Using Feature and Cluster Weighted FCM for Reducing ANFIS Rules

Solmaz Abdollahizad
Department of Computer Engineering, Sardroud Center, Tabriz Branch, Islamic Azad University, Tabriz, Iran
Email: Solmaz.abdollahizad@iau.ac.ir

Receive Date: 10 June 2022, Revise Date: 15 July 2022, Accept Date: 20 August 2022

Abstract

Fuzzy c-means (FCM) is assumed that all the features are of equal importance. In real applications, however, the importance of the features is different and there exist some features that are more important than the others. These important features should basically have more effects than the other features in the forming of optimal clusters. The basic FCM algorithm does not support this idea. Also, the FCM algorithm suffers from another problem; the algorithm is very sensitive to initialization, whereas a bad initialization leads to a poor local optimum. In this paper, motivated by these weaknesses of the FCM, the goal is to solve the two problems at the same time. In doing so, an automatic local feature weighting scheme is proposed to properly weight the features of each clusters. And, a cluster weighting process is performed to mitigate the initialization sensitivity of the FCM. Feature weighting and cluster weighting are performed simultaneously and automatically during the clustering process resulting in high quality clusters, regardless of the initial centers. Extensive experiments conducted on a synthetic dataset and 16 real world datasets indicate that the proposed algorithm outperforms the state-of-the-arts algorithms. The convergence proof of the proposed algorithm is also provided

Keywords: Fuzzy c-means, Clustering, Feature, ANFIS

1. Introduction

Automatic data partitioning-clustering is one of the most important tools in data mining, and the related areas such as machine learning, machine vision and pattern recognition [1,2]. The goal is to divide a set of samples into homogeneous partitions (clusters) based on an objective function, so as to increase intra cluster similarity and decrease inter-cluster similarity. In recent decades, various algorithms have been introduced for clustering of which k-means [3] and fuzzy c-means (FCM) [4] are the most well-known algorithms among the rest. Some other versions of these algorithms have also been proposed (more details and evaluations can be found in [5–9]). However, these two algorithms have always drawn the interest of researchers due to some key ideas that they possess; simple structure, easy implementation, fast convergence, and need for low storage capacity to name.

When the boundaries among the clusters are vague, FCM demonstrates a better performance than k-means [10,11] and has shown successful results in various applications such as [12–20]. This is due to the fuzzy nature of this method and thus, it applies soft-clustering in contrast to k-means that does hard clustering. Furthermore, the fuzzy membership matrix in this method provides us a better understanding of the complicated relationships between the clusters and the samples [21]. In most real world applications, obviously, some of the extracted features are irrelevant to the target problem or are less important than the others. FCM and k-means algorithms however,do not take the importance of the features into account, and therefore, their performances drop markedly in some applications [22].

In addition, these algorithms are sensitive to the initial cluster points, whereas a bad initialization may lead to a poor local optimum [23,24].

In order to take the advantages of the important features over the others in the clusters formation, one may apply a weighting scheme on the features [9,25,26]. There are two groups of methods for doing so. The first group includes the algorithms that assign weights to the features globally. That is, a given feature is assigned with only one weight along all the clusters. The methods in the second group assign local weights to the features, and thus, a given feature has different weights in different clusters. This scheme has shown a better performance than the global weighting [9]. For example, the performance of a global weighting method presented in [27] was improved by Chan et al. [28] employing a local weight scheme. For handling the sensitivity of FCM to initialization, there are also two groups of approaches. In the first group, the policy is to systematically prevent the formation of poor-quality clusters during the algorithm restarts (repeats). While in the second group, the policy is to make the algorithms independent of a random initialization, hence they do not need any restarting procedure. For example, in global k-means algorithm [29] and some of its modifications such as methods in [30,31] the process starts from a single cluster, and incrementally other clusters are added according to a criterion. In fact, at each step of algorithm run, a new cluster is added to the solution. Despite the fact that these algorithms are somehow able to mitigate the sensitivity of the algorithms to the initialization, their computational cost in general is very high.

In this paper, in order to overcome the above mentioned problems simultaneously, we propose a new FCM clustering algorithm based on a combination of feature weighting and cluster weighting. Feature weighting is carried out locally and automatically during the clustering procedure. At the same time, in order to deal with the problem of initialization, we apply cluster weighting as well. The weights of the clusters are calculated at each restart based on the sum of their members' intra-cluster distances and member's current features' weights. Calculating the weights of the clusters in this way prevents the creation of clusters that have a large sum of intra-cluster weighted-feature distances (SIWD). And consequently, allows high quality clusters to be formed regardless of the initial centers. In addition to the combination of feature weighting and cluster weighting, an objective function based on a non-Euclidean distance metric is used in the proposed method. The advantage of using this distance metric is that the noise and outliers cannot affect the feature weighting process very much [32,33]. The contributions of this work are threefold:

(1) Feature weighting is carried out locally in a way that the features have different weights depending on their importance in the clusters, thus increasing the quality of clustering;

(2) The weight of the clusters is calculated dynamically while taking the importance of the features in each cluster into consideration;

(3) Feature weighting and cluster weighting are performed simultaneously and automatically during the clustering process resulting in high quality clusters regardless of the initial centers.

The performance of the proposed algorithm is evaluated over a large number of standard datasets and extensive experiments are performed to examine the effectiveness of each solution proposed for feature weighting and initialization problem. The results show that our algorithm achieves a higher performance compared to the existing competitors. The rest of the paper is organized as follows: Section 2, gives a brief review to the related work; in Section 3, the proposed algorithm, its convergence proof and its complexity analysis are provided; Section 4 presents the results of experiments; and finally, Section 5 deals with conclusion and ideas for future work.

2. Related Work

In this section, we review the existing clustering methods that have been proposed to improve the clustering performance of the basic FCM and k-means algorithms; Section 2.1 reviews the related work based on feature weighting scheme, and Section 2.2 reviews the methods against initialization sensitivity.

2.1. Global and local feature weighting

There are two groups of methods for weighting the features in the clustering approaches. As a pioneering work (in 1984), SYNCLUS algorithm used global feature weighting technique in order to determine the importance of different features in the clustering process [34]. This algorithm starts by an initial set of feature weights and uses the k-means algorithm to cluster the data into k clusters. Then, it tries to find optimal weights by optimizing a weighted mean-square cost function. These steps are iterated until the procedure converges to an optimal set of weights. Although SYNCLUS Algorithm computes the feature weights automatically, the feature group weights should be given manually. Moreover, this algorithm is very time-consuming [35], so it may not be efficiently used for large datasets.

In 2004, Wang et al. [36] also used global feature weighting scheme for FCM. To weight the features, their method utilized a learning based approach and an evolutionary fitness function. The gradient descent algorithm was employed to find proper weights. The reported results showed improvements on the original FCM. However, the evolutionary function and the similarity measure used in this method were very complicated. The complexity of this method is high. This is due to a traditional way of feature-weight learning process that depends on a high number of iterations. Also, this method is based on an assumption that the distribution of the data is uniform which is not the case in most of the real world datasets.

Huang et al. [27] in 2005 proposed W-k-means clustering algorithm. They added an additional step to the basic k-means algorithm in order to determine the weights of features at each iteration. The weight of each feature was estimated based on the sum of the within-cluster variances of the feature. As such, noise features can be identified and their effects on the clustering result can be reduced. However, their method requires users to subjectively specify an additional parameter, so-called the exponent of feature weights. Hence, it is laborious for users to determine a proper value for this parameter to obtain a high clustering quality result [37]. In addition, feature weights generated using this method might not highlight the representative features (or dim irrelevant features) well. It is also observed that the weight differences among features are somehow unobvious when the number of features is high [37].

In 2008, Hung et al. [38] presented a FCM-based algorithm for image segmentation that was carried out through global feature weighting. They estimated the feature weights using the bootstrap technique. Although their method has high computational complexity, their proposed approach has shown a good effectiveness and performance and it performed better than the aforementioned FWL [36] approach. However, the feature weights calculated by this algorithm may be unsuitable for some extreme cases [39]. In other words, in some datasets, the weight of the features does not properly represent the importance of the features [39].

In 2014, Xing et al. [40] presented two methods to improve the performance of FCM algorithm. In their basic method, they introduced a global feature weighting algorithm, namely IFWFCM. IFWFCM automatically computed feature weights in the clustering process. They then generalized the proposed method to a kernelized version by utilizing a kernelized distance measure (IFWFCM_KD). Experimental results revealed that the performance of IFWFCM is better than IFWFCM_KD on certain datasets. Recently, in contrast to the global feature weighting methods, local feature weighting methods have gained more attentions. For example, the method proposed by Frigui et al. [9] determines the weights of features for each cluster independently through an optimization procedure during the clustering process. Compared to a global weighting method proposed in [38], their method showed a higher accuracy. They applied their algorithm for image
segmentation and achieved interesting results. Frigui et al. [9] introduced two local feature weighting methods based on simultaneous clustering and attribute discrimination techniques (SCAD1 and SCAD2). In the second version of their algorithm (SCAD2), a central weighting scheme was used that can make the algorithm very sensitive to the initialization of centers. Also, in these two algorithms, in some situations, the obtained weights of the features do not properly represent the importance of features in the clusters [41].

Jing et al. [42] extended k-means algorithm to determine feature weights in each clusters, and used the weights' values to find the important features. To this end, they defined an objective function using the weight entropy for their clustering algorithm. Although their algorithm was sensitive to initial centers, in the case of proper selection of initial centers, it was able to improve the clustering performance. In 2014, Zhi et al. [33] presented a hard c-means based clustering method which automatically assigned local weights for features during the clustering process. They used new distance metric that reduced the effect of noise and outliers in the clustering process. In fact, it was an improved version of the method presented in [28]. Although the proposed method was not sensitive to noise and outliers, it was sensitive to the selection of the initial points.

In 2014, Ferreira et al. [43] carried out extensive researches on feature weighting both locally and globally. Using basic FCM and kernel distance, they introduced a clustering method which was able to improve the clustering qualities by utilizing adaptive distances. According to the experimental analysis of these algorithms, the method using local adaptive distance are superior to the method using global adaptive distance in most cases [44]. Although these methods can produce better results than traditional FCM, they are unsuitable for clustering large datasets [44].

In 2016, a maximum-entropy-regularized weighted fuzzy c-means (EWFCM) method was introduced in [22]. In this method, in order to find the optimal feature weights, the feature-weight entropy regularization technique was introduced to be included in the objective function of the clustering process. Also, the kernel based EWFCM (KEWFCM) method was proposed for clustering the data that includes non-spherical shaped clusters.

In 2016, in [45], two techniques were proposed for weighting a multivariate FCM method. In the first technique, each sample, feature and cluster has a proper weight. The aim is to determine the relevance of each feature when calculating the degree of membership for a sample regarding a cluster. The second one introduced a weighted distance which was used to consider the variability of each feature and cluster. The weights were computed at each iteration of algorithm. These adaptive distances allow the clustering algorithm to find clusters with different sizes and shapes.

In a recently proposed algorithm [46], a new objective function based on a kernel metric is proposed. They used local feature weighting scheme to find those clusters that have linearly non-separable patterns or non-hyper-spherical shapes. In their work, a multi-objective optimization method was proposed to be used in a feature weighted clustering process. Two separate objective functions, taking into account the inter cluster separation and intra cluster compactness, were optimized simultaneously.

In conclusion, algorithms that utilize the local feature weighting are highly accurate and perform better than global feature weighting algorithms. On the other hand, both types of feature weighting algorithms are highly sensitive to the selection of initial centers. If the initial centers are not appropriately selected, the algorithm's efficiency drops sharply. Moreover, some of the existing local feature weighting methods are not able to accurately weigh the features based on their real importance in some situations, so the accuracy of the clustering is reduced.

2.2. Methods against initialization sensitivity

In this section, a review of the methods that have been proposed to solve bad initialization problem are provided. These methods can be categorized into two groups. In the first group, the policy is to automatically prevent the formation of poor-quality clusters during the algorithm restarts (repeats). For example, in [47], the initial centers were determined using a stochastic function to cover the entire data space. k-means++ method [47] initializes the centers in k-means algorithm by choosing the samples that are further far from each other. The main drawback of this method from the scalability point of view is its inherent sequential nature. That is, the choice of the next center depends on the current set of centers [48]. Accordingly, k-means++ can be only applied on datasets of moderate size and only for modest values of k [48]. Methods in [49,50] randomly select initial centers, and during the clustering process, penalize the clusters with respect to the winning frequency of their representatives. Method in [49] was combined with a frequency sensitive competitive learning as a counter mechanism that penalizes the assignment of a data point to a crowded cluster. The principle is intuitively appealing, but it was approached without the emphasis on efficiency or quality. Also, complexity analysis and the corresponding empirical comparisons were not provided [51]. Method in [50] is suitable for clustering linearly non-separable patterns or non-hyper-spherical datasets, as it uses kernel distances. However, this method has difficulties in finding a suitable kernel function to map the data points from the input space to the linear kernel space.

Method in [52] introduced a technique to refine the initial point to a point probably to be close to the modes of the joint probability density of the samples. The main goal of this method was to cope with large datasets. The authors of this method argued that their algorithm leads to a refined starting seed that is not corrupted by outliers or other influential data points. However, the main problem with random methods is that they do not guarantee obtaining an optimal solution [53]. A study in [54] introduced Min-Max k-means method to deal with the sensitivity of k-means algorithm to the initial points by modifying k-means objective. This method starts using arbitrary centers and then, at sum of squared error. This method has a good performance for balanced datasets, however, it does not work well for imbalanced databases.

In the second group, the policy is to make the algorithms independent of a random initialization; hence, they do not need any restarting procedure. A global kernel k-means method was presented in [55], which utilized a kernel based clustering technique to incrementally identify nonlinearly separable clusters. This method is an incremental approach, where at each stage, one cluster is added through a global search process consisting of several repetitions of kernel k-means from proper initializations.

Another incremental and deterministic clustering algorithm was presented in [56] to deal with the bad initialization problem. In this algorithm, a kernel function is used to map data samples from the input space to a higher dimensional feature space. Then, the clustering error is optimized using the data points in new space. Both methods presented in [57] and [58] have a high computational complexity. We can conclude that although the aforementioned algorithms are not sensitive to the selection of the initial points, they suffer from three shortcomings. First, they assume that all the features have the equal importance. Second, their computational complexities are relatively high [54]. Three, as most of these algorithms use Euclidean distance, they are sensitive to noise and outliers [33].

Considering the all above described limitations of existing methods, we propose a FCM clustering method that is able to perform feature weighting and cluster weighting, simultaneously. Feature weighting is done locally and at the same time, cluster weighting is applied to cope with the problem of initialization. Carrying out these two operation in a concurrent manner during the clustering process, form high quality clusters regardless of the initial centers. Also, a non-Euclidean distance metric is used in our objective function which makes the algorithm more robust against the effects of outliers in the feature weighting process.

3. Proposed Algorithm

In this section, we introduce our clustering algorithm. First, an overview of the proposed algorithm is presented. Then, the steps of the algorithm are described in detail, and finally, convergence proof of the algorithm and the analysis of its complexity are provided.

