A Bi-directional Fuzzy C-Means Clustering Ensemble Algorithm Considering Local Information

Chunhua Ren^{1,2} · Linfu Sun^{1,2}

Received: 8 April 2021 / Accepted: 23 August 2021
© The Author(s) 2021

Abstract

The classic Fuzzy C-means (FCM) algorithm has limited clustering performance and is prone to misclassification of border points. This study offers a bi-directional FCM clustering ensemble approach that takes local information into account (LI_BIFCM) to overcome these challenges and increase clustering quality. First, various membership matrices are created after running FCM multiple times, based on the randomization of the initial cluster centers, and a vertical ensemble is performed using the maximum membership principle. Second, after each execution of FCM, multiple local membership matrices of the sample points are created using multiple K-nearest neighbors, and a horizontal ensemble is performed. Multiple horizontal ensembles can be created using multiple FCM clustering. Finally, the final clustering results are obtained by combining the vertical and horizontal clustering ensembles. Twelve data sets were chosen for testing from both synthetic and real data sources. The LI_BIFCM clustering performance outperformed four traditional clustering algorithms and three clustering ensemble algorithms in the experiments. Furthermore, the final clustering results has a weak correlation with the bi-directional cluster ensemble parameters, indicating that the suggested technique is robust.

Keywords FCM · Local information · Multiple K-nearest neighbors · Vertical ensemble · Horizontal ensemble · Bi-directional clustering ensemble algorithm

1 Introduction

As one of the most commonly used data analysis methods in machine learning, data mining, and artificial intelligence, clustering divides data sets into clusters according to their features so that the sample points in the same cluster are highly similar, while those in different clusters are dissimilar [1]. Without knowing the class relations of any data, clustering is an unsupervised learning method, which is widely used in image processing [2], information security [3], and market analysis [4], and other fields since it can discover the potential values of data. According to various clustering theories, clustering can be classified into five categories

[5]: partition-based clustering, density-based clustering, grid-based clustering, hierarchical clustering, and model-based clustering. Among them, partition-based clustering such as K-means [6] and FCM [7, 8] are simple and efficient. Therefore, they are broadly applied in the engineering field.

First proposed by Dunn in 1974 and different from the hard clustering of K-means, FCM [7] offers a more flexible way of clustering and introduces fuzzy membership. The fuzzy membership function is introduced in fuzzy mathematics and uncertainty theory [9, 10]. The optimal memberships and cluster centers of sample points are obtained by iteratively calculating and are finally clustered following the principle of maximal membership. Although FCM has been applied successfully in many sectors since its emergence, it still has some shortcomings, e.g., sensitivity to initial cluster centers, noise points, and boundary points, poor performance in unbalanced data sets, and the tendency for local optima during iteration.

This work proposes a bi-directional FCM clustering ensemble technique that takes local information into account (LI_BIFCM) to address the drawbacks of FCM. The

✉ Chunhua Ren
418327014@qq.com

¹ School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China

² Sichuan Provincial Key Laboratory of Manufacturing Industry Chains Collaboration and Information Support Technology, Southwest Jiaotong University, Chengdu 611756, China

suggested technique considers not just clusters diversity but also sample points local information.

The following are the paper's significant innovations and contributions:

1. A vertical clustering ensemble is used to keep the algorithm stable.
2. Using multiple horizontal clustering ensembles, the technique effectively prevents border points from being misclassified.
3. LI_BIFCM increases clustering performance even further by employing horizontal and vertical ensembles.

The rest of this paper is laid out as follows: the second section examines and summarizes the associated research. The core ideas of FCM are briefly described in Sect. 3. The LI_BIFCM method is discussed in detail in Sect. 4. The experimental evaluation of the suggested algorithm is discussed in Sect. 5. Section 6 wraps up this paper and looks ahead to future research.

2 Related Works

Many FCM-derived algorithms have been presented and examined in past research work to address FCM's shortcomings.

