.* **2010**, *6*, 3319–3326.
15. Lee, H.-M.; Chen, C.-M.; Chen, J.-M.; Jou, Y.-L. An efficient fuzzy classifier with feature selection based on fuzzy entropy. *IEEE Trans. Syst. Man, Cybern. Part B (Cybernetics)* **2001**, *31*, 426–432.
16. Wu, Z.; Chen, G.; Yao, J. The Stock Classification Based on Entropy Weight Method and Improved Fuzzy C-means Algorithm. In Proceedings of the 2019 4th International Conference on Big Data and Computing - ICBDC 2019; New York, NY, USA, May 2019; pp. 130–134.
17. Kahali, S.; Sing, J.K.; Saha, P.K. A new entropy-based approach for fuzzy c-means clustering and its application to brain MR image segmentation. *Soft Comput.* **2018**, *23*, 10407–10414.
18. Zhou, J.; Chen, C.L.P.; Chen, L. Maximum-entropy-based multiple kernel fuzzy c-means clustering algorithm. In Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC); San Diego, CA, USA, 5–8 October 2014; pp. 1198–1203.
19. Zheng, W.; Mo, S.; Duan, P.; Jin, X. An improved pagerank algorithm based on fuzzy C-means clustering and information entropy. In Proceedings of the 2017 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE), Beijing, China, August 2017; pp. 615–618.
20. Yang, C.-H.; Chuang, L.-Y.; Lin, Y.-D. Epistasis Analysis using an Improved Fuzzy C-means-based Entropy Approach. *IEEE Trans. Fuzzy Syst.* **2019**, *1*.
21. De Luca, A.; Termini, S. A definition of non-probabilistic entropy in the setting of fuzzy sets theory. *Information Control* **1972**, *20*, 301–312.
22. A. De Luca, S. Termini, Entropy and Energy Measures of Fuzzy sets. In *Advances in Fuzzy Set Theory and Applications*; Gupta, M.M., Ragade, R.K., Yager, R.R., Eds.; North-Holland: Amsterdam, The Netherlands, 1982.
23. Hathaway, R.; Hu, Y. Density-Weighted Fuzzy c -Means Clustering. *IEEE Trans. Fuzzy Syst.* **2008**, *17*, 243–252.
24. Ji, Z.; Xia, Y.; Chen, Q.; Sun, Q.-S.; Xia, D.; Feng, D.D. Fuzzy c-means clustering with weighted image patch for image segmentation. *Appl. Soft Comput.* **2012**, *12*, 1659–1667.
25. Gong, M.; Liang, Y.; Shi, J.; Ma, W.; Ma, J. Fuzzy C-Means Clustering with Local Information and Kernel Metric for Image Segmentation. *IEEE Trans. Image Process.* **2012**, *22*, 573–584.
26. Havens, T.C.; Bezdek, J.C.; Leckie, C.; Hall, L.O.; Palaniswami, M. Fuzzy c-Means Algorithms for Very Large Data. *IEEE Trans. Fuzzy Syst.* **2012**, *20*, 1130–1146.

27. Ming, T.K. *Encyclopedia of Machine Learning*; Sammut, C., Webb, G.I.; Springer: Berlin/Heidelberg, Germany, 2011; ISBN 978-0-387-30164-8.
28. Tharwat, A. Classification assessment methods. *Appl. Comput. Informatics* **2018**, *13*.



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).