Iranian Journal of Fuzzy Systems

Volume 17, Number 4, (2020), pp. 85-104

University of Sistan and Baluchestan

Refining membership degrees obtained from fuzzy C-means by re-fuzzification

M. Javadian[1], R. Vaziri[2], S. Haghzad Klidbary[3] and A. Malekzadeh[4]

[1] *Department of Computer Engineering, Faculty of Information Technology, Kermanshah University of Technology, Kermanshah, Iran*

[2] *Islamic Azad University Central Tehran Branch, Tehran, Iran*

[3] *Department of Computer Engineering, Faculty of Engineering, University of Zanjan, Zanjan, Iran*

[4] *Department of Computer Science and Statistics, Faculty of Mathematics, K.N. Toosi University of Technology, Tehran, Iran*

mo.javadian@gmail.com, rez.vaziri@iauctb.ac.ir, s.haghzad@znu.ac.ir, admalekzadeh@yahoo.com

Abstract

Fuzzy C-mean (FCM) is the most well-known and widely-used fuzzy clustering algorithm. However, one of the weaknesses of the FCM is the way it assigns membership degrees to data which is based on the distance to the cluster centers. Unfortunately, the membership degrees are determined without considering the shape and density of the clusters. In this paper, we propose an algorithm which takes the FCM clustering results and re-fuzzifies them by taking into account the shape and density of the clusters. The algorithm first defuzzifies the FCM clustering results. Then the crisp result is fuzzified again. Re-fuzzification in our algorithm has some advantages. The main advantage is that the fuzzy membership degrees of data points are obtained based on the shape and density of clusters. Adding the ability to eliminate noise and outlier data is the other advantage of our algorithm. Finally, our proposed re-fuzzification algorithm can slightly improve the FCM clustering quality, because the data points change their clusters according to similarity to the shape and density of their respective clusters. These advantages are supported by simulations on real and synthetic datasets.

Keywords: Fuzzy C-means, FCM, re-fuzzification, F3CM, fuzzified FCM, fuzzy clustering, KFCM.

1 Introduction

Recently, with the ever-increasing utilization of digital sensors, cameras, the internet, social networks on one hand, and the higher capacity of data storage, on the other hand, we are confronted with large volumes of data. Furthermore, industrial enterprises, factories, trade companies, information centers, and even medical offices are interested in analyzing such large data and converting them into useful information. Naturally, the first step while confronting large data is categorization. Hence, many researchers have attempted to present methods to categorize large data [5, 53].

Categorization methods are generally divided into three groups. Categorization with supervision (or classification), without supervision (or clustering), or semi-supervised. In classification, the number of categories and a considerable amount of labeled data exist. In clustering, the number of categories and labels are not clear. In Semi-supervised categorization, few amounts of data labels exist, and there are a large amount of unlabeled data. Many algorithms for each of the categorization methods are proposed [4, 5]. Also, each of the categorization methods has a particular application [20].

There are many cases, where data labels and the number of classes are not known in advance. Clustering algorithms (or unsupervised learning) have been developed for such cases. In clustering, it is attempted to place similar data in one cluster. On the other hand, data in each cluster must have the least similarities with data in the other clusters. The criterion for similarity measure is different in various clustering algorithms. Some of the clustering algorithms

Corresponding Author: M. Javadian

Received: April 2019; Revised: January 2020; Accepted: February 2020.

try to find the densely populated areas of data points, such as UALM [28], DENCLUE [23, 24], and DBSCAN [18]. Some methods divide the space into sections by considering the distances of the data points to center points, such as kmeans [40] and kmedoid [32]. Yet other methods try to match data with a distribution model, such as EM [15]. Finally, in some methods, the clustering of data is done hierarchically based on the distances between data points [5]. However, some researchers make use of evolutionary algorithms for clustering purposes [16, 26, 65].

The clustering result for a data point can be of two different kinds; either data is placed in a cluster and assigned a label, or the data is not similar enough to any cluster and is treated as an outlier. Clustering methods that behave this way are called crisp clustering.

After the fuzzy logic was proposed by *Zadeh* [66] a third kind of clustering result was introduced in the year 1973 [10, 17] called fuzzy clustering which quickly found many applications. In this kind of clustering, a data point does not belong to a single cluster, and it potentially can belong to any cluster but with different degrees of belongings which are values between 0 and 1. Some of the benefits of the fuzzy clustering are 1) more realistic labeling of the data, 2) more accurate decision making based on clusters and data. For instance, in medical sciences using crisp methods on lab results leads to the conclusion that the subject is either sick or not sick. However, in many medical applications, we need to know with what degree of certainty the subject is sick or not [31]. For such cases, fuzzy methods can be effective. Hence, fuzzy methods can express uncertainty in clusters efficiently.

Among fuzzy clustering algorithms, Fuzzy C-means (FCM) is the most well-known and widely-used. This algorithm which is based on the optimization of an objective function was proposed by *Dunn* in 1973 [17] and developed by *Bezdek* [10]. In 1979 *Gustafson* and *Kessel* [21] proposed fuzzy clustering using fuzzy covariance matrix. Later on, the method was extensively developed, and various versions of it were introduced [45]. One of the benefits of FCM which has made it particularly attractive is its ease of use and less computation time. In practice, with few repetitions a final solution is achievable.

However, the FCM algorithm also has many problems regarding the clustering of data. One of the problems is its lack of ability to recognize and identify clusters with arbitrary shapes and densities [27]. Another issue is that the sum of membership degrees should be equal to one. This pre-condition may cause problems when clustering some special datasets such as the Raspini butterfly as well as confronting outliers [47]. Another issue of this method is its sensitivity to the initial values of cluster centers and the probability of getting trapped in local minima [42]. Also, this method is sensitive to outliers and noise data in a way that such data can affect the final clustering and displace the centers of the clusters [64]. Finally, since this method determines the membership degrees based on the distance to the centers of the clusters, it may not be appropriate when dealing with clusters with varying shapes and densities. These problems have led to the publication of various improved versions of the algorithm in the literature, which we will discuss in the Related Work Section.

Due to its simplicity and ease of use, FCM has many applications in many fields [9, 46], such as image processing [13, 30, 49], electrical power engineering [11, 41, 58], medical engineering [1, 60, 61], chemistry [22, 39, 59], and economy and financial management [8,12,14,51]. The problems that current algorithms have with the fuzzy expression of clusters drove us to present a new method which resolves some of the problems. Generally, our proposed algorithm has three advantages:

1. More natural degrees of Membership: Obtaining more natural degrees of memberships based on shape and density and not merely as a function of the distance from data point to the cluster centers. This advantage is our motivation to propose re-fuzzification of the FCM algorithm.

2. **Raising the Quality of Clustering**: Considering that membership degrees are obtained based on shape and density, during defuzzification the final result may differ from that of the original clustering. As has been demonstrated in the simulations, this difference in most cases leads to slightly higher quality clustering.

3. Avoiding Sensitivity to Outliers and noise Data: Noise data points and outliers could change the center position of the clusters in the FCM and PCM algorithms, which leads to unreliable results of clustering. However, by adding the proposed method to the algorithms as a post-process the noise data points and outliers will be rejected efficiently.

The remainder of this paper is organized as follows. In Section 2, related works including the FCM and its different variations are discussed. The proposed algorithm is described in Section 3. Section 4 shows the experimental results of the quality of FCM, KFCM, and GPFCM in comparison with the fuzzified version of FCM (F3CM) and KFCM (FKFCM). In order to show the advantage of our approach in noisy environments, synthetic datasets with variable amounts of noise have been employed in simulations. Parameter sensitivity analysis and a parameter selection method are also introduced in Section 4. Finally, Section 5 concludes this paper.

2 Related works

Up until now the majority of the fuzzy clustering algorithms have been based on the FCM algorithm which has been proposed by Dunn and Bezdek [10,17]. FCM is a clustering algorithm which allows an object to be a member of more than one cluster, but with different membership degrees.

The standard FCM algorithm is a simple method which attempts to minimize an objective function iteratively. Here we briefly state the formulas of this algorithm. If we assume $X = \{x_k, k = 1, 2, \ldots, n\}$ is a set of vectors with p features in feature space such that $x_k = \left(x_k^1, \ldots,\ x_k^p\right) \in R^P$, and c is the ideal number of clusters. Also $v = \{v_i, i = 1, 2, \ldots, c\}$, where v_i is the ith vector of cluster centers and $v_i \in R^P$. Partition matrix $U = u_{ik} \in [0, 1]\,,\ i = 1, 2, \ldots, c,\ k = 1, 2, \ldots, n$, where each element u_{ik} is the membership degree of x_k in the ith section of subset X where satisfy the following relationships:

$$\sum_{i=1}^{c} u_{ik} = 1 \quad for\ all\ k \tag{1}$$

The FCM algorithm has an objective function as the following:

$$J_m = \sum_{i=1}^{c}\sum_{k=1}^{n} u_{ik}^m\ ||x_k - v_i||^2 \tag{2}$$

Where $m > 1$ is the weighting exponent for each fuzzy membership function. Considering (1) and (2), and repeating the optimization steps, FCM can find the appropriate values of u_{ik} and v_i.

The algorithm is composed of the steps in figure 1.

1- *Initialize* $U = [u_{ik}]$ *matrix,* $U^{(0)}$

2- *Atjth step: calculate the center's vectors* $v^{(j)} = [v_i] with U^{(j)}$

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m . x_k}{\sum_{k=1}^{n} u_{ik}^m}$$

3- *Update* $U^{(j)}$, $U^{(j+1)}$

$$u_{ik} = \frac{1}{\sum_{l=1}^{c}\left(\frac{\|x_k - v_i\|}{\|x_k - v_l\|}\right)^{\frac{2}{m-1}}}$$

4- *If* $\left\|U^{(j+1)} - U^{(j)}\right\| < \varepsilon$, *then STOP; otherwise return to step 2.*

Fig. 1. Fuzzy c-means algorithm.

After the membership degrees of u_{ik} for all values of i and k are obtained, we could defuzzify the results by labeling the input vector of x_k with j_k which is obtained by (3).

$$j_k = \arg_i \max u_{ik} \quad for\ all\ k. \tag{3}$$

Using a *sum-of-one* for membership degrees for datasets such as Raspini's butterfly, causes similar degrees to be assigned to far and near data points of a cluster. To resolve this problem possibilistic c-mean (PCM) [47] was introduced. In this method, the pre-condition of the *sum-of-one* has been eliminated. Therefore, each data point is assigned membership degrees to all clusters based on the Euclidean distances to the centers. In this method, farther data are assigned a lesser membership degree. Hence, there is a better chance of identifying outliers. The lack of ability of FCM and PCM in identifying clusters with varying shapes and densities has been considerably resolved by Kernel-based methods [19]. In these methods, by applying a mapping, the data are mapped to a higher space in a way that in the new space they can be separated linearly. However, these methods also have their own problems, such as every kernel is not appropriate for every dataset. Hence, multi-kernel methods have been developed as well [25]; although this remains a challenging issue in this subject [2].

Another problem of this method is its sensitivity to the initial values of the cluster centers and getting trapped in the local minima [6]. To avoid this problem, solutions have been presented [29]. For instance, Soft DBSCAN algorithm [56] initially uses the DBSCAN algorithm to identify the number and location of cluster centers based on the density of data. Next, it applies the FCM method to the dataset using Mahalanobis distance criterion. Density-based fuzzy c-means

clustering algorithm (D-FCM) [48] is another algorithm which was developed to cope with the problem of initializing the number of clusters. D-FCM considers a density function for each data point and then uses the density peaks in order to determine the number of clusters and the initial membership matrix automatically.

Another problem of this method is its sensitivity to outliers and noise data in a way that such data can affect the final clustering and displace the cluster centers. To resolve this problem also some solutions have been proposed. One is the use of median instead of average which leads to less sensitivity to outliers [38, 43]. However, the problem will remain in spaces with lots of noise. The generalized form of Possibilistic Fuzzy C-Means algorithm (GPFCM) is also presented in order to tackle the clustering of noisy data [7]. In GPFCM a function of distance is used instead of the distance itself to dampen noise contributions. GPFCM finds accurate cluster centers where the data are highly noisy and FCM, PCM, and PFCM algorithms usually fail.

Lastly, we can refer to the membership degree calculation method which is a function of the distance of the data point to the cluster centers; however, this is not a proper method for datasets with varying cluster densities and shapes. To resolve this, Gustafson and Kassel, using a Co-variance matrix, also define oval-shaped clusters [21]; however, if the clusters are not circular or oval, the problem will remain.

Some algorithms such as extended versions of the fuzzy c-means (E-FCM) and the Gustafson–Kessel (E-GK) algorithms [33] deal with the clustering of non-convex clusters by merging similar clusters which are formed by the basic versions of the FCM and GK. These two algorithms have more computational cost (three to four times more) than the basic versions.

In this paper, we propose an algorithm which reduces some of the problems of FCM and other related algorithms. Our proposed algorithm adds a post-process to the FCM algorithm, which can improve the quality of the FCM clustering algorithm by resolving some of the above-mentioned problems. As stated earlier, one of the biggest problems of the FCM algorithm is the membership degree calculation; which is a function of the distance to the center of the cluster, while cluster shape and density do not play a role in this function. To state membership degrees more accurately we must consider cluster shapes and densities. The main purpose of the proposed algorithm (our motivation) in this paper is representing the membership degrees by considering the shape and density of the clusters. It should be mentioned that we do not propose a new clustering algorithm. However, we add a simple and beneficial procedure to the FCM algorithm which acts as a post-processing method on the FCM clustering results. The proposed algorithm, at first defuzzifies the FCM results, then does the re-fuzzification process on the crisp results. As a result of this re-fuzzification, three desirable outcomes are obtained. The first and main advantage is the better membership degree assignment which is based on the shape and density of clusters. The second advantage is improving the ability of the FCM algorithm for eliminating and handling noise and outliers. The third advantage is improving the accuracy and quality of the FCM clustering result because the re-fuzzification procedure makes some data points change their clusters based on the similarity of the shape and density of the clusters. Finally it should be noted that if we apply the *sum-of-one* pre-condition for calculating the membership degrees the algorithm will be similar to FCM; on the other hand, if we ignore this pre-condition, the algorithm will be similar to the PCM algorithm.

Table 1: Comparison of different versions of FCM algorithm

Weaknesses	Proposed F3CM	Soft DBSCAN, D-FCM	E-FCM, E-GK	Kernel-Based	GK	PCM	FCM
Pre-condition of the *sum-of-one*						▲	▼
Lack of ability to recognize arbitrarily shaped clusters.	◄	◄	▲	▲	►	▼	▼
Sensitivity to the initial values of cluster centers	▼	▼	▼	▼	▼	▼	▼
Need to know the number of clusters	▼	▲	◄	▼	▼	▼	▼
Lack of ability to recognize outliers and noise data	▲	▲	◄	►	◄	◄	▼
Membership degree calculation as a function of distance to cluster centers	▲	▼	▼	▼	▼	▼	▼

Guide

May have a problem or not	▲Problem is resolved	►Problem is almost removed	◄Almost has a problem	▼Has a problem

As figure 2 shows the difference of applying the fuzzification to FCM and PCM are only the amplitude of the membership degrees, however, the membership functions of these two algorithms have the same shapes. Therefore, the application of the proposed algorithm to FCM and PCM has the same effect. Table 1 demonstrates some of the prominent weaknesses of the FCM algorithm. The table also compares our proposed method with other methods taking into account the previously mentioned weaknesses.

3 The proposed algorithm

The proposed algorithms have three distinct steps. The first step is the execution of the FCM method. The second step is converting the result into a crisp form. Finally, the third step is the re-fuzzification of the result. The first and second steps of the algorithm are similar to the steps mentioned in the previous section, and use the algorithm of figure 1 for FCM clustering and (3) for defuzzification. Hence, in the following, we only explain the third step. Figure 2 shows the flowchart of our proposed algorithm illustrated by an example.

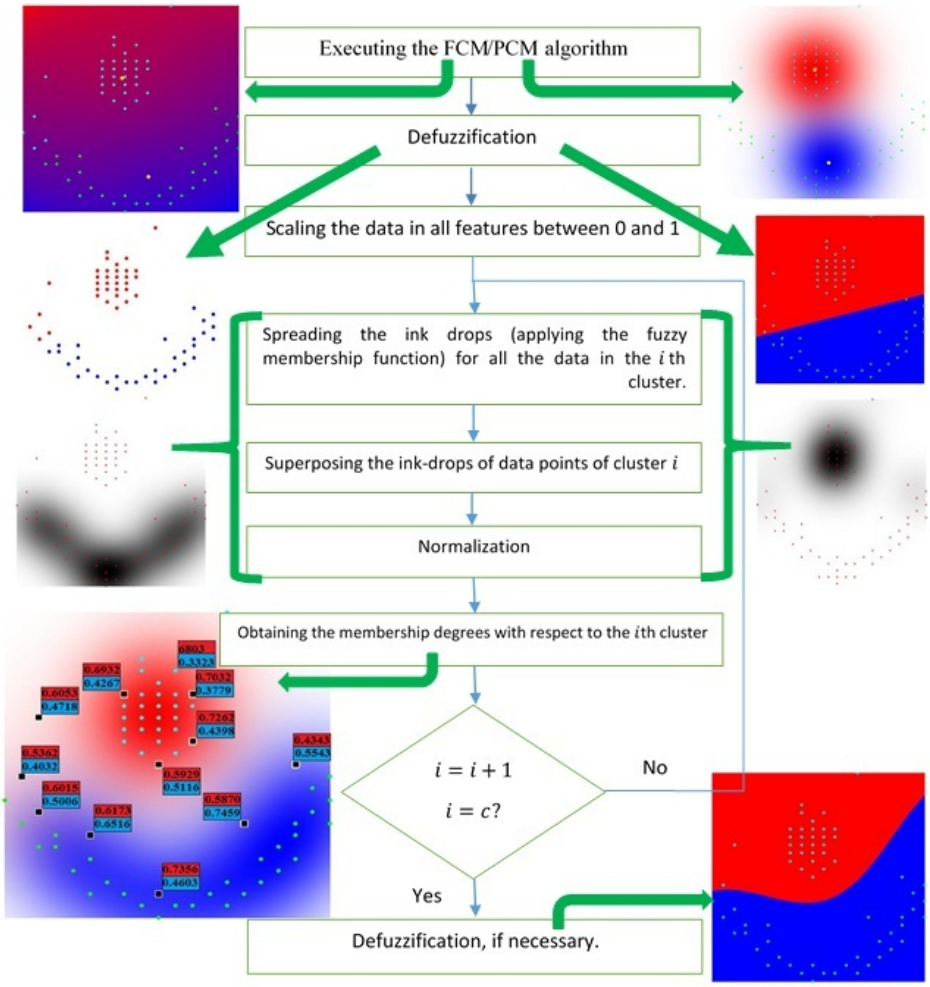


Fig. 2. The Flowchart of the proposed algorithm

3.1 Scaling the data

Initially, to spread the ink drops symmetrically, all data in all dimensions are scaled between 0 and 1. Equation (4) shows the scaling function for the pth feature.

$$S\left(x_k^p\right) = \frac{x_k^p - Min^p}{Max^p - Min^p} \tag{4}$$

Where x_k^p is the pth feature of the kth data point, Min^p is the minimum of the pth feature among all data points. In other words, $Min^p = \min_{k=1,\ldots,n} x_k^p$. Similarly, Max^p definition is $Max^p = \max_{k=1,\ldots,n} x_k^p$.

3.2 Spreading the ink drop

In this step, for all data of the ith cluster, an ink drop is spread. The idea of spreading the ink drop is inspired by the ALM algorithm [27,28,34–37,44,52,54,55] which is a powerful method in fuzzy modeling. In this method by assigning an ink drop to each data, the uncertainty for data is modeled by a membership function [3]. Common sense implies that each sample data not only has information at its exact point, but also is valid in its neighboring points with a lesser degree of confidence. As you go away from the point of sample data, the confidence degree decreases. Figure 3(a) depicts two such membership functions overlapped and aggregated. An IDS plane after applying the IDS operator to the five data samples is shown in Figure 3(b).