To overcome its tendency for local optima, two improved FCM algorithms (named FCM-IDPSO and FCM2-IDPSO) were proposed by Silva [11], which dynamically adjusted the parameters based on improved particle swarm optimization and provided a better balance between exploration and exploitation. Experimental results suggested that the proposed method delivered excellent clustering results at a faster speed. To automatically identify the cluster center and initial location of FCM, an entropy-based fuzzy clustering method was presented by Yao [12] and to determine the cluster center by calculating the entropy of each sample point. A new method was also devised to estimate the initial membership functions of fuzzy sets, which has good predictive output values. Proposed by Ding et al. [13], the improved FCM algorithm combined genetic algorithm and Gaussian kernel technique, which could overcome the deficiency of FCM's failure to determine the number of clusters and boost the clustering performance. Zou et al. [14] presented an initialization method of FCM, which extracted approximate cluster centers from samples based on grid and density, and adopted the number of approximate cluster centers to initialize the number of cluster centers. Experiments indicated that this method could effectively improve the clustering performance and shorten the clustering time. To tackle the problem of FCM's sensitivity to initial cluster centers and noise points, the improved FCM algorithm based on the initial center optimization method, density clustering, and grid

clustering was proposed by Shi [15] and its effectiveness was proved by taxi trajectory data sets. Aiming to improve FCM's clustering performance in noisy data sets, the effective objective function with the cluster center learning method-based quadratic mean distance, entropy, and regularization terms was presented [16]. The cluster centers were updated according to the new objective function, and the strengths of this method were validated by varied data sets. Given that most of the improved FCM neglected the dissimilarity in clusters, Qamar et al. [17] introduced a dissimilarity measure between clusters, designed an objective function that generated highly dissimilar clusters and verified better performance of the improved FCM through data sets experiments. Li et al. [18] proposed a double fuzzy C-means clustering model, which comprised two interconnected and interactive FCM algorithms, and redesigned a new objective function to enhance the intra-cluster compactness and inter-cluster separation and to improve the clustering accuracy. Since FCM is easily affected by the Euclidean distance, Wang et al. [19] presented a weighted FCM algorithm based on the weighted Euclidean distance, which included feature weights into the weighted Euclidean distance. Experiments suggested that the clustering performance could be enhanced by the improved FCM. Haldar et al. [20] devised an improved FCM algorithm based on the Mahalanobis distance, which improved the clustering quality compared with the traditional FCM algorithm. Considering the varying weights of sample points, Wu et al. [21] proposed an improved FCM algorithm combining adaptive weights with SA-PSO, which could avoid local optima and effectively improve the clustering performance. To solve the fuzzy boundary value and fuzzy integrodifferential equations problem, the reproducing kernel algorithm is adopted by Arqub [22, 23]. In response to different sample weights and feature weights, Wu [24] further proposed an improved FCM algorithm, which introduced the adaptive data weight vector, the adaptive feature weights matrix, and constructed a new objective function. Experimental results demonstrated that the clustering performance of the algorithm was improved significantly. Although the above-related studies of FCM have made some progress, the performance of an individual or improved FCM clustering algorithm is limited for different data sets, and poor stability.

With the deepening of the research and the differences in the actual application of data sets, the individual FCM clustering algorithm has certain limitations. Hence, the idea of integrating different clustering algorithms has emerged to upgrade the clustering results and meet the actual application requirements [25]. As an unsupervised learning technique based on ensemble, clustering ensemble combines multiple individual clustering results into a unified robust result and can achieve higher accuracy than the individual clustering method [26]. To overcome the failure of traditional FCM to cluster large data sets effectively, Li et al. [27] proposed an FCM-based ensemble clustering algorithm for large data sets, which could