3.1. Algorithm overview

Motivated by the problems of standard FCM clustering algorithm, we aim to design an approach that employs a local feature weighting scheme to increase the accuracy of the clustering, and a cluster weighting technique to alleviate the sensitivity of the clustering to bad initialization as well. In addition, since the Euclidean distance metric is sensitive to noise and outliers [32], we define a new objective function based on a non-Euclidean distance metric. This distance metric is introduced in [33] and is not sensitive to noise and outliers

$$F(\mathbf{U},\mathbf{C},\mathbf{W},\boldsymbol{z}) = \sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{m=1}^{M} u_{nk}{}^{\alpha} w_{km}^{q} z_k^{p} d^2\left(x_{nm}-c_{km}\right). \qquad (1)$$

In Eq. (1), N refers to the number of data samples, M is the number of features, K is the number of clusters, U = [u_{nk}] is a K by N matrix in which u_{nk} indicates the degree of nth sample's membership to the center of kth cluster, C = [c_{km}] is a K by M matrix, where c_{km} indicates the center of kth cluster and is defined by u_{nk}. Also, in this equation, W = [w_{km}] is a K by M matrix, where w_{km} refers to the weight of mth feature in kth cluster, z = [z_k] is a vector with the length of K, where z_k refers to kth cluster weight, and the term d2(x_{nm} − c_{km}) indicates a non-Euclidean distance metric and is defined as follows:

$$distance = 1 - \exp(-\gamma_m(x_{nm} - c_{km})^{2}) \qquad (2)$$

$$\gamma_m = \frac{1}{var_m} \qquad (3)$$

$$var_m = \sum_{n=1}^{N} \frac{(x_{nm}-x_m)^2}{N} \qquad (4)$$

And α is the fuzzification coefficient (α > 1) (fuzzifier). It is commonly used by fuzzy clustering methods to determine the level of fuzziness in the formed clusters. A large α results in smaller membership values, and thus, fuzzier clusters are formed. In the limit α= 1, the memberships converge to 0 or 1 implying a hard partitioning. In the absence of domain knowledge, α is commonly set to 2 [22,45].

Similar to the method presented in [54], the appropriate value of parameter p (0 ≤ p < 1) is found during the operation of the proposed clustering algorithm. This parameter controls the sensitivity of the cluster weight updating to the relative difference of the SIWDs (sum of the intra-cluster weighted-feature distance). This parameter prevents the formation of clusters with big SIWDs and thus, makes them balanced in terms of the total SIWDs. This metric indicates the quality of clustering [54]. Following to the method presented in [27], the principle that should be considered for weighting features in each cluster is as follows: assign a larger weight to a feature that has a smaller variance in the related cluster, and a smaller weight to a feature that has a larger variance. In this paper, we refer to this principle as the ''feature weighting principle''. Inspired by the research carried out in [33], in order to be able to implement this principle, we employ parameter q in Eq. (1) (in range q > 1 and q < 0). Further details on how to calculate proper values for p and q, and study their influences on the performance of the method are discussed in Section 3.2.

As stated in introduction, k-partitioning algorithms are sensitive to initialization. This sensitivity exists in both algorithms with and without feature weighting. In fact, feature weighting is necessary but not enough. After a bad initialization, it is possible that some clusters with large SIWDs are merged together, and those with low SIWD are broken into smaller clusters. Therefore, even if there are some clusters with balanced SIWDs in the dataset, after running the algorithm, it is possible that some clusters with unbalanced SIWDs are formed. This is the case that happens in k-means based algorithms. Cluster weighting solves this problem to a large extent. The principle that is used for weighting clusters is as follows: assign a larger weight to a cluster that has a larger SIWD, and a smaller weight to a cluster that has a smaller SIWD. We call this principle the ‘‘cluster weighting principle’’. By assigning a higher weight to a cluster with a big SIWD, we can prevent formation of clusters with an unbalanced SIWD. Considering the criterion drawn from SIWD, cluster weighting has the best efficiency on the datasets where there are groups (clusters) having a nearly balanced SIWD. This is due to emergence of the clusters with a balanced SIWD during the algorithm restart. However, in practice, there are not always such balanced groups (clusters) in the datasets. To solve this problem, feature weighting is implemented along with the cluster weighting. In this way, one can achieve the best efficiency of a clustering algorithm irrespective of the structure of the data and the initialization.

3.2. Algorithm description

Details of the proposed algorithm are presented in Fig. 1. According to the explanations given in Section 3.2, the performance of the proposed algorithm with the simultaneous weighting of the clusters and the features is based on the optimal adjustment of parameters p and q, that is discussed in next subsection.

3.2.1. Parameter tuning

Similar to the method presented in [54], a repeat-based algorithm is used to find the optimal value for p. Three values p_{init}, p_{step} and p_{max} are defined and the algorithm starts with a small value for p(p_{init}). In each iteration, the value of p is increased by p_{step} until the maximum value p_{max} is reached. If an empty cluster or a cluster with only one sample appears, the value of p is reduced by p_{step}, regardless of whether p is equal to p_{max} or not. In this step, we select the values of u_{nk}, w_{km} and z_{km} based on the previous p. The algorithm continues until the difference between the two successive values of the objective function is smaller than the threshold ε, or the number of iterations reaches a maximum. Considering the explanation presented in the previous section, p_{max} should not be chosen very large or close to 1. To improve the stability of the algorithm, similar to the method proposed in [54], we add a memory effect parameter to the weights. By using this parameter, smoother transitions of the values are performed between consecutive iterations. In other words, the influence of the weights from the previous iteration to the current update is controlled. Weighting the clusters and the features simultaneously may slow down the convergence speed of the algorithm. However, we will show (Section 4.3) that choosing the value of β carefully will eliminate this drawback.

3.2.2. Proposed method vs. bad initialization

In this section we aim to illustrate the performance of the proposed method against the bad initialization of cluster centers. To this end, we design a test that shows how our proposed algorithm alternates between the U, C, W and z optimization steps to get a local optimum of F. As illustrated in Fig. 2(a), we run the algorithm on a synthetic dataset, including some samples in four clusters in a two-dimensional feature space. Then, we track the path of the initial centers to the final centers of the clusters obtained by the algorithm. The obtained centers for four clusters along the first seven restarts of algorithm are depicted in Fig. 2(a) using different colors. The arrows show the path between obtained consecutive cluster centers. As it is evident in this figure, although inappropriate initial centers are deliberately selected, the algorithm can easily achieve a good local optimum. As shown in Fig. 2(b), the value obtained for the objective function is reduced in the second iteration markedly. The obtained centers in the second iteration are the reason of such behavior. The value of the objective function has been significantly reduced from second to seventh iterations, and there are, however, slight decreases from iterations 7 to 52. This shows that the proposed algorithm achieves nearly optimal solution in the seventh iteration. By performing the feature weighting along with the cluster weighting in a concurrent manner, the algorithm achieves an optimal solution very fast irrespective of the initialization.

4. Methods

In this section, the performance of the proposed algorithm is evaluated, and the results are compared with state-of-the art algorithms, namely the standard k-means (k-means) [3], the standard FCM [4], simultaneous clustering and attribute discrimination (SCAD2) [9], k-means++ [47], the hard c-means clustering with local weighting (RLFWHCM) [33], fuzzy c-means clustering with the entropy of feature weight (EWFCM) [22], and the improved version of k-means with Min–Max method (MMKMC) [54]. Threshold value ε and the maximum number of restarts are the common parameters in all the algorithms. In the experiments, we set them to 10^{-5} and t = 200, respectively. Parameter α is common between other fuzzy algorithms and our method and is set to 2 in all algorithms. Additionally, we adjust p_{step} =0.01, p_{init} = 0 and p_{max} = 0.5, and for each test dataset, we consider the number of clusters equal to the number of labeled classes.

4.1 Preparation of data

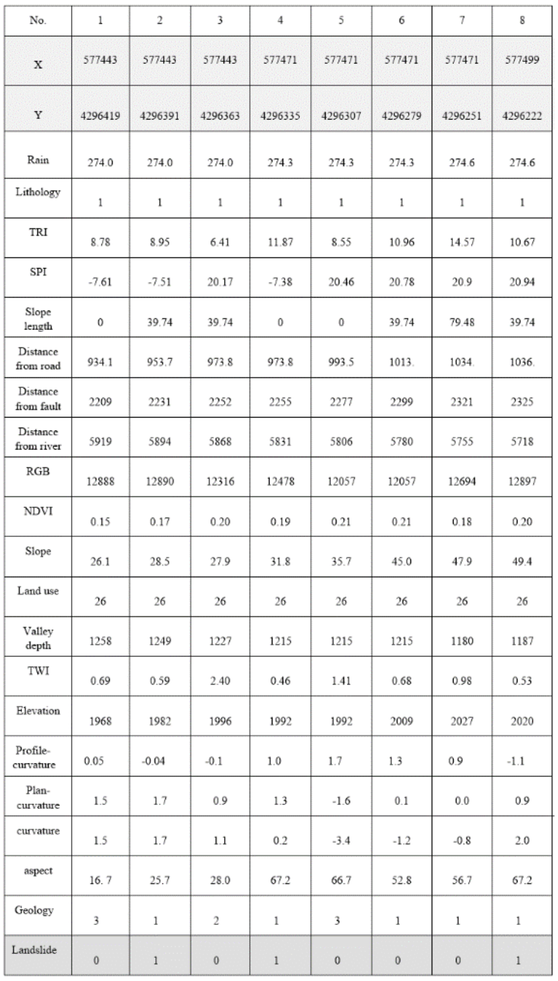
A crucial input for scrutinizing the correlation between the locations' spatial distribution and conditioning factors is the landslide inventory map (Lee et al. 2013). Hence, in the present study, a landslide inventory map was prepared via historical data on individual landslide occurrences, all-embracing field surveys corroborated by the Iran National Cartographic Center and Geological Organization also with handheld GPS devices, interpretation of aerial photographs and Google Earth images.

A sum of 766 landslides was finally mapped in the study and comprehensive reports on landslides have been exploited within a lifespan of > 37 years (since 1983). The landslides with the smallest and largest sizes found in the study were nearly 200 and 3000 (m^2) respectively. For landslide spatial modelling through the so-called amalgamation of models, the locations of the landslide were separated into two subdivisions (viz., training (70) and validating (30) appertaining to a random selection scheme. Fig. 3 indicates a series of recorded landslides in the field surveys.

Along with the landslide inventory map, numerous inter-related factors impact the landslides. A total of 10 landslide influencing factors, including slope, aspect, NDVI, elevation, distance from fault, land use, plan curvature, profile curvature, TWI, and rain were used in the proposed framework for spatial modeling [17-19].

In the present study, the ESRI ArcGIS 10.3 software was utilized with the intention of producing and displaying the data layers. All the layers of data were organized in raster format with a pixel size of 30 (m)×30 (m). The influencing factors were obtained from ASTER Global DEM, the geological map, and a topographic map with the same resolution [22]. Then the geological map was used in order to obtain the lithology map, which was then converted into a raster format. Landsat 8 (OLI images) were used in order to derive the NDVI as well as land use maps. For further analyses, all the so-called factors were standardized by a similar scale of 30 ×30 (m^2). Besides, the conditioning factors that were of continuous data were reclassified into distinct subsections with the intention of transforming continuous data to sections at specific intervals. In order to achieve the identical output scaling, the other discrete conditioning variables were reclassified into groups (Fig. 4). For training/modelling, 536 (70%) landslide locations were utilized in the present analysis, and 230 (30%) landslides were utilized for validating. A value of "1" was allocated to the landslide training instances [28]. Moreover, from the landslide-free zones, a similar amount of non-landslide points (766) was randomly produced, and a value of '0' was allocated to these instances, which were randomly divided into two sections with a ratio of 70/30 as well.

Table 1. Sample of Dataset



No.	1	2	3	4	5	6	7	8
X	577443	577443	577443	577471	577471	577471	577471	577499
Y	4296419	4296391	4296363	4296335	4296307	4296279	4296251	4296222
Rain	274.0	274.0	274.0	274.3	274.3	274.3	274.6	274.6
Lithology	1	1	1	1	1	1	1	1
TRI	8.78	8.95	6.41	11.87	8.55	10.96	14.57	10.67
SPI	-7.61	-7.51	20.17	-7.38	20.46	20.78	20.9	20.94
Slope length	0	39.74	39.74	0	0	39.74	79.48	39.74
Distance from road	934.1	953.7	973.8	973.8	993.5	1013.	1034.	1036.
Distance from fault	2209	2231	2252	2255	2277	2299	2321	2325
Distance from river	5919	5894	5868	5831	5806	5780	5755	5718
RGB	12888	12890	12316	12478	12057	12057	12694	12897
NDVI	0.15	0.17	0.20	0.19	0.21	0.21	0.18	0.20
Slope	26.1	28.5	27.9	31.8	35.7	45.0	47.9	49.4
Land use	26	26	26	26	26	26	26	26
Valley depth	1258	1249	1227	1215	1215	1215	1180	1187
TWI	0.69	0.59	2.40	0.46	1.41	0.68	0.98	0.53
Elevation	1968	1982	1996	1992	1992	2009	2027	2020
Profile-curvature	0.05	-0.04	-0.1	1.0	1.7	1.3	0.9	-1.1
Plan-curvature	1.5	1.7	0.9	1.3	-1.6	0.1	0.0	0.9
curvature	1.5	1.7	1.1	0.2	-3.4	-1.2	-0.8	2.0
aspect	16. 7	25.7	28.0	67.2	66.7	52.8	56.7	67.2
Geology	3	1	2	1	3	1	1	1
Landslide	0	1	0	1	0	0	0	1

.

4.2 Neuro-Fuzzy Model

To create a fuzzy-neural system, a type of clustering was used to limit the rules in Anfis, and then three optimization algorithms were used to optimize the parameters of the input and output membership functions of Anfis.

1.1 Creating infinite initial conditions by clustering data

Two data clustering methods were used to create the initial structure of Anfis.

It is a clustering method where each data is assigned to each cluster with different degrees of membership. This is done by minimizing the following relation:

The number of possible rules in a fuzzy system is equal to the product of the number of membership functions, so in an infinite with 10 inputs, each of which has 5 membership functions, the number of possible rules will be equal to 5^{10}. A large number of rules will greatly increase the parameters of Anfis.

$$if\ x_1\ is \begin{cases} MF_{1'1} \\ . \\ . \\ MF_{5'1} \end{cases} AND\ \ x_2\ \ is \begin{cases} MF_{1'2} \\ . \\ . \\ MF_{5'2} \end{cases} AND \dots\dots\ x_{10}$$

$$is \begin{cases} MF_{1'10} \\ . \\ . \\ MF_{5'10} \end{cases} then\ output\ is\ \begin{cases} Lmf_{i1'i2'\dots\dots i10} \\ i_k = 1 \dots 5 \end{cases}$$

$$x_1 = slope\quad x_2 = aspect \dots\dots\dots x_{10} = rain$$

This increase makes it practically impossible to optimize parameters by optimization algorithms. Fig 3. shows binary clustering. The possible states in this diagram are four areas, only areas one and four have data. Anfis considers all possible states as a rule, while two of them are redundant and cause the addition of Anfis parameters. The coordinates of the dimensions of the center of each cluster are considered as the center of the Gaussian function in the membership function.

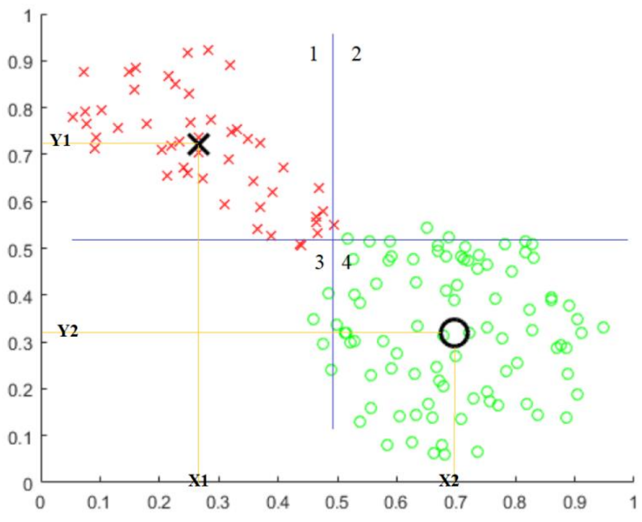


Fig 1. The effect of clustering on the elimination of redundant rules

o, by using c-means fuzzy clustering method, one rule can be considered for each cluster. Therefore, the number of rules is significantly reduced and accordingly the number of parameters is also reduced and the speed of convergence of optimization algorithms increases exponentially.

$$if\ \ x_1\ is\ MF_{j'1}\ \ AND\ \ x_2\ is\ MF_{j'2}\ \ AND\ \dots.\ \ x_{10}\ is\ MF_{j'10}$$
$$then\ output\ is\ Lmf_j\ (j = 1\ to\ 12)$$

$$Lmf_j = P_{0'j} + P_{1'j}x_1 + P_{2'j}x_2 + \dots + P_{10'j}x_{10}$$

$$MF(x) = e^{-\frac{(x-C)^2}{2\sigma^2}}$$

In this research, the number of fuzzy clustering output clusters is equal to 12. This number of clusters is obtained from the number of 10 to 15 clusters (less than the square root of the number of data) and by comparing the output results.

The two main drawbacks of FCM clustering are:

A) Being sensitive to the initial conditions that make the result of clustering not stable and sometimes divergent.

b) Sensitive to noisy and outlier data because features and clusters have the same weight. The strengths of this method are its simple structure, easy implementation, fast convergence, and the need for less storage capacity. In a research conducted in [32], a developed method for FCM was proposed, which, unlike before, solved both forms of FCM at the same time. This new method, called FCMWFWC, has the following cost function.

$$F(U;C;W;z)=\sum_{n=1}^{N}\sum_{k=1}^{K}\sum_{m=1}^{M}u_{nk}^{\alpha}\,w_{km}^{q}z_{k}^{P}d^{2}(x_{nm}-c_{km})$$
$$(0\le P\le 1\ ;\ 0\le q\le 10\ ;\alpha\ge 2)$$

The output of the FCMWFCM was used in three steps to create initial conditions in the hybrid models for the input and output membership functions of anfis.

a) The cluster centers (μ) with a range of ±5% were used to create the initial population in parameter C, using the Gaussian membership function.

b) The parameter of standard deviation of each cluster with coefficient (0.2, 0.4, 0.6) K was used for parameter σ of Gaussian membership function.

c) The parameters P_1~P_10 in each rule were randomly selected in the range of zero to one, and then the parameter P_0 was calculated according to the equation (3-3) (taking into account that each rule must give an output of one for the center of each cluster).

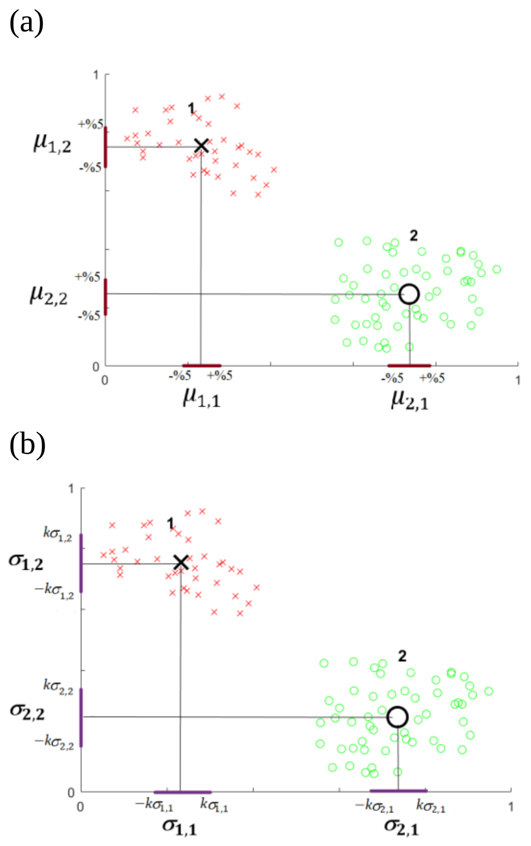


Fig 2. Using mean and std. as initial condition of anfis.

$$MF(x)=e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
$$MF_{ij}=\begin{cases}\mu_{ij}=\mu_{ij}\pm\%5\mu_{ij}\\ \sigma=k\sigma_{ij}\{k=0.2-0.4-0.6\}\end{cases}$$
$$Rule1: P_{01}+P_{11}x_1+P_{21}x_2+\cdots+P_{101}x_{10}$$
$$Rule2: P_{02}+P_{12}x_1+P_{21}x_2+\cdots+P_{102}x_{10}$$
$$\ldots$$
$$Rule12: P_{012}+P_{112}x_1+P_{212}x_2+\cdots+P_{1012}x_{10}$$
$$Rule1: P_{01}+P_{11}\mu_{11}+P_{21}\mu_{12}+\cdots+P_{101}\mu_{110}=1$$
$$Rule2: P_{02}+P_{12}\mu_{21}+P_{21}\mu_{22}+\cdots+P_{102}\mu_{210}=1$$
$$\ldots$$
$$Rule12: P_{012}+P_{112}\mu_{121}+P_{212}\mu_{122}+\cdots+P_{1012}\mu_{1210}=1$$

$$\forall\ P_{ij}(i=1\ \ to\ 10\ ;\ j=1\ \ to\ 12)\in[0;1]\ (Random)$$
$$P_{0j}(\,j=1\ \ to\ 12)=1-(P_{1j}\mu_{j1}+P_{2j}\mu_{j2}+\cdots+P_{10j}\mu_{j10})$$

The membership function of ANFIS-SFLA before and after optimization were shown in Fig 4. Fig 5. shows the Sugeno inference method for ANFIS-PSO with 12 rules and 10 inputs.

The status of membership functions of ANFIS-SFLA inputs as an example for the first rule, before and after applying the optimization algorithm is shown in Fig 6. Figure (a) shows the membership functions of the first input obtained from FCM and Figure (b) shows the same functions after applying the SFLA algorithm. Figure (c) shows the membership functions of the first input obtained from FCMWFWC and Figure (t) shows the same functions after applying the SFLA algorithm. As it is known, the latter two forms are similar, unlike the previous two forms, and this means that FCMWFWC provides close to optimal initial conditions for ANFIS-SFLA.

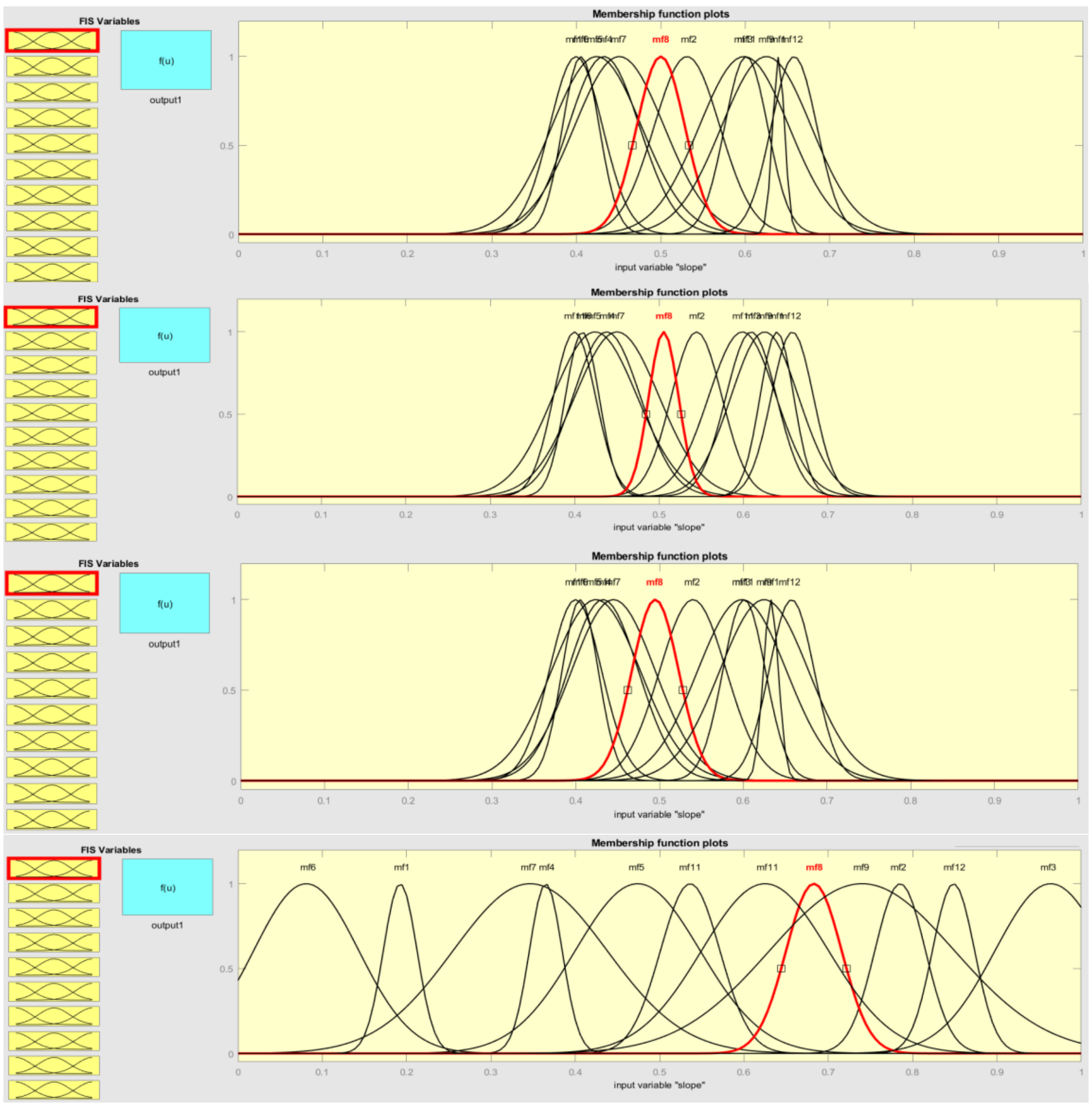


Fig 3. Initial conditions of 1’st membership function

In this section, we investigate the performance of the proposed algorithm regarding the local weighting of the features. That is, we aim to examine whether the proposed algorithm and its competitors, EWFCM [61], RLFWHCM [60] and SCAD2 [59] properly assign weights to features or not. To this end, we use the synthetic dataset. To demonstrate the importance of each feature in this dataset, we visualize the data in different 2D subspaces. Fig. 6 presents the result of this visualization. From this figure, it is understood that Cluster1 and Cluster2 are mostly formed based on Feature1 and Feature2, and Cluster3 and Cluster4 are mostly formed based on Feature2 and Feature3. This means that the first and the second features in Cluster1 and Cluster2 are more important than the third and the fourth features. Similarly, the second and the third features in Cluster3 and Cluster4 are more important than the first and the second features. We run the proposed algorithm and the other three algorithms. After the completion of the clustering process, we evaluate the weights of the features found by each algorithm. Table 2. compares the values of the weights obtained for each of the features in different clusters using different algorithms. As shown in this figure, the proposed algorithm compared to the other algorithms, assigns more weight to the first and the second features in Cluster1 and Cluster2, and to the second and the third features in Cluster3 and Cluster4. The differences in the weights found for the various features are fully consistent with the prediction made based on the data visualization in Fig. 6. This clearly shows that our algorithm does local feature weighting properly. The SCAD2 algorithm correctly performed weighting of the features in Cluster1 and Cluster3.

However, in Cluster2 and Cluster4, the weighting of the features was not done correctly. This is due to the fact that the importance of the features in those clusters was not properly recognized. So in SCAD2, the weight of the features does not properly represent the importance of each feature in each cluster.

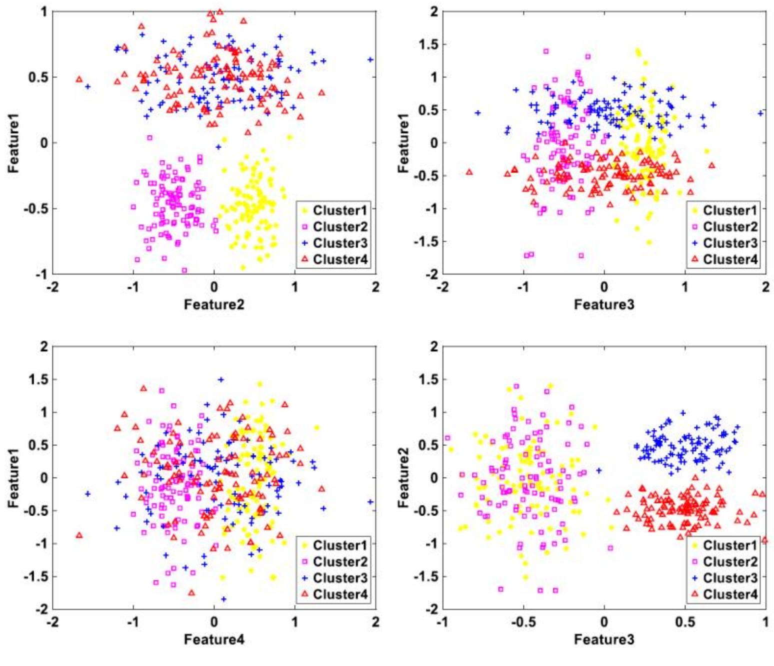
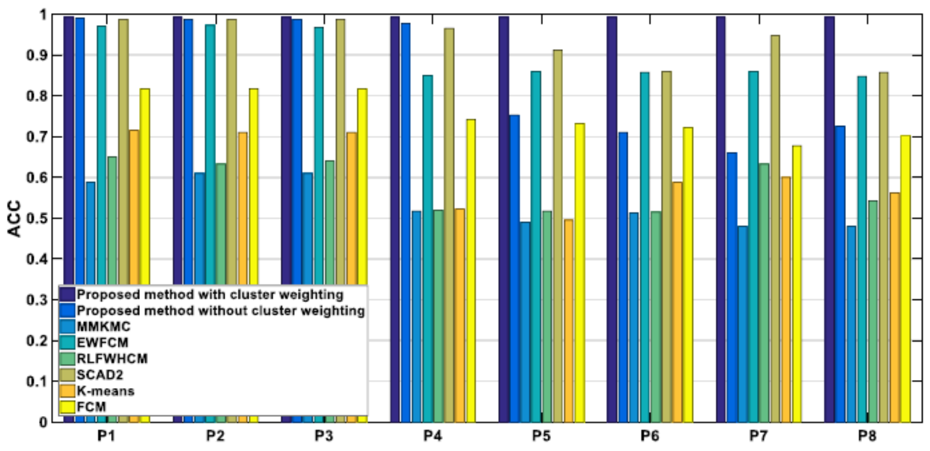
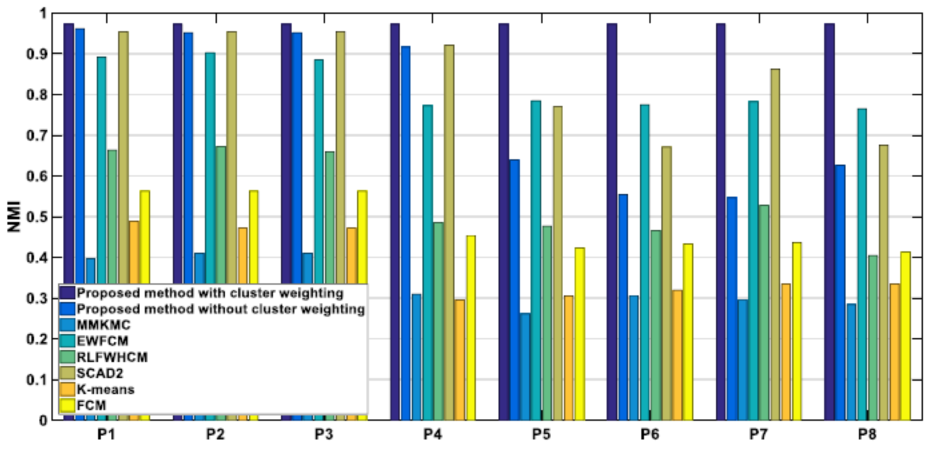


Fig 4. Visualization of synthetic dataset in various subspaces of 2D features.



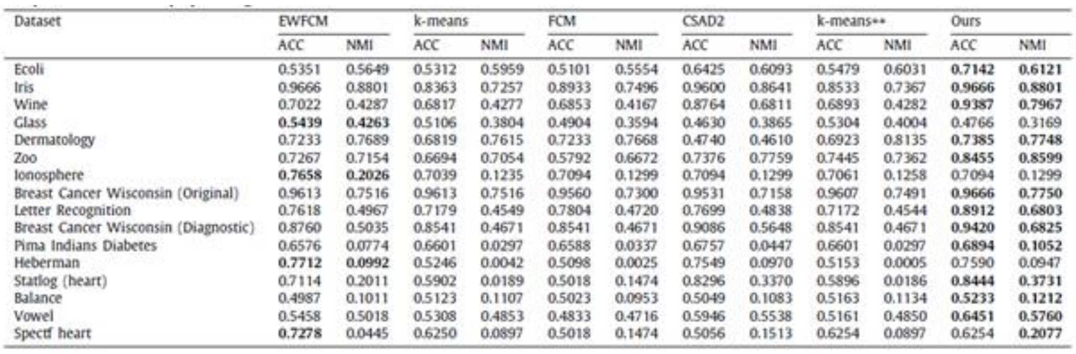
(a)



(b)

Fig 5. The performance of different algorithms using the selected initial points

Table 2. The performances of the proposed algorithm and EWFCM on the dataset.



Dataset	EWFCM		k-means		FCM		CSAD2		k-means++		Ours	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
Ecoli	0.5351	0.5649	0.5312	0.5959	0.5101	0.5554	0.6425	0.6093	0.5479	0.6031	**0.7142**	**0.6121**
Iris	0.9666	0.8801	0.8363	0.7257	0.8933	0.7496	0.9600	0.8641	0.8533	0.7367	**0.9666**	**0.8801**
Wine	0.7022	0.4287	0.6817	0.4277	0.6853	0.4167	0.8764	0.6811	0.6893	0.4282	**0.9387**	**0.7967**
Glass	**0.5439**	**0.4263**	0.5106	0.3804	0.4904	0.3594	0.4630	0.3865	0.5304	0.4004	0.4766	0.3169
Dermatology	0.7233	0.7689	0.6819	0.7615	0.7233	0.7668	0.4740	0.4610	0.6923	0.8135	**0.7385**	**0.7748**
Zoo	0.7267	0.7154	0.6694	0.7054	0.5792	0.6672	0.7376	0.7759	0.7445	0.7362	**0.8455**	**0.8599**
Ionosphere	**0.7658**	**0.2026**	0.7039	0.1235	0.7094	0.1299	0.7094	0.1299	0.7061	0.1258	0.7094	0.1299
Breast Cancer Wisconsin (Original)	0.9613	0.7516	0.9613	0.7516	0.9560	0.7300	0.9531	0.7158	0.9607	0.7491	**0.9666**	**0.7750**
Letter Recognition	0.7618	0.4967	0.7179	0.4549	0.7804	0.4720	0.7699	0.4838	0.7172	0.4544	**0.8912**	**0.6803**
Breast Cancer Wisconsin (Diagnostic)	0.8760	0.5035	0.8541	0.4671	0.8541	0.4671	0.9086	0.5648	0.8541	0.4671	**0.9420**	**0.6825**
Pima Indians Diabetes	0.6576	0.0774	0.6601	0.0297	0.6588	0.0337	0.6757	0.0447	0.6601	0.0297	**0.6894**	**0.1052**
Heberman	**0.7712**	**0.0992**	0.5246	0.0042	0.5098	0.0025	0.7549	0.0970	0.5153	0.0005	0.7590	0.0947
Statlog (heart)	0.7114	0.2011	0.5902	0.0189	0.5018	0.1474	0.8296	0.3370	0.5896	0.0186	**0.8444**	**0.3731**
Balance	0.4987	0.1011	0.5123	0.1107	0.5023	0.0953	0.5049	0.1083	0.5163	0.1134	**0.5233**	**0.1212**
Vowel	0.5458	0.5018	0.5308	0.4853	0.4833	0.4716	0.5946	0.5538	0.5161	0.4850	**0.6451**	**0.5760**
Spectf heart	**0.7278**	0.0445	0.6250	0.0897	0.5018	0.1474	0.5056	0.1513	0.6254	0.0897	0.6254	**0.2077**

Conclusion

So far, numerous researches have been conducted in designing clustering algorithms that can benefit from feature weighting. Extensive researches have also been proposed to reduce the sensitivity of the fuzzy c-means algorithm with respect to initialization. Till now, however, feature weighting in a local manner and not being sensitive with respect to initialization has not yet been applied simultaneously. In this research, a clustering algorithm was proposed. The algorithm assigns weights to the clusters based on their intra cluster distances, and also, it takes the importance of the features in each cluster into account (sum of the intra-cluster weighted feature distance). The results of the experiments on a large real world dataset as well as a synthetic dataset confirmed that the proposed algorithm has a very promising performance, and it is not sensitive to the initialization of the cluster centers, even for bad initialization, and demonstrated better clustering results than the competitors. According to the nature of the algorithm designed in the proposed method and the results of extensive experiments on various datasets, it seems that the proposed method can be efficiently used as an online feature weighting and cluster weighting-based clustering method on big data clustering and co-clustering applications. As a future direction, we are interested in improving the proposed algorithm using Kernel-based clustering approach [62] which allows the algorithm to distinguish the clusters that have linearly non-separable patterns or non-hyper-spherical shapes.

References

[1] V.N. Vapnik, V. Vapnik, Statistical Learning Theory, Vol. 1, Wiley, New York, 1998.

[2] J. Han, J. Pei, M. Kamber, Data Mining: Concepts and Techniques, Elsevier, 2011.

[3] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, 1967, pp. 281-297.

[4] J.C. Bezdek, Objective function clustering, in: Pattern Recognition with Fuzzy Objective Function Algorithms, Springer, 1981, pp. 43–93.

[5] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu, An efficient k-means clustering algorithm: Analysis and implementation, IEEE Trans. Pattern Anal. Mach. Intell. 24 (2002) 881–892.

[6] L. Zhu, F.-L. Chung, S. Wang, Generalized fuzzy c-means clustering algorithm with improved fuzzy partitions, IEEE Trans. Syst. Man Cybern. B 39 (2009) 578–591.

[7] P. Huang, D. Zhang, Locality sensitive C-means clustering algorithms, Neurocomputing 73 (2010) 2935–2943.

[8] Y. Liu, F. Tian, Z. Hu, C. DeLisi, Evaluation and integration of cancer gene classifiers: identification and ranking of plausible drivers, Sci. Rep. 5 (2015).

[9] H. Frigui, O. Nasraoui, Unsupervised learning of prototypes and attribute weights, Pattern Recognit. 37 (2004) 567–581.

[10] S. Sivarathri, A. Govardhan, Experiments on hypothesis fuzzy k-means is better than k-means for clustering, Int. J. Data Mining Knowl. Manag. Process 4 (2014) 21.

[11] A. Stetco, X.-J. Zeng, J. Keane, Fuzzy C-means++: Fuzzy C-means with effective seeding initialization, Expert Syst. Appl. 42 (2015) 7541–7548.

[12] S. Askari, N. Montazerin, M.H. Fazel Zarandi, Generalized possibilistic Fuzzy C-means with novel cluster validity indices for clustering noisy data, Appl. Soft Comput. 53 (2017) 262–283.

[13] Y. Chen, H. Zhang, Y. Zheng, B. Jeon, Q.M.J. Wu, An improved anisotropic hierarchical fuzzy c-means method based on multivariate student distribution for brain MRI segmentation, Pattern Recognit. 60 (2016) 778–792.

[14] J. Aparajeeta, P.K. Nanda, N. Das, Modified possibilistic fuzzy C-means algorithms for segmentation of magnetic resonance image, Appl. Soft Comput. 41 (2016) 104–119.

[15] Z. Ji, J. Liu, G. Cao, Q. Sun, Q. Chen, Robust spatially constrained fuzzy c-means algorithm for brain MR image segmentation, Pattern Recognit. 47 (2014) 2454–2466.

[16] O.P. Mahela, A.G. Shaik, Recognition of power quality disturbances using S-transform based ruled decision tree and fuzzy C-means clustering classifiers, Appl. Soft Comput. 59 (2017) 243–257.

[17] E. Esme, B. Karlik, Fuzzy C-means based support vector machines classifier for perfume recognition, Appl. Soft Comput. 46 (2016) 452–458.

[18] K. Siang Tan, N.A. Mat Isa, Color image segmentation using histogram thresholding – Fuzzy C-means hybrid approach, Pattern Recognit. 44 (2011) 1–15.

[19] V. T, Performance based analysis between k-means and Fuzzy C-means clustering algorithms for connection oriented telecommunication data, Appl. Soft Comput. 19 (2014) 134–146.

[20] A. Klomsae, S. Auephanwiriyakul, N. Theera-Umpon, A string grammar fuzzy-possibilistic C-medians, Appl. Soft Comput. 57 (2017) 684–695.

[21] R. Xu, D. Wunsch, Survey of clustering algorithms, IEEE Trans. Neural Netw. 16 (2005) 645–678.

[22] J. Zhou, L. Chen, C.L.P. Chen, Y. Zhang, H.-X. Li, Fuzzy clustering with the entropy of attribute weights, Neurocomputing 198 (2016) 125–134.

[23] J.M. Pe○a, J.A. Lozano, P. Larra○aga, An empirical comparison of four initialization methods for the K-means algorithm, Pattern Recognit. Lett. 20 (1999) 1027–1040.

[24] M.E. Celebi, H.A. Kingravi, P.A. Vela, A comparative study of efficient initialization methods for the k-means clustering algorithm, Expert Syst. Appl. 40 (2013) 200–210.

[25] D.S. Modha, W.S. Spangler, Feature weighting in k-means clustering, Mach. Learn. 52 (2003) 217–237.

[26] L. Parsons, E. Haque, H. Liu, Subspace clustering for high dimensional data: a review, SIGKDD Explor. Newsl. 6 (2004) 90–105.

[27] J.Z. Huang, M.K. Ng, H. Rong, Z. Li, Automated variable weighting in k-means type clustering, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 657–668.

[28] E.Y. Chan, W.K. Ching, M.K. Ng, J.Z. Huang, An optimization algorithm for clustering using weighted dissimilarity measures, Pattern Recognit. 37 (2004) 943–952.

[29] A. Likas, N. Vlassis, J.J. Verbeek, The global k-means clustering algorithm, Pattern Recognit. 36 (2003) 451–461.

[30] A.M. Bagirov, Modified global k-means algorithm for minimum sum-of squares clustering problems, Pattern Recognit. 41 (2008) 3192–3199.

[31] A.M. Bagirov, J. Ugon, D. Webb, Fast modified global k-means algorithm for incremental cluster construction, Pattern Recognit. 44 (2011) 866–876.

[32] K.-L. Wu, M.-S. Yang, Alternative c-means clustering algorithms, Pattern Recognit. 35 (2002) 2267–2278.

[33] X.-b. Zhi, J.-l. Fan, F. Zhao, Robust local feature weighting hard c-means clustering algorithm, Neurocomputing 134 (2014) 20–29.

[34] W.S. DeSarbo, J.D. Carroll, L.A. Clark, P.E. Green, Synthesized clustering: A method for amalgamating alternative clustering bases with differential weighting of variables, Psychometrika 49 (1984) 57–78.

[35] P.E. Green, J. Kim, F.J. Carmone, A preliminary study of optimal variable weighting in k-means clustering, J. Classification 7 (1990) 271–285.

[36] X. Wang, Y. Wang, L. Wang, Improving fuzzy c-means clustering based on feature-weight learning, Pattern Recognit. Lett. 25 (2004) 1123–1132.

[37] C.-Y. Tsai, C.-C. Chiu, Developing a feature weight self-adjustment mechanism for a K-means clustering algorithm, Comput. Statist. Data Anal. 52 (2008) 4658–4672.

[38] W.-L. Hung, M.-S. Yang, D.-H. Chen, Bootstrapping approach to feature weight selection in fuzzy c-means algorithms with an application in color image segmentation, Pattern Recognit. Lett. 29 (2008) 1317–1325.

[39] H. Xing, X. Wang, M. Ha, A comparative experimental study of feature weight learning approaches, in: 2011 IEEE International Conference on Systems, Man, and Cybernetics, 2011, pp. 3500-3505.

[40] H.-J. Xing, M.-H. Ha, Further improvements in feature-weighted Fuzzy C-means, Inform. Sci. 267 (2014) 1–15.

[41] H. Shen, J. Yang, S. Wang, X. Liu, Attribute weighted mercer kernel based fuzzy clustering algorithm for general non-spherical datasets, Soft Comput. 10 (2006) 1061–1073.

[42] L. Jing, M.K. Ng, J.Z. Huang, An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data, IEEE Trans. Knowl. Data Eng. 19 (2007).

[43] M.R.P. Ferreira, F. d. A. T. de Carvalho, Kernel fuzzy c-means with automatic variable weighting, Fuzzy Sets and Systems 237 (2014) 1–46.

[44] Z. Zhou, S. Zhu, Kernel-based multi objective clustering algorithm with automatic attribute weighting, Soft Comput. 22 (2018) 3685–3709.

[45] B.A. Pimentel, R.M.C.R. de Souza, Multivariate Fuzzy C-means algorithms with weighting, Neurocomputing 174 (2016) 946–965.

[46] Z. Zhou, S. Zhu, Kernel-based multiobjective clustering algorithm with automatic attribute weighting, Soft Comput. (2017).

[47] D. Arthur, S. Vassilvitskii, k-means++: The advantages of careful seeding, in: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, 2007, pp. 1027-1035.

[48] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, S. Vassilvitskii, Scalable k-means++, Proc. VLDB Endow. 5 (2012) 622–633.

[49] A. Banerjee, J. Ghosh, Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres, IEEE Trans. Neural Netw. 15 (2004) 702–719.

[50] C.-D. Wang, J.-H. Lai, J.-Y. Zhu, A conscience on-line learning approach for kernel-based clustering, in: Data Mining (ICDM), 2010 IEEE 10th International Conference on, 2010, pp. 531-540.

[51] R. Duwairi, M. Abu-Rahmeh, A novel approach for initializing the spherical K-means clustering algorithm, Simul. Model. Pract. Theory 54 (2015) 49–63.

[52] P.S. Bradley, U.M. Fayyad, Refining initial points for K-Means clustering, in: ICML, 1998, pp. 91-99.

[53] S. Ting, J. Dy, A deterministic method for initializing K-means clustering, in: 16th IEEE International Conference on Tools with Artificial Intelligence, 2004, pp. 784-786.

[54] G. Tzortzis, A. Likas, The minmax k-means clustering algorithm, Pattern Recognit. 47 (2014) 2505–2516.

[55] G. Tzortzis, A. Likas, The global kernel k-means clustering algorithm, in: Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on, 2008, pp. 1977-1984.

[56] G.F. Tzortzis, A.C. Likas, The global kernel $k $-means algorithm for clustering in feature space, IEEE Trans. Neural Netw. 20 (2009) 1181–1194.

[57] Y. Li, M. Dong, J. Hua, Localized feature selection for clustering, Pattern Recognit. Lett. 29 (2008) 10–18.

http://archive.ics.uci.edu/ml/index.php.

[58] J.C. Bezdek, A convergence theorem for the fuzzy ISODATA clustering algorithms, IEEE Trans. Pattern Anal. Mach. Intell. (1980) 1–8.

[59] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2002) 583–617.

[60] F.J. Valverde-Albacete, C. PelJez-Moreno, 100% classification accuracy considered harmful: The normalized information transfer factor explains the accuracy paradox, PLoS One 9 (2014) e84217.

[61] M. Filippone, F. Camastra, F. Masulli, S. Rovetta, A survey of kernel and spectral methods for clustering, Pattern Recognit. 41 (2008) 176–190.