Generally, an ink drop is a mapping from R^p space to R^{p+1}, which maps every point M in space R^p to a continuous region in space R^{p+1}. The region at hand which is called the Ink of point M is described by the following Equation (5). The coordinates of M are $M(a_1, a_2, \ldots, a_p)$.

$$\begin{aligned} Ink(M) = \{(x_1, x_2, \ldots, x_p, f(u_M(x_1, x_2, \ldots, x_p))) \,|\, u_M(x_1, x_2, \ldots, x_p) \\ < R, f(u_M + \varepsilon) \leq f(u_M), f(0) = \max(f(u_M)) \ = 1\} \end{aligned} \tag{5}$$

in which $\varepsilon > 0$ and $u_M(x_1, x_2, \ldots, x_p) = \sqrt{\sum_{j=1}^{p} (x_j - a_j)^2}$. Also, f is a descending function which means as we get farther from the point's location, the value of function f diminishes. Also, R is the radius of the ink drop spread. Since we want the ink drop of each data point to spread in the entire space, we consider the R value to be equal to 1. (Reminder: In the scaling step all features' values of data are forced to be between 0 and 1). The parameters of function f are the parameters of our algorithm. For instance, if the function f is Gaussian the value of the standard deviation for the Gaussian function is the parameter for our algorithm. The value of this parameter depends heavily on the density of data. Usually selecting the value of this parameter less than $R/10$ results in a good outcome. Hence, there is a high degree of membership for data neighboring M, and a low degree of membership for farther data.

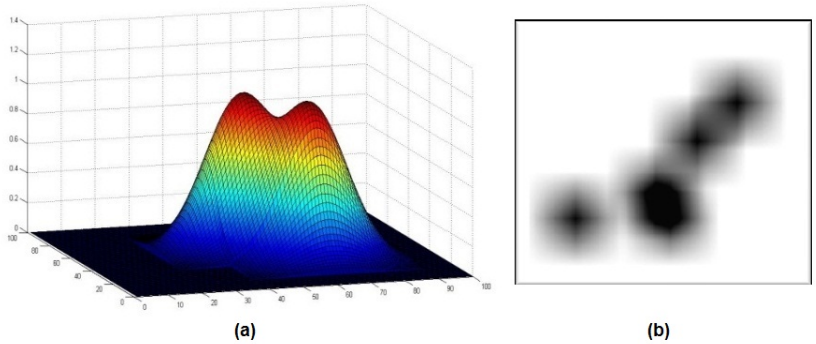


Fig. 3. (a) An ink-drop pattern with pyramid shape; (b) Five drops are diffused on the plane.

3.3 Superposing the effects

In this step, all the ink drops of the data in the ith cluster are superposed together. This is called Ink Drop Spread of the ith cluster and is represented by IDS_i (6).

$$IDS_i = \sum_{k=1}^{n_i} Ink\left(M_k^i\right) = \left\{ \left(x_1, x_2, \ldots, x_p, d^i(x_1, x_2, \ldots, x_p)\right) \,|\, d^i(x_1, x_2, \ldots, x_p) = \sum_{k=1}^{n_i} f\left(u_{M_k^i}\right) \right\} \tag{6}$$

in which M_k^i is the kth point in the ith cluster, and n_i is the total points in the ith cluster. Also $d^i(x_1, x_2, \ldots, x_p)$ is a function from R^p to R which shows the darkness of coordinate $(x_1, x_2, \ldots, x_p)$ resulted from the superposition of the inks of the ith cluster.

3.4 Normalization

To place the darkness values between zero and one, after superposing the ink drops, the results are normalized by (7).

$$\overline{IDS_i}=\left\{\left(x_1,x_2,\ldots,x_p,\overline{d^i}\left(x_1,x_2,\ldots,x_p\right)\right)\left|\overline{d^i}\left(x_1,x_2,\ldots,x_p\right)=\frac{d^i\left(x_1,x_2,\ldots,x_p\right)}{\max_{R^p} d^i\left(x_1,x_2,\ldots,x_p\right)}\right.\right\} \tag{7}$$

in which $\overline{IDS_i}$ is the normalized value of IDS_i. In fact, $\overline{IDS_i}$ is the membership function of the ith cluster.

3.5 Assigning membership degrees

In this step, the darkness values of the ith cluster for all data points are calculated. The normalized darkness value of each cluster in the location of each data point identifies the membership degree of the data point to that cluster. Obviously, the darkness value of a cluster in the location of data points which belong to that cluster would be higher than the darkness value of data points which belong to other clusters. The output of this step is the partition matrix of P:

$$P=\left[\begin{pmatrix} m_{11} & \cdots & m_{1c} \\ \vdots & \ddots & \vdots \\ m_{n1} & \cdots & m_{nc} \end{pmatrix}\right]_{n\times c} \tag{8}$$

in which n is the total number of points and c is the number of clusters. m_{ik} is obtained by (9).

$$m_{ik}=\overline{d^i}\left(x_k^1,x_k^2,\ldots,x_k^p\right) \tag{9}$$

3.6 Defuzzification

In this step, defuzzification is done on the clusters. By this operation, the cluster which has the highest membership degree for a point assigns its label to that point. Defuzzification is expressed by (10) using the values from the partition matrix (8):

$$L_k=\underset{k=1,\ldots,c}{\operatorname{argmax}} m_{ik}\ ,\quad i=1,\ldots,n \tag{10}$$

in which m_{ik} is the membership degree of the kth data point to the ith cluster. L_k is the label for the kth data point, and n is the total number of points.

3.7 Complexity analysis

In this section, we would like to analyze the complexity of the proposed FCM re-fuzzification algorithm. Since the proposed algorithm considers a fuzzy membership function for each cluster, to compute the membership function for all the clusters, np summations are needed, where n is the number of data points of the data set and p is the number of features. Hence, the computational complexity for obtaining the membership functions of clusters will be in the order of $O(np)$. Also, if we want to obtain the membership degrees for all data with respect to all clusters, np summations will be performed for each data point. Hence, the computational complexity for obtaining the membership degrees of all data points with respect to all clusters will be of the order $O(n^2p)$.

It should be noted that the complexity of the proposed algorithm will be added to the complexity of the fuzzy clustering algorithms; because the algorithm refines the membership degrees obtained by the original fuzzy clustering algorithm and acts as a post process algorithm.

4 Simulation

One of the problems with the FCM algorithm is how membership degrees of data points to clusters are calculated. In FCM, membership degrees are calculated solely as a function of the distance to the cluster centers without considering shape or density of the clusters. In this paper we have not presented a new clustering algorithm, however, by applying a post-process to the FCM results we have re-fuzzified it again. Although the proposed algorithm adds time complexity in the order of $O(n^2p)$ (n is the total number of data points and p is the feature size) to the original fuzzy clustering algorithms, but this re-fuzzification removes the problem mentioned above and has the following benefits:

1. Obtaining more natural degrees of memberships based on shape and density.
2. Raising the quality of clustering.

3. Avoiding sensitivity to outliers and noise data.

To prove the above three claims, simulations have been performed for each claim which will be presented in this section. Also, the sensitivity of the algorithm with respect to its parameters have been evaluated. Real world and synthetic data sets are used for evaluations. For a better understanding of the benefits of our proposed algorithm, we also provide an example with plenty of noise data injected to the original data set (Figure 4).

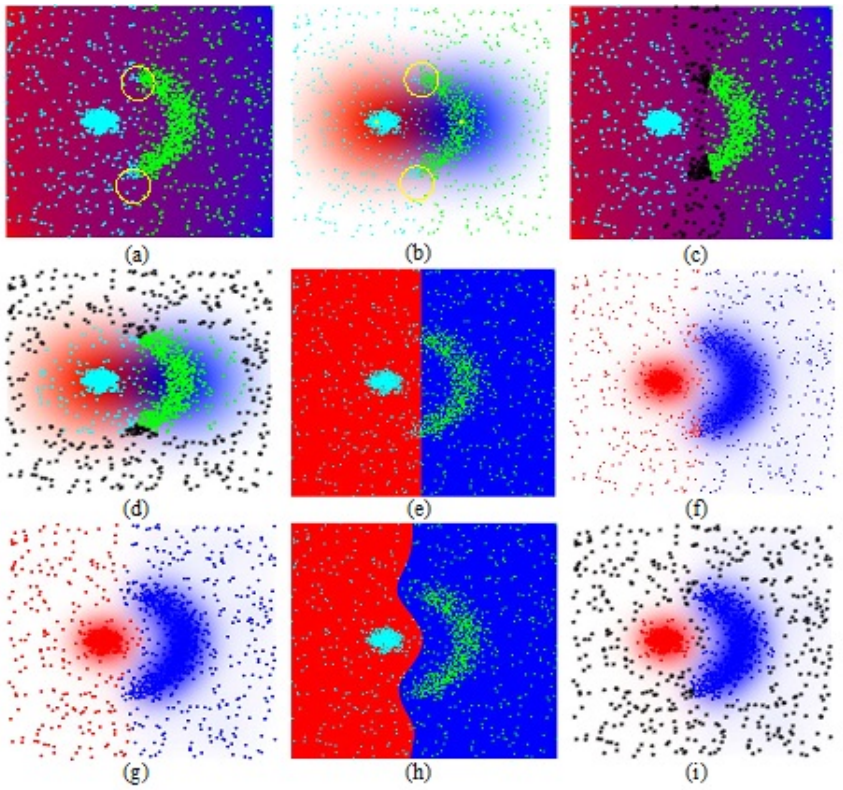


Fig. 4. (a) The FCM clustering result. The FCM algorithm could not correctly cluster the data set. (b) The PCM clustering result. The PCM algorithm could not correctly cluster the data set. (c) The FCM clustering result after applying a threshold on the membership degrees to reject outliers. FCM wrongly considers the data points around the perpendicular bisector of a line segment between cluster centers as noise. (d) The PCM clustering result after applying a threshold on the membership degrees to reject outliers. PCM wrongly considers some of the rightmost cluster data as noise. It also fails to detect all noise data points. (e) The crisp result of the FCM and PCM clustering. (f) The result of spreading the inks of the clusters. (g) The result of defuzzification based on the value of darkness of each cluster at the position of each data point. (h) The crisp result of our proposed algorithm. (i) The result of the proposed algorithm after applying a threshold on the membership degrees to reject noise data.

Figure 4(a) and Figure 4(b) show the clustering result of FCM and PCM algorithms respectively. As it is shown by the circle the algorithms could not cluster the data sets correctly. Figure 4(c) and Figure 4(d) shows the results of FCM and PCM algorithms when they apply a threshold on the membership degrees to reject noise data points. As it is depicted, the algorithms fail to consistently detect noise data points. The FCM algorithm assigns the lowest membership degrees (about 0.5) to the data points which are located near the perpendicular bisector of a line segment between the cluster centers. The reason for this is the *sum-of-one* condition. Therefore, the FCM algorithm rejects the data points around the boundary between the two clusters as depicted in Figure 4(c). On the other hand, by relaxing the *sum-of-one* condition on membership degrees, PCM introduces noise points based on the distance to the cluster centers.

FCM and PCM algorithms show two problems when confronting noise points. The first problem is due to the displacement of cluster centers because of the existence of noise points in the data set. As a result of this displacement, the centers of the clusters will not be the real centers. In some cases, the centers of clusters may be determined to be out of the real clusters and near noise points. In these cases, some noise points which are near to the cluster centers may take the highest membership degrees and some of the non-noise data points may be considered as noise points.

The second problem is due to the shape of the clusters. FCM and PCM are good for clustering convex shape data sets. However, not all clusters have convex shapes. As Figure 4(d) shows the PCM algorithm wrongly detects some of the points of the right cluster as noise points. This is because of the crescent-shape of the cluster. Figure 4(e) shows the crisp result of both FCM and PCM. By feeding this crisp result to our proposed algorithm and spreading the inks of data points, the results of Figure 4(f) is achieved. Figure 4(f) shows how the idea of spreading ink drops could compensate and modify the wrong results of FCM and PCM algorithms. Figure 4(g) shows the result of defuzzification on the darkness values of the data points and Figure 4(h) shows the crisp result of the proposed algorithm. Figure 4(i) shows the result of the proposed algorithm after applying the threshold on the membership degrees, which demonstrates the capabilities of the algorithm in rejecting noise. As a result, the proposed algorithm is much more reliable for data mining applications.

4.1 Obtaining better membership degrees based on the shape and density of clusters

Membership degrees calculated by FCM have no relation to the shape of the clusters, and they are a function of the distance from the cluster centers. Our main claim in this paper is to refine membership degrees for clusters, found by FCM, in a manner that they have the maximum conformity with the shape and density of the clusters. As an example consider the synthetic data set (Figure 5(a)) where the FCM result is as depicted in Figure 5(b):

As Figures 6.a, b, and c show the membership functions obtained by PCM (or FCM) algorithm (m=11) are radially distributed around the cluster centers. However as Figures. 7.a, b, and c show the proposed algorithm produces membership functions based on the shape and density distribution of the clusters; therefore, the membership degrees are more informative, because they are much more relative to the cluster shapes and densities. Consequently, the results obtained from re-fuzzification of the FCM algorithm is much more reliable for data mining applications than FCM clustering algorithm.

To show the advantage of our proposed algorithm in comparison to FCM in expressing the fuzzy clusters, the fuzzy cluster validation parameters have been examined. Overall these parameters can be divided into two main groups [63, 64]. The group that only uses the membership degrees, and another group that uses membership degrees along with the data themselves. The latter group which is mostly based on a function of the distance to the cluster center or the distance between cluster centers generally get biased by the respective fuzzy method. Hence, using the validation method of the first group which only uses the membership degree is more suitable for us to prove our claim. Two well-known indices in this group are Partition Coefficient (PC) and Classification Entropy (CE).
We have used PC to evaluate the clustering results of FCM and the proposed Fuzzified FCM (F3CM). In fact, the second index calculates the fuzzy level of the membership degrees. Obviously, the level of fuzziness for the FCM method is higher than the level of fuzziness for the proposed method. Also in this paper, we have used a new clustering validation parameter JS, based on the definition of the clustering. In this definition of clustering data are clustered in a way that data which belong to the same cluster have the most similarity to each other and the least similarity to data from the other clusters. Hence, the JS index is defined as the following:

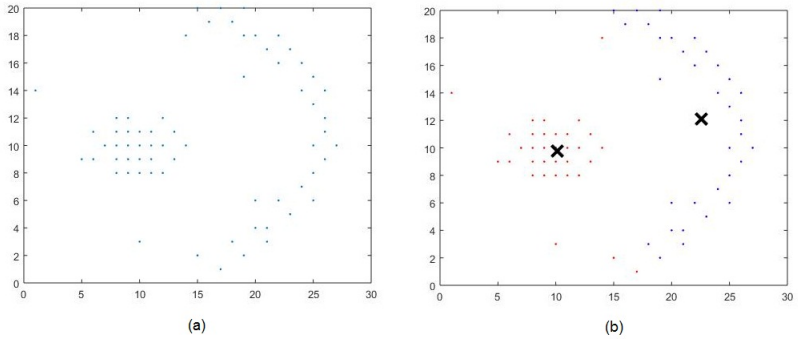


Fig 5. a) The synthetic data set, b) the FCM clustering result on the synthetic data set.

$$JS=\frac{1}{n\times c}\sum_{k=1}^{n}\left(\max_{i}u_{ik}+\sum_{i=1,i\neq \mathrm{argmax}_i\, u_{ik}}^{c}(1-u_{ik})\right), \tag{11}$$

In Equation (11), u_{ik} is the membership degree of the kth data point to the ith cluster, n is the total number of data,

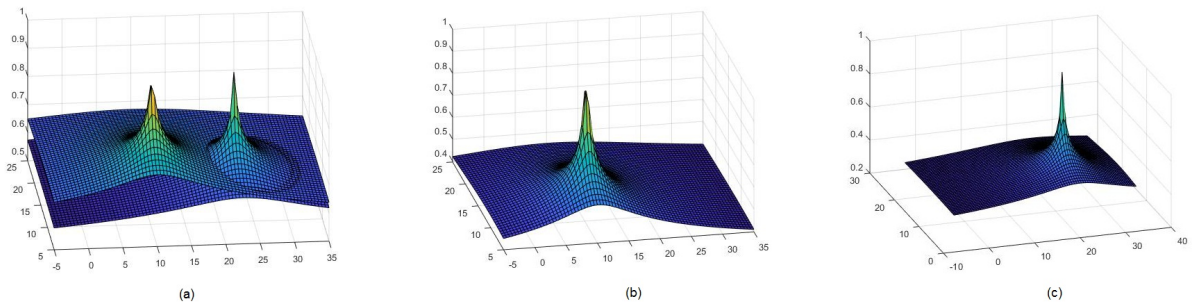


Fig. 6. a) The membership functions of the PCM, c) The membership function of the first (left) cluster, with m=11, d) The membership function of the second (right) cluster, with m=11

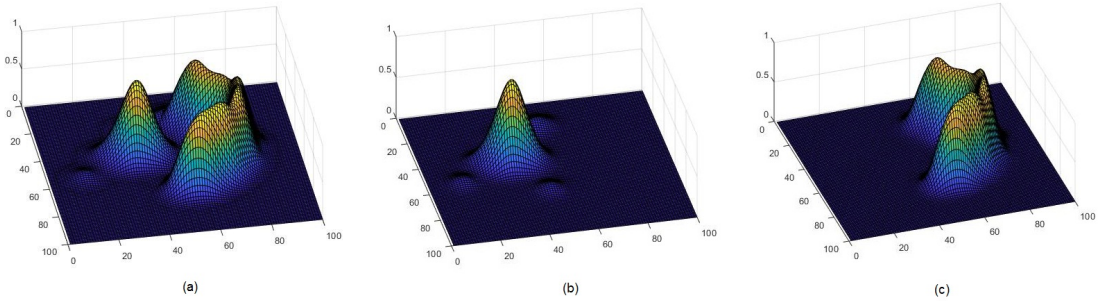


Fig. 7. a) The membership functions of the clusters obtained from the FCM re-fuzzification, b) The membership function of the first (left) cluster, c) The membership function of the second (right) cluster.

and c is the total number of clusters. The JS equation is composed of two terms. The first term expresses similarity among the data of the same cluster, and the second term expresses dissimilarity among data from different clusters.

We have used the real and synthetic data sets for this part of our simulations. The characteristics of data sets are provided in Table 2.

In Table 3 the simulation result on the existing data in Table 2 is shown to compare F3CM, FCM, (kernel-based FCM) KFCM, fuzzified KFCM (FKFCM) and GPFCM. For a fair comparison, when using the two validation parameters PC and JS the condition of "sum of 1" is enforced to the proposed algorithm. Also for the ink drop spread function, the same as FCM's was used (i.e. Euclidean distance). The value of the proposed algorithm's parameter (σ) for each data set is shown in Table 3 as well.

As Table 3 shows the quality of fuzzy clusters resulted from re-fuzzification of FCM compared with the original FCM, KFCM, FKFCM, and GPFCM has increased. In order to do a better comparison, the statistical hypothesis test was done on the results of Table 3. Table 4 shows the result of statistical analysis which was performed by t-test. As Table 4 shows the quality of fuzzy clusters of F3CM algorithm is much better than the other algorithms, however, the quality of F3CM and FKFCM algorithms are almost the same. Therefore, although the results of KFCM is much better than those of FCM and GPFCM, however, our proposed algorithm yields better fuzzy evaluation indices. It should be mentioned that the fuzzified version of KFCM also shows better evaluation than KFCM. Consequently, the proposed algorithm could obtain more reliable and more accurate membership degrees compared to the original algorithm.

4.2 Increasing the quality of clustering

Although our main motivation for re-fuzzifying the FCM algorithm was to obtain membership degrees which are more natural and more fitting with the shape and density of the clusters, however, our simulations show that by using the proposed method the quality of clusters obtained from FCM can be slightly improved in most cases as well. The reason for this is the displacement of some data from one cluster to others. Such displacements occur because of the higher similarity of data to the shape and density of the destination cluster.

In this part, simulations are used to show another advantage of re-fuzzification of FCM and FKFCM over that of original FCM, KFCM, and GPFCM. The algorithms are run over data sets of Table 2. Next, the quality of final results of the algorithms (i.e. the crisp results of the algorithm after defuzzification) are compared based on the several external

Table 2: Characteristics of data sets

	# Objects	# Dim.	# Classes	Source
R15	600	2	15	http://cs.joensuu.fi/sipu/data sets/
Iris	150	4	3	http://archive.ics.uci.edu
Pen-based	10992	16	10	http://archive.ics.uci.edu
Ionosphere	351	33	2	http://archive.ics.uci.edu
Vehicle	846	18	4	http://archive.ics.uci.edu
Waveform	5000	21	3	http://archive.ics.uci.edu
Birch1	100000	2	100	http://cs.joensuu.fi/sipu/data sets/
Birch2	100000	2	100	http://cs.joensuu.fi/sipu/data sets/
Yeast	1484	8	10	http://archive.ics.uci.edu
Satimage	6435	36	7	http://archive.ics.uci.edu
Twonorm	7400	20	2	http://archive.ics.uci.edu
D31	3100	2	31	http://cs.joensuu.fi/sipu/data sets/
Breast	198	33	2	http://archive.ics.uci.edu
Dermatology	358	34	6	http://archive.ics.uci.edu
3D64	64000	3	64	http://ee.sharif.edu/∼acl/Projects/Clustering

Table 3: Comparison between the Fuzzified FCM (F3CM), FCM, KFCM, and GPFCM, based on the fuzzy validation indices.

	PC					JS					Refuzzification Parameter
	F3CM	FCM	KFCM	FKFCM	GPFCM	F3CM	FCM	KFCM	FKFCM	GPFCM	σ
R15	0.9544	0.7914	0.7914	0.7914	0.7148	0.9964	0.9833	0.9833	0.9833	0.9767	0.02
Iris	0.8897	0.7832	0.8617	0.9351	0.3334	0.9481	0.9048	0.9415	0.9712	0.5581	0.07
Pen-based	0.8818	0.2635	0.2643	0.8670	0.2578	0.9843	0.8788	0.8782	0.9813	0.8706	0.07
Ionosphere	0.9951	0.6512	0.6507	0.9951	0.5000	0.9962	0.7383	0.7379	0.9962	0.5000	0.05
Vehicle	0.7349	0.6932	0.4506	0.7509	0.2500	0.9081	0.8985	0.7958	0.9102	0.6250	0.12
Waveform	0.9886	0.4445	0.3333	0.5398	0.3333	0.9947	0.7174	0.5556	0.7618	0.5556	0.1
Birch1	0.7779	0.3236	0.3221	0.7800	0.3023	0.9969	0.9899	0.9899	0.9969	0.9878	0.01
Birch2	0.9358	0.7141	0.7029	0.9565	0.6555	0.9991	0.9931	0.9961	0.9994	0.9905	0.001
Yeast	0.9277	0.1415	0.1002	0.9539	0.1000	0.9898	0.8415	0.8210	0.9935	0.8200	0.01
Satimage	0.7636	0.2335	0.8331	0.9993	0.1667	0.9436	0.7969	0.9691	0.9998	0.7551	0.05
Twonorm	0.7888	0.5000	0.5000	0.7506	0.5000	0.7090	0.5000	0.5000	0.8195	0.5000	0.1
D31	0.8320	0.5341	0.5251	0.8213	0.2938	0.9924	0.9795	0.9789	0.9920	0.9626	0.02
Breast	0.9182	0.5289	0.9245	0.9341	0.5000	0.9441	0.6051	0.9439	0.9530	0.5000	0.05
Dermatology	0.7991	0.4631	0.4445	0.7993	0.1667	0.9512	0.8711	0.8654	0.9518	0.7222	0.1
3D64	0.8508	0.3610	0.3508	0.8745	0.3034	0.9964	0.9851	0.9850	0.9970	0.9743	0.02

Table 4: The statistical hypothesis test with t-test on the results of Table 3.

	FCM	KFCM	F KFCM	GPFCM
PC	+7.1928	5.1117	0.5250	+12.5607
JS	+3.7925	2.8263	0.1581	+4.5336

clustering evaluation measures. External evaluation and internal evaluation measures are two types of clustering quality evaluation. If the evaluation is done based on the data that was itself clustered, it is called an internal evaluation. On the other hand, in an external evaluation, clustering results are evaluated based on data that has well-known class labels. Using internal criteria in cluster evaluation is biased toward algorithms that use the same cluster model. As a result, the best approach for fair evaluations so far is using external evaluation measures [50, 67]. More details on the evaluation metrics can be found in the literature [5]. The measures which we have used in this part of simulations are F-measure, AMI [62], RI, ARI [57], and Accuracy.

In this paper, accuracy is defined as the measure of the homogeneity of clusters with respect to previously known classes (12). This is the same as the definition of precision in [67]. Hence, the best approach for fair evaluations so far

is the external evaluation measures [50, 67].

$$Accuracy_i = 1/n_i max_{j=1}^{k} \{n_{ij}\} \tag{12}$$

The accuracy of clustering C is defined as the weighted sum of the cluster-wise accuracy values (13).

$$Accuracy = \sum_{i=1}^{r} n_i/n Accuracy_i, \tag{13}$$

where the ratio n_i/n denotes the fraction of points in cluster C_i.

Table 4 shows the comparison of the crisp results of the F3CM, FCM, KFCM, FKFCM and GPFCM algorithms. As the results show, the fuzzified versions of FCM (F3CM) and KFCM (FKFCM) algorithms mostly cause the quality of the clustering to be increased. This is because the data points on the borders of the clusters have been moved into their true cluster considering shape and density.

Our main motivation for re-fuzzifying the results of the main algorithm was not better quality clustering, but it was obtaining fuzzy clusters whose membership degrees are more natural and more perfect. However, by applying the proposed algorithm to various data sets we observed that the quality of clusters is improved in most cases as well. Table 5 demonstrates this very well. For a better and more accurate comparison, we have used a statistical hypothesis test method. Table 6 shows the results of a statistical t-test on data from Table 5. As Table 6 shows, the FKFCM results is better crisp clustering than the other algorithms. The crisp clustering results of FKFCM is even better than the F3CM algorithm; however, the crisp results of F3CM algorithm is better than FCM, KFCM and GPFCM algorithms. As a result, the proposed algorithm absolutely improves the clustering quality of the FCM and KFCM.

4.3 Ability to recognize outliers and noise data

In this part, we first compare the F3CM and FCM algorithms with respect to their ability to eliminate noise data. Then we will discuss the noise elimination of F3CM with respect to the threshold parameter. To do this part of simulation we inject noise on two data sets R15 and Iris and use their ground truth and external measures for better comparison. The percentage of detected noise, the percentage of false negatives and the AMI index are used for comparison. It should be noted that, in noisy data sets, F-measure is not an appropriate metric to draw a comparison among clustering algorithms because it only takes into account already clustered data points; hence AMI is employed because it tries to quantify the amount of shared information between the resulting clusters and the true clusters.

Figure 8 and figure 9 show the ability of F3CM algorithm in comparison with the FCM algorithm in noisy environments. In figure 7, 200 percent noise is added to the R15 data set. Next, to eliminate noise by changing the threshold parameter on membership degree, the percentage of detected noise, the percentage of data that have been mistakenly recognized as noise, and also the AMI value have been measured. As it is shown in figure 8 the detected percentage of noise for F3CM algorithm increases as threshold increases. While the percentage of false negatives in low threshold values is almost zero, as the threshold passes above a certain value, the percentage of false negatives starts to increase.

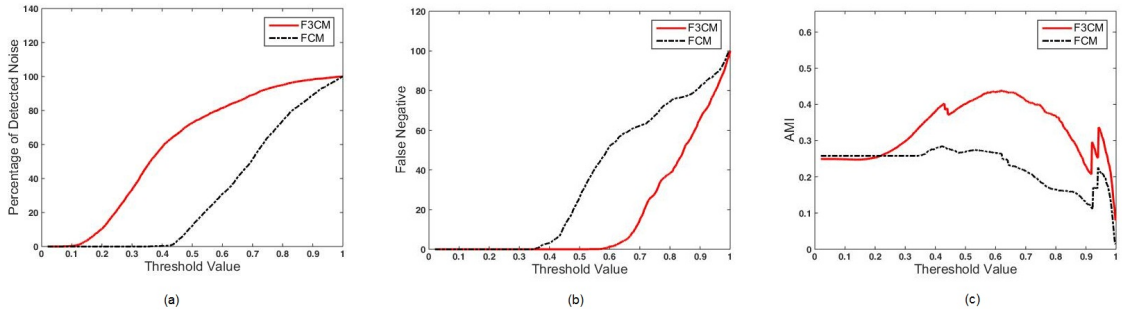


Fig. 8. Comparison of the proposed F3CM algorithm with FCM algorithm while 200 percent white noise is injected to the R15 data set, shows that the proposed algorithm works better than FCM in the presence of noise. (a) The percentage of detected noise with respect to the membership threshold value. (b) The percentage of false negatives versus the membership threshold. (c) The AMI measure versus the membership threshold.

The AMI value shows the above two changes, in the way that, in the beginning when the noise data are being eliminated AMI has an increasing trend, and at the end when the cluster data are being mistakenly separated from the clusters

Table 5: Comparison of the crisp results of the F3CM, FCM, KFCM, FKFCM and GPFCM

	Algorithm	F-measure	AMI	RI	ARI	Accuracy	parameter
R15	F3CM	0.9967	0.9938	0.9991	0.9928	0.9967	$\sigma = 0.02$
	FCM	0.9967	0.9938	0.9991	0.9928	0.9967	
	KFCM	0.9967	0.9938	0.9991	0.9928	0.9967	$\sigma^2 = 100$
	FKFCM	0.9967	0.9938	0.9991	0.9928	0.9967	$\sigma = 0.02$
	GPFCM	0.9967	0.9938	0.9991	0.9928	0.9967	
Iris	F3CM	0.8983	0.7667	0.8859	0.7445	0.9169	$\sigma = 0.07$
	FCM	0.8923	0.7419	0.8797	0.7294	0.9028	
	KFCM	0.9466	0.8299	0.9341	0.8508	0.9471	$\sigma^2 = 1$
	FKFCM	0.96	0.8623	0.9495	0.8857	0.9605	$\sigma = 0.05$
	GPFCM	0.611	0.591	0.7777	0.5609	0.8403	
Pen-based	F3CM	0.2609	0.2809	0.523	0.0871	0.8074	$\sigma = 0.07$
	FCM	0.2543	0.2597	0.5152	0.0723	0.7999	
	KFCM	0.265	0.2572	0.518	0.0765	0.7829	$\sigma^2 = 100$
	FKFCM	0.3084	0.2792	0.5238	0.876	0.8130	$\sigma = 0.07$
	GPFCM	0.2412	0.2555	0.5132	0.0687	0.7990	
Ionosphere	F3CM	0.7544	0.3287	0.629	0.2562	0.7911	$\sigma = 0.01$
	FCM	0.6991	0.1246	0.5865	0.1727	0.6985	
	KFCM	0.6991	0.1246	0.5865	0.1727	0.6985	$\sigma^2 = 100$
	FKFCM	0.7249	0.1669	0.6067	0.2135	0.7266	$\sigma = 0.03$
	GPFCM	0.6498	0.0807	0.5469	0.094	0.6598	
Vehicle	F3CM	0.4392	0.1798	0.6603	0.1297	0.4505	$\sigma = 0.12$
	FCM	0.4368	0.1716	0.6506	0.1182	0.4538	
	KFCM	0.4339	0.1707	0.6709	0.1132	0.4410	$\sigma^2 = 144$
	FKFCM	0.4346	0.1613	0.6691	0.1257	0.4421	$\sigma = 0.12$
	GPFCM	0.4039	0.1023	0.6547	0.0976	0.4093	
Waveform	F3CM	0.5314	0.3319	0.6627	0.2433	0.5329	$\sigma = 0.02$
	FCM	0.53	0.3209	0.66	0.2364	0.5311	
	KFCM	0.5276	0.1253	0.5844	0.1052	0.5700	$\sigma^2 = 1$
	FKFCM	0.6363	0.2864	0.6738	0.2823	0.6655	$\sigma = 0.1$
	GPFCM	0.4771	0.0505	0.579	0.0548	0.4803	
Birch1	F3CM	0.834	0.9374	0.9957	0.7044	0.8125	$\sigma = 0.01$
	FCM	0.816	0.8683	0.9937	0.6956	0.8070	
	KFCM	0.7934	0.8695	0.9937	0.6971	0.8224	$\sigma^2 = 4$
	FKFCM	0.7957	0.8713	0.9938	0.7018	0.8250	$\sigma = 0.01$
	GPFCM	0.7989	0.8533	0.9912	0.6949	0.8038	
Birch2	F3CM	0.8487	0.9438	0.9961	0.8161	0.9128	$\sigma = 0.001$
	FCM	0.8489	0.9371	0.9956	0.7946	0.8482	
	KFCM	0.8487	0.9429	0.996	0.8144	0.9165	$\sigma^2 = 100$
	FKFCM	0.8486	0.9442	0.9961	0.8198	0.9167	$\sigma = 0.001$
	GPFCM	0.8332	0.9354	0.9953	0.7949	0.8233	
Yeast	F3CM	0.3199	0.2082	0.7334	0.1438	0.5122	$\sigma = 0.3$
	FCM	0.2393	0.1436	0.7172	0.0913	0.4260	
	KFCM	0.155	0.1116	0.6356	0.0915	0.4610	$\sigma^2 = 100$
	FKFCM	0.2982	0.2006	0.7083	0.1363	0.4918	$\sigma = 0.3$
	GPFCM	0.2082	0.1224	0.6734	0.0687	0.4732	
Satimage	F3CM	0.4973	0.2956	0.5102	0.1877	0.8995	$\sigma = 0.2$
	FCM	0.4116	0.2274	0.4075	0.0836	0.8995	
	KFCM	0.3609	0.2658	0.7913	0.1265	0.8350	$\sigma^2 = 1$
	FKFCM	0.57	0.325	0.8217	0.3041	0.7913	$\sigma = 0.2$
	GPFCM	0.2772	0.0242	0.3412	0.0073	0.8810	

	Algorithm	F-measure	AMI	RI	ARI	Accuracy	parameter
Twonorm	F3CM	0.6864	0.1579	0.5804	0.1024	0.9427	$\sigma = 0.5$
	FCM	0.6534	0.1019	0.5064	0.013	0.9427	
	KFCM	0.636	0.1633	0.5941	0.1165	0.9160	$\sigma^2 = 1$
	FKFCM	0.936	0.721	0.9715	0.8577	0.9181	$\sigma = 0.$
	GPFCM	0.6529	0.0004	0.5016	0.0009	0.9425	
D31	F3CM	0.8773	0.9166	0.9888	0.8266	0.9106	$\sigma = 0.02$
	FCM	0.8744	0.911	0.9884	0.8196	0.9048	
	KFCM	0.716	0.9084	0.9881	0.8162	0.9049	$\sigma^2 = 100$
	FKFCM	0.8735	0.9126	0.9883	0.8206	0.9100	$\sigma = 0.02$
	GPFCM	0.6232	0.8033	0.9685	0.6032	0.7888	
Breast	F3CM	0.6022	0.0275	0.518	0.0366	0.7582	$\sigma = 0.01$
	FCM	0.6021	0.016	0.5123	0.0249	0.7584	
	KFCM	0.6052	0.0289	0.5037	0.0047	0.7734	$\sigma^2 = 1$
	FKFCM	0.6021	0.0353	0.5223	0.0452	0.7577	$\sigma = 0.5$
	GPFCM	0.6039	0.0033	0.4979	0.0029	0.7624	
Dermatology	F3CM	0.7153	0.6302	0.8534	0.5693	0.8063	$\sigma = 0.6$
	FCM	0.3222	0.0838	0.6986	0.028	0.3860	
	KFCM	0.3492	0.0882	0.703	0.0359	0.4054	$\sigma^2 = 500$
	FKFCM	0.7012	0.6677	0.8695	0.6249	0.8050	$\sigma = 0.6$
	GPFCM	0.3578	0.2531	0.709	0.2024	0.4523	
3D64	F3CM	0.8513	0.9068	0.9923	0.7672	0.8319	$\sigma = 0.05$
	FCM	0.8172	0.8767	0.9901	0.7019	0.7932	
	KFCM	0.8395	0.9165	0.9929	0.7838	0.8920	$\sigma^2 = 4$
	FKFCM	0.8426	0.9245	0.9932	0.7941	0.8979	$\sigma = 0.02$
	GPFCM	0.8182	0.8621	0.992	0.7212	0.8111	

Table 6: The statistical hypothesis test with t-test on the results of Table 5.

	F-measure	AMI	RI	ARI	Accuracy
FCM	+1.8589	+2.0788	+2.3792	+1.9732	+1.7566
KFCM	+2.2956	+1.9070	+0.1021	+1.4614	+1.2324
FKFCM	-1.4536	-0.7301	-1.5718	-1.4351	-0.1705
GPFCM	+3.4326	+4.3890	+3.6157	+4.0453	+2.7074

as noise, AMI has a decreasing trend. Contrary to F3CM, in FCM the percentage of detected noise and false negatives both begin to increase after a certain threshold value. In other words, by increasing the threshold, not only noise data exit the clusters but also cluster data mistakenly leave clusters as well. This is not an ideal situation.

In figure 9, 100 percent noise is injected to the Iris data set. The value of the AMI for the proposed F3CM algorithm is better than the FCM algorithm for almost all values of the membership threshold. Figure 9(c) shows that at first the AMI index increases because the percentage of detected noise increases as shown in figure 9(a). Meanwhile, as figure 9(b) shows the percentage of false negatives is almost zero. As the percentage of false negatives increases, the increasing trend of AMI slows down and eventually begins to decrease. This is because by further increasing the membership threshold value, the percentage of false negatives rapidly increases. As a result, the proposed algorithm improves the quality of the FCM clustering algorithm in noisy environment.

In these experiments, we used Gaussian ink drop. The values of standard deviation for R15 and Iris data sets are 0.02 and 0.1 respectively.

4.4 Evaluating sensitivity with respect to the parameter

To evaluate the sensitivity of the algorithm with respect to the parameter, an experiment has been done on the Vehicle data set. This is done in the way of changing parameter σ from 0.002 to 1 in 0.002 steps while plotting the AMI value in each step.

Figure 10 shows three curves. Curve No. 1 is the AMI value resulted from our proposed algorithm, while the final result of the FCM clustering is considered as ground truth. Curves No. 2 and No.3 show the value of AMI resulted from our proposed algorithm and FCM algorithm respectively, while the real ground truth is considered. Since the σ parameter does not have any effects on FCM, its AMI value is held constant. Figure 10 also shows a zoomed picture for parts of curves 2 and 3. Comparing curve 2 against curve 3 shows that in low values of σ the result of F3CM clustering

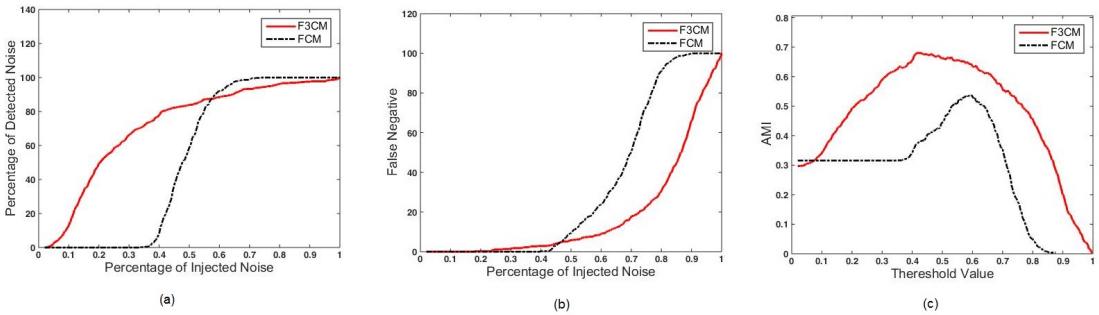


Fig. 9. Comparison of the proposed F3CM algorithm with FCM algorithm, while 100 percent white noise is injected to the Iris data set, shows that the proposed algorithm works better than FCM in the presence of noise. (a) The percentage of detected noise with respect to the membership threshold value. (b) The percentage of false negatives versus membership threshold. (c) The AMI measure versus membership threshold.

has a lower quality with respect to that of FCM. This is because if σ has a small value, adjacent ink drops do not affect each other. Hence, the fuzzification of the clustering will not express the shape and density of the data sets. By increasing the value of σ the quality of F3CM clustering also increases with respect to that of FCM. The appropriate value for the σ parameter must be selected from this region. By more increase in the value of σ, the quality of F3CM clustering diminishes. This is because with a disproportionate increase in the value of σ adjacent clusters will overlap which causes a decrease in the AMI index. The reason being a cluster may dominate adjacent clusters. In other words, it causes data to be displaced among clusters inappropriately which in turn decreases the quality of the clustering.

4.5 Parameter selection method for the proposed algorithm

As mentioned earlier the appropriate region of σ value is where the quality of F3CM clustering is higher than FCM clustering. As figure 9 shows the quality of F3CM is better than FCM for a wide range of σ parameter. However, we propose a method to determine a reliable value for this parameter. One proposed method for selecting this parameter without knowing the ground truth is as follows. First, we consider the FCM clustering as ground truth, then plot the AMI curve with respect to σ. The region where σ has yielded the highest values for AMI determines the appropriate values for this parameter. Figure 10 shows an appropriate value for this parameter. Although our proposed method

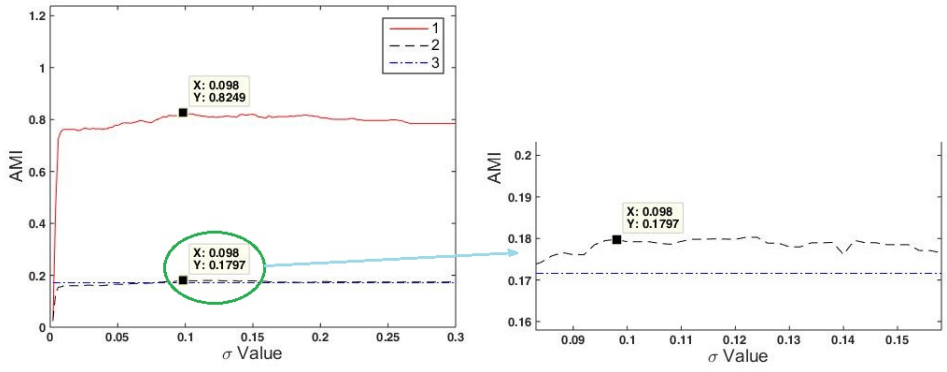


Fig. 10. Parameter sensitivity analysis for the proposed algorithm, by applying the algorithm on Vehicle data set for different values of the algorithm parameter.

suggests a suitable value of σ in most cases, sometimes it is not the best value. For instance, in the Iris data set the proposed value for σ is $\sigma = 0.13$ (Figure 11) which is not located in the optimal region, although the proposed method eventually can find a value whose clustering quality is not lower than the FCM algorithm. Also, our main claim in this paper is presenting fuzzy clusters in which membership degrees of the data is based on the shape and density, and the above method suggests a suitable value for this claim. This is because the above method shows a range of sigma values in which the maximum matching exists between F3CM and FCM algorithms.

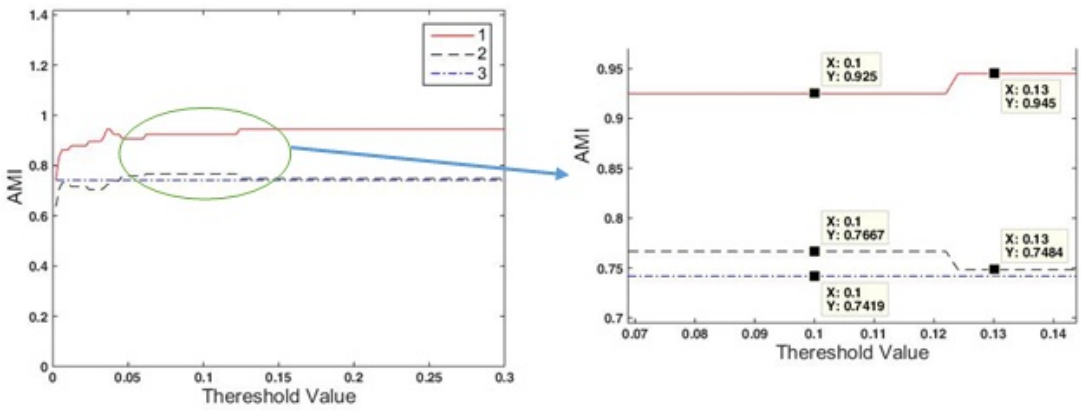


Fig. 11. Parameter sensitivity analysis for the proposed algorithm, by applying the algorithm on Iris data set for different values of the algorithm parameter.

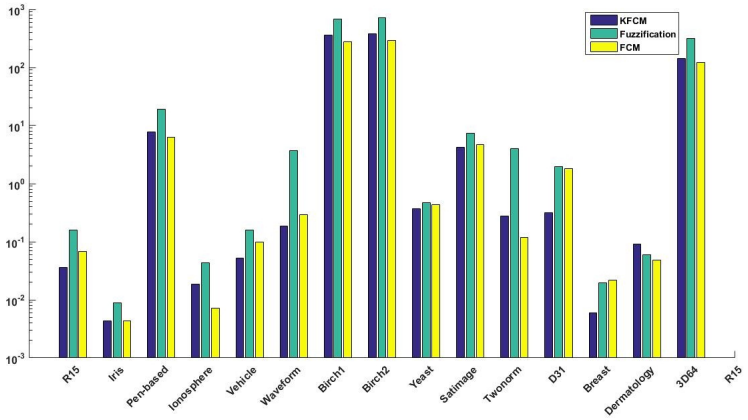


Fig. 12. The execution Time comparison of the three algorithms, KFCM, Refuuzzification and FCM.

4.6 Execution time comparison

In this part we compare the speed of implementation of different algorithms in terms of actual CPU execution time in second. It should be noted that the proposed algorithm is a post processing algorithm and the computational time of the algorithm will be added to the computational time of the original fuzzy clustering algorithm. Figure 12 shows the average execution time of the KFCM, FCM and the proposed Refuzzification algorithm for all the datasets listed in Table 2 with a logarithmic y-axis plot. As our simulations shows the execution time of the proposed algorithm is from 0.6 to 30 times more than the original fuzzy clustering algorithms depending on the size of the datasets. All simulations were conducted on a personal computer with the following specifications: Pentium CORE^{TM} 2 Due CPU, 64-bit, 2.20GHz, 4GB RAM.

5 Conclusion

In this paper we have proposed a method for "re-fuzzification" of the most well-known and widely used fuzzy clustering algorithm (i.e. FCM) which leads to three improvements:

1. Obtaining membership degrees based on the shape and density of clusters which are more real, natural, and informative. Obtaining better membership degrees is our motivation as well as the main advantage of our proposed algorithm.

2. Improvement of the clustering results of the FCM algorithm after defuzzification in most cases.

3. Resolving FCM problems in recognizing noise and outlier data.

The improvements occur because the membership degrees of each data to each cluster is assigned based on the shape and the density of the cluster. However, in the FCM algorithm, membership degrees are assigned based on a reverse function of the distance to the cluster centers. It is worth mentioning that in the proposed algorithm, no changes are made to FCM, but a post-process is performed on the output results of FCM. Also, simulations were done to demonstrate the above benefits. However these improvements achieved by adding the time complexity in the order of $O(n^2p)$ (where n is the total number of data points, and p is the feature size) to the original fuzzy clustering algorithm.

In order to show that membership degrees obtained from our proposed method are better than that of FCM, KFCM, and GPFCM, we have used a fuzzy index (PC) as well as a newer index called JS. The simulations of the first part were performed on both real and synthetic data sets.

In the second part of the simulations, improved accuracy of FCM in clustering was evaluated. Although our main motivation for re-fuzzification of FCM was not to improve the accuracy of FCM, but to present membership degrees which are more natural and more in line with the shape and density of the clusters. Statistical hypothesis tests show that the proposed algorithm can improve the quality of FCM and KFCM as well.

Also in our simulations, we showed that the proposed algorithm has the ability of eliminating the noise and outlier data from the FCM algorithm. Finally, we analyzed the sensitivity of the proposed algorithm with respect to its only parameter σ; then, we proposed a method for determining the proper value for this parameter. As the simulations show, our proposed algorithm works well for a wide range of its parameter values.

References

[1] Q. Abbas, *Segmentation of differential structures on computed tomography images for diagnosis lung-related diseases*, Biomedical Signal Processing and Control, **33** (2017), 325-334.

[2] A. A. Abin, H. Beigy, *Active constrained fuzzy clustering: A multiple kernels learning approach*, Pattern Recognition, **48**(3) (2015), 953-967.

[3] I. E. P. Afrakoti, S. B. Shouraki, F. M. Bayat, M. Gholami, *Using a memristor crossbar structure to implement a novel adaptive real-time fuzzy modeling algorithm*, Fuzzy Sets and Systems, **307** (2017), 115-128.

[4] C. C. Aggarwal, *Data mining: The textbook*, Springer International Publishing, 2015.

[5] C. C. Aggarwal, C. K. Reddy, *Data clustering: Algorithms and applications*, Publisher of Humanities, Social Science and STEM Books, CRC Press, 2013.

[6] A. Ansari, A. Riasi, *Customer clustering using a combination of fuzzy c-means and genetic algorithms*, International Journal of Business and Management, **11**(7) (2016), 59.

[7] S. Askari, N. Montazerin, M. F. Zarandi, *Generalized possibilistic fuzzy c-means with novel cluster validity indices for clustering noisy data*, Applied Soft Computing, **53** (2017), 262-283.

[8] C. Bai, D. Dhavale, J. Sarkis, *Complex investment decisions using rough set and fuzzy c-means: An example of investment in green supply chains*, European Journal of Operational Research, **248**(2) (2016), 507-521.

[9] I. Berget, B. H. Mevik, T. Næs, *New modifications and applications of fuzzy c-means methodology*, Computational Statistics and Data Analysis, **52**(5) (2008), 2403-2418.

[10] J. C. Bezdek, R. Ehrlich, W. Full, FCM: *The fuzzy c-means clustering algorithm*, Computers and Geosciences, **10**(2-3) (1984), 191-203.

[11] B. Biswal, P. K. Dash, B. K. Panigrahi, *Power quality disturbance classification using fuzzy c-means algorithm and adaptive particle swarm optimization*, IEEE Transactions on Industrial Electronics, **56**(1) (2009), 212-220.

[12] C. Budayan, I. Dikmen, M. T. Birgonul, *Comparing the performance of traditional cluster analysis, self-organizing maps and fuzzy c-means method for strategic grouping*, Expert Systems with Applications, **36**(9) (2009), 11772-11781.

[13] S. Das, S. De, *A modified genetic algorithm based FCM clustering algorithm for magnetic resonance image segmentation*, in Proceedings of the 5th International Conference on Frontiers in Intelligent Computing: Theory and Applications, Springer, 2017.

[14] J. De Andrés, P. Lorca, F. Javierde Cos Juez, F. Sánchez Lasheras, *Bankruptcy forecasting: A hybrid approach using fuzzy c-means clustering and multivariate adaptive regression splines (MARS)*, Expert Systems with Applications, **38**(3) (2011), 1866-1875.

[15] A. P. Dempster, N. M. Laird, D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, Journal of the Royal Statistical Society, Series B (methodological), (1977), 1-38.

[16] M. B. Dowlatshahi, H. Nezamabadi-Pour, GGSA: *A grouping gravitational search algorithm for data clustering*, Engineering Applications of Artificial Intelligence, **36** (2014), 114-121.

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

[18] H. K. M. Ester, J. Sander, X. Xu. *A density-based algorithm for discovering clusters in large spatial databases with noise*, Proceedings of the Second International Conference on Knowledge Discovery and Data MiningAugust, (1996), 226-231.

[19] D. Graves, W. Pedrycz, *Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study*, Fuzzy Sets and Systems, **161**(4) (2010), 522-543.

[20] R. L. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar, R. Namburu, *Data mining for scientific and engineering applications*, Springer Science and Business Media, **2** (2013).

[21] D. E. Gustafson, W. C. Kessel, *Scientific systems*, Inc. 186 Alewife Brook Parlcway Cambridge, Massachusetts 02138, (1979).

[22] M. Hanesch, R. Scholger, M. Dekkers, *The application of fuzzy c-means cluster analysis and non-linear mapping to a soil data set for the detection of polluted sites*, Physics and Chemistry of the Earth, Part A: Solid Earth and Geodesy, **26**(11) (2001), 885-891.

[23] A. Hinneburg, D. A. Keim, *An efficient approach to clustering in large multimedia databases with noise*, KDD'98: Proceedings of the Fourth International Conference on Knowledge Discovery and Data MiningAugust, (1998), 58-65.

[24] A. Hinneburg, H. H. Gabriel, *Denclue 2.0: Fast clustering based on kernel density estimation*, in IDA, Springer, Berlin, Heidelberg, 2007.

[25] H. C. Huang, Y. Y. Chuang, C. S. Chen, *Multiple kernel fuzzy clustering*, IEEE Transactions on Fuzzy Systems, **20**(1) (2012), 120-134.

[26] M. Z. Islam, et al., *Combining k-means and a genetic algorithm through a novel arrangement of genetic operators for high quality clustering*, Expert Systems with Applications, **91** (2018), 402-417.

[27] M. Javadian, S. B. Shouraki, S. S. Kourabbaslou, *A novel density-based fuzzy clustering algorithm for low dimensional feature space*, Fuzzy Sets and Systems, **318** (2017), 34-55.

[28] M. Javadian, S. B. Shouraki, *UALM: Unsupervised active learning method for clustering low-dimensional data*, Journal of Intelligent and Fuzzy Systems, **32**(3) (2017), 2393-2411.

[29] B. Jayaram, F. Klawonn, *Can fuzzy clustering avoid local minima and undesired partitions?* Computational Intelligence in Intelligent Data Analysis, (2013), 31-44.

[30] X. L. Jiang, et al., *Robust level set image segmentation algorithm using local correntropy-based fuzzy c-means clustering with spatial constraints*, Neurocomputing, **207** (2016), 22-35.

[31] S. Kannan, et al., *Effective fuzzy possibilistic c-means: An analyzing cancer medical database*, Soft Computing, (2016), 1-11.

[32] L. Kaufman, P. Rousseeuw, *Clustering by means of medoids*, Faculty of Mathematics and Informatics, North-Holland, 1987.

[33] U. Kaymak, M. Setnes, *Fuzzy clustering with volume prototypes and adaptive cluster merging*, IEEE Transactions on Fuzzy Systems, **10**(6) (2002), 705-712.

[34] S. H. Klidbary, S. B. Shouraki, *A novel adaptive learning algorithm for low-dimensional feature space using memristor-crossbar implementation and on-chip training*, Applied Intelligence, **48**(11) (2018), 4174-4191.

[35] S. H. Klidbary, S. B. Shouraki, B. Linares-Barranco, *Digital hardware realization of a novel adaptive ink drop spread operator and its application in modeling and classification and on-chip training*, International Journal of Machine Learning and Cybernetics, (2018), 1-21.

[36] S. H. Klidbary, S. B. Shouraki, I. E. P. Afrakoti, *An adaptive efficient memristive ink drop spread (IDS) computing system*, Neural Computing and Applications, (2018), 1-22.

[37] S. H. Klidbary, et al. *Outlier robust fuzzy active learning method (ALM)*, in 2017, 7th International Conference on Computer and Knowledge Engineering (ICCKE). IEEE, (2017).

[38] R. Krishnapuram, A. Joshi, L. Yi, *A fuzzy relative of the k-medoids algorithm with application to web document and snippet clustering*, in Fuzzy Systems Conference Proceedings, FUZZ-IEEE'99, (1999).

[39] L. Liu, et al., *A modified fuzzy c-means (FCM) clustering algorithm and its application on carbonate fluid identification*, Journal of Applied Geophysics, **129** (2016), 28-35.

[40] J. MacQueen, *Some methods for classification and analysis of multivariate observations*, in Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. Oakland, CA, USA, (1967).

[41] O. P. Mahela, A. G. Shaik, *Recognition of power quality disturbances using S-transform based ruled decision tree and fuzzy c-means clustering classifiers*, Applied Soft Computing, **59** (2017), 243-257.

[42] U. Maulik, S. Bandyopadhyay, *Fuzzy partitioning using a real-coded variable-length genetic algorithm for pixel classification*, IEEE Transactions on Geoscience and Remote Sensing, **41**(5) (2003), 1075-1081.

[43] J. P. Mei, L. Chen, *Fuzzy clustering with weighted medoids for relational data*, Pattern Recognition, **43**(5) (2010), 1964-1974.

[44] F. Merrikh-Bayat, S. B. Shouraki, *The neuro-fuzzy computing system with the capacity of implementation on a memristor crossbar and optimization-free hardware training*, EEE Transactions on Fuzzy Systems, **22**(5) (2014), 1272-1287.

[45] J. Nayak, B. Naik, H. Behera, *Fuzzy c-means (FCM) clustering algorithm*, A decade review from 2000 to 2014, in Computational Intelligence in Data Mining-Volume 2., Springer, (2015), 133-149.

[46] J. V. de Oliveira, W. Pedrycz, *Advances in fuzzy clustering and its applications*, Wiley Online Library, 2007.

[47] N. R. Pal, et al., *A possibilistic fuzzy c-means clustering algorithm*, IEEE Transactions on Fuzzy Systems, **13**(4) (2005), 517-530.

[48] H. X. Pei, et al., *D-FCM: Density based fuzzy c-means clustering algorithm with application in medical image segmentation*, Procedia Computer Science, **122** (2017), 407-414.

[49] P. Qian, et al., *Knowledge-leveraged transfer fuzzy c-means for texture image segmentation with self-adaptive cluster prototype matching*, Knowledge-Based Systems, 2017.

[50] E. Rendón, et al., *Internal versus external cluster validation indexes*, International Journal of Computers and Communications, **5**(1) (2011), 27-34.

[51] M. J. Rezaee, M. Jozmaleki, M. Valipour, *Integrating dynamic fuzzy c-means, data envelopment analysis and artificial neural network to online prediction performance of companies in stock exchange*, Physica A: Statistical Mechanics and Its Applications, 2017.

[52] B. S. Saeed, N. A, Honda, *New method for establishing and saving fuzzy membership functions*, Proc. Of Fuzzy System Symposium, **13** (1997), 91-94.

[53] T. Sajana, C. S. Rani, K. Narayana, *A survey on clustering techniques for big data mining*, Indian Journal of Science and Technology, **9**(3) (2016), 1-12.

[54] S. B. Shouraki, *A novel fuzzy approach to modeling and control and its hardware implementation based on brain functionality and specifications*, in Ph.D. Dissertation. The University of Electro-Communications, Chofu, Japan, March 2000.

[55] S. B. Shouraki, N. Honda. *Simulation of brain learning process through a novel fuzzy hardware approach*, In Proceeding of International Conference on Systems, Man Cybernetics (SMC'99), Tokyo, Japan, IEEE, 1999.

[56] A. Smiti, Z. Eloudi. *Soft dbscan: Improving dbscan clustering method using fuzzy set theory*, in 2013 6th International Conference on Human System Interactions (HSI). IEEE, 2013.

[57] D. Steinley, *Properties of the Hubert-Arable adjusted rand index*, Psychological Methods, **9**(3) (2004), 386.

[58] K. Sudha, Y. B. Raju, A. C. Sekhar, *Fuzzy c-means clustering for robust decentralized load frequency control of interconnected power system with generation rate constraint*, International Journal of Electrical Power and Energy Systems, **37**(1) (2012), 58-66.

[59] P. Teppola, S. P. Mujunen, P. Minkkinen, *Adaptive fuzzy c-means clustering in process monitoring*, Chemometrics and Intelligent Laboratory Systems, **45**(1) (1999), 23-38.

[60] T. M. Tuan, *Dental segmentation from X-ray images using semi-supervised fuzzy clustering with spatial constraints*, Engineering Applications of Artificial Intelligence, **59** (2017), 186-195.

[61] H. Verma, R. Agrawal, A. Sharan, *An improved intuitionistic fuzzy c-means clustering algorithm incorporating local information for brain image segmentation*, Applied Soft Computing, **46** (2016), 543-557.

[62] N. X. Vinh, J. Epps, J. Bailey, *Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance*, The Journal of Machine Learning Research, **11** (2010), 2837-2854.

[63] W. Wang, Y. Zhang, *On fuzzy cluster validity indices*, Fuzzy Sets and Systems, **158**(19) (2007), 2095-2117.

[64] K. L. Wu, M. S. Yang, *Alternative c-means clustering algorithms*, Pattern Recognition, **35**(10) (2002), 2267-2278.

[65] Z. S. Younus, et al., *Content-based image retrieval using PSO and k-means clustering algorithm*, Arabian Journal of Geosciences, **8**(8) (2015), 6211-6224.

[66] L. A. Zadeh, *Fuzzy sets*, Information and Control, **8**(3) (1965), 338-353.

[67] M. J. Zaki, W. Meira Jr, W. Meira, *Data mining and analysis: Fundamental concepts and algorithms*, Cambridge University Press, 2014.

[68] M. F. Zarandi, M. Faraji, M. Karbasian, *An exponential cluster validity index for fuzzy clustering with crisp and fuzzy data*, Scientia Iranica, Transaction E, Industrial Engineering, **17**(2) (2010), 95.