improve the clustering accuracy by clustering atoms on data sets. To improve the robustness of clustering, Su et al. [28] proposed a link-based pairwise matrix method for the clustering ensemble of FCM, which adopted a fuzzy graph to represent the relationship between component clusters and then obtained the final ensemble clustering results. Experimental results illustrated that the proposed method outperformed other methods. Su et al. [29] also presented a hierarchical fuzzy clustering ensemble approach, which employed FCM and hierarchical clustering method to generate base clusters and achieve consensus functions, and verified its advantages in clustering accuracy and time efficiency on large data sets. The fuzzy C-means clustering algorithm with improved random projection was proposed [30], which improved the efficiency of clustering through singular value decomposition of the concatenation of membership matrices. To unify the fuzzy clustering partitions, Wan et al. [31] presented an FCM-based fuzzy consensus clustering framework (FCC), which redesigned the objective function and translated FCC into a weighted and segmented FCM clustering, and verified its effectiveness theoretically and experimentally. At present, much attention is paid to the generation methods of cluster membership and design of consensus function in the field of clustering ensembles, whereas little consideration is given to local information of sample points in the process of the ensemble.

In conclusion, the quality of a single FCM cluster is limited, and effective border point allocation is impossible. As a result, more research is required in response to these flaws. In contrast, our research will focus on the membership category of cluster border points and increase the clustering effect by focusing on the idea of clustering ensemble. The original FCM and the proposed algorithm are described in depth in the next section.

3 Fuzzy C-Means Clustering Algorithm

As a classical partition clustering algorithm, the FCM algorithm is mainly worked by obtaining the fuzzy membership of each sample point to all cluster centers through optimizing the objective function, to determine the category of sample points. Given the data set $X = \{x_1, \dots, x_i, \dots, x_n\}$, it is divided into c clusters, and the center of each cluster is $c_j (j = 1, 2, \dots, c)$.

The objective function of FCM is defined as follows:

$$J_e = \sum_{i=1}^n \sum_{j=1}^c u_{ij}^e \|x_i - c_j\|^2, \quad (1)$$

$$\text{s.t. } \sum_{j=1}^c u_{ij} = 1,$$

where u_{ij} is the fuzzy membership of x_i in the c_j cluster, e is the membership factor, and $\|x_i - c_j\|$ represents the Euclidean distance from x_i to c_j .

In the process of fuzzy clustering, the membership u_{ij} and the cluster center c_j are iterated constantly until clustering is completed, that is when the membership does not change greatly or the number of iterations t satisfies the following equation, or when the objective function J_e reaches the local optimum (minimum):

$$|u_{ij}^{(t+1)} - u_{ij}^{(t)}| < \varepsilon. \quad (2)$$

where ε is an error threshold.

To get the minimum J_e , the updating calculation method of u_{ij} and c_j is shown in the following equations:

$$c_j = \frac{\sum_{i=1}^n u_{ij}^e * x_i}{\sum_{i=1}^n u_{ij}^e}. \quad (3)$$

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{e-1}}}. \quad (4)$$

FCM determines the final clustering result C_{xi}^{result} according to the principle of maximum membership, as shown in the following equation:

$$C_{xi}^{\text{result}} = \max(u_{ij}). \quad (5)$$

The clustering algorithm of FCM is as follows:

Algorithm 1 The FCM algorithm

Input: A dataset X , clusters c , fuzzy factor e (usually take 2), stop iterating the condition ε , and maximum number of iterations T ;

Output: The clustering result C ;

- 1: Random initial clustering center c_j and initialize the membership matrix $u_{ij} = 0$;
 - 2: In the iteration process, the membership matrix u_{ij} and the clustering center c_j are updated respectively according to Equations (4) and (3);
 - 3: If Equation (2) or $t = T$ is satisfied, the iteration is stopped. Otherwise, set $t = t + 1$ and go back to step 2;
 - 4: According to Equation (5), the classification of each sample point is determined;
 - 5: Return C .
-

4 The LI_BIFCM Algorithm

This section explains the LI_BIFCM algorithm's framework and the essential steps of the proposed technique in depth.

4.1 The LI_BIFCM Algorithm Framework

Figure 1 depicts the LI_BIFCM structure, which is separated into three sections: vertical ensemble, horizontal ensemble, and final clustering ensemble. First, the parameters introduced into the system are the data set, cluster center, ensemble times m , multiple KNN parameters p (p is a percentage, normally takes 2%), and s (s is used to control the number of executions of multiple KNN). Then, the data set is executed for m times by FCM, yielding distinct membership matrices due to the randomness of the FCM cluster center, resulting in the vertical ensemble. Following that, after each cluster, multiple K-nearest neighbors are utilized to build the local membership matrix of sample points to form a horizontal ensemble, and the m times cluster can produce m horizontal ensemble clustering results. Finally, the vertical and horizontal ensembles are combined to generate the final clustering results.

4.2 Vertical Ensemble

A single FCM is used as the base cluster member in LI_BIFCM. Clustering results are different because FCM randomly generates the initial cluster center each time, and this feature is used to generate diverse and stable clustering results.

Suppose the data set has n sample points $X = \{x_1, \dots, x_i, \dots, x_n\}$ and c clusters $\{c_1, \dots, c_j, \dots, c_c\}$. FCM performs m times to get m membership matrices $U = [U_1, \dots, U_f, \dots, U_m]$. Each membership matrix is shown in the following equation:

$$U_f = \begin{bmatrix} u_{x_1 c_1}^f & u_{x_1 c_2}^f & \dots & u_{x_1 c_c}^f \\ u_{x_2 c_1}^f & u_{x_2 c_2}^f & \dots & u_{x_2 c_c}^f \\ \dots & \dots & \dots & \dots \\ u_{x_n c_1}^f & u_{x_n c_2}^f & \dots & u_{x_n c_c}^f \end{bmatrix}. \quad (6)$$

After each cluster is completed, the corresponding maximum membership matrix is calculated, as shown in the following equation:

$$L^f = [l_{x_1}^f, l_{x_2}^f, \dots, l_{x_n}^f], \quad (7)$$

$$l_{x_i}^f = \arg \max_{1 \leq j \leq c} (u_{x_i c_j}^f),$$

where $l_{x_i}^f$ represents the class number of the maximum membership of the sample point x_i .

FCM can obtain the maximum membership class number $l_{x_i}^f$ of each sample point after performing m times of clustering, and the clustering results after vertical ensemble are shown in the following equation:

$$C_{x_i}^{\text{Vertical}} = \arg \max_{1 \leq j \leq c} (l_{x_i}^f \in c_j). \quad (8)$$

4.3 Horizontal Ensemble

The multiple K-nearest neighbors approach is used to consider the local information of sample points to solve the problem of boundary point misclassification.

First, the K-nearest neighbors of each sample point is calculated, and its calculation formula is shown in the following equation:

$$\text{KNN}(x_i) = \{x_j \in X \mid d(x_i, x_j) \leq d(x_i, \text{NN}_k(x_i))\}, \quad (9)$$

Fig. 1 LI_BIFCM clustering ensemble framework

where $d(x_i, x_j)$ denotes the Euclidean distance between the point x_i and x_j , $NN_k(x_i)$ represents the k -th nearest neighbor of x_i , and $k \in [1, n]$.

The membership matrix is obtained after one FCM clustering, and the maximum membership matrix $C_{x_i}^{1,k}$ of K -nearest neighbors of the sample points is calculated, as shown in the following equation:

$$C_{x_i}^{1,k} = \arg \max_{1 \leq j \leq c} (l_{km(x_i)}^1 \in c_j). \quad (10)$$

Then, the maximum membership matrix $C_{x_i}^{1,k+s}$ of multiple K -nearest neighbors (k takes different values) of the sample points are calculated, as shown in the following equation:

$$C_{x_i}^{1,k+s} = \arg \max_{1 \leq j \leq c} (l_{(k+s)nm(x_i)}^1 \in c_j), \quad (11)$$

where s is used to control the number of executions of multiple K -nearest neighbors, and its range is 1 to $n - 1$.

A horizontal ensemble is carried out by combining the clustering result of one FCM and the membership matrix of multiple K -nearest neighbors to obtain its clustering result $C_{x_i}^{1,Horizontal}$, as shown in the following equation:

$$C_{x_i}^{1,Horizontal} = \arg \max \left(\left(l_{x_i}^1 \cup \left(\bigcup_{p=k}^{k+s} C_{x_i}^{1,p} \right) \right) \in c_j \right). \quad (12)$$

Similarly, m horizontal clustering ensemble results $C_f^{Horizontal}$ can be obtained after performing m times of clustering.

Last, the final clustering results $C_{x_i}^{result}$ are obtained by integrating the clustering results of vertical ensemble and m horizontal ensembles, as shown in the following equation:

$$C_{x_i}^{result} = \arg \max \left(\left(C_{x_i}^{Vertical} \cup \left(\bigcup_{p=1}^m C_{x_i}^{p,Horizontal} \right) \right) \in c_j \right). \quad (13)$$

4.4 The LI_BIFCM Algorithm Flow

The following is the LI_BIFCM algorithm flow:

1. Set the number of cluster centers as c , the fuzzy factor as e , the condition of stopping iteration as ϵ , the maximum number of iteration as T , the number of cluster ensemble parameters m and s .
2. Random initial cluster center: $c_1, c_2, \dots, c_c$ and membership matrix: U_0 .
3. Execute the FCM algorithm m times to get membership matrix $U = [U_1, U_2, \dots, U_m]$ as shown in Eq. (6).

4. Calculate the maximum membership matrix L^f of each clustering by Eq. (7), and carry out a vertical ensemble by Eq. (8) to obtain the vertical ensemble clustering results $C_{x_i}^{Vertical}$.
5. Calculate $KNN(x_i)$ of each sample point x_i according to Eq. (9).
6. Calculate the maximum membership matrix $C_{x_i}^{1,k}, C_{x_i}^{1,k+1}, \dots, C_{x_i}^{1,k+s}$ of multiple K -nearest neighbors after each run of FCM according to Eqs. (10) and (11).
7. In combination with the membership matrix of FCM and the maximum membership matrix of multiple K -nearest neighbors, the result $C_{x_i}^{1,Horizontal}$ of a horizontal clustering ensemble is obtained according to Eq. (12).
8. Similarly, m times of horizontal ensemble clustering results $C_f^{Horizontal}$ can be obtained if FCM performs m times.
9. According to Eq. (13), the final clustering ensemble result $C_{x_i}^{result}$ can be obtained by combining the clustering result of the vertical ensemble with that of m times horizontal ensemble.

LI_BIFCM's pseudo code is given as follows based on the above analysis.

Algorithm 2 The LI_BIFCM algorithm

Input: A data set X , clusters c , the number of ensemble m , KNN parameters p and s ;

Output: The clustering result C ;

- 1: $X = \text{normalize}(X)$;
- 2: **for** $f = 1 : m$ **do**
- 3: execute FCM on X ;
- 4: calculate $l_{x_i}^f$ use (6) and (7);
- 5: **for** $k = n * p : (n * p + s)$ **do**
- 6: calculate $KNN(x_i)$ use (9);
- 7: **for** $k = n * p : (n * p + s)$ **do**
- 8: calculate $C_{x_i}^{1,k}$ use (10);
- 9: **end for**
- 10: calculate $C_{x_i}^{1,k+s}$ use (11);
- 11: **end for**
- 12: calculate $C_{x_i}^{1,Horizontal}$ use (12);
- 13: **end for**
- 14: calculate $C_{x_i}^{Vertical}$ use (8);
- 15: calculate $C_{x_i}^{Result}$ use (13);
- 16: return the clustering result C .

4.5 Time Complexity Analysis

The time complexity of LI_BIFCM is given when the preceding description is combined. Let n be the number of test sample sets, c be the number of cluster centers, and t be the number of FCM iterations. Our algorithm's time complexity is determined by the following two factors: (a) the time it takes to run FCM numerous times determines the vertical clustering ensemble. Executing FCM takes $O(nct)$ time while performing vertical ensemble takes $O(mnct)$, where m is the number of layers in the vertical ensemble. (b) Horizontal clustering ensemble uses multiple K-nearest neighbors to calculate the time of local information. The time complexity of searching for K-nearest neighbors of a point is $O(n)$. As a result, the time complexity of K-nearest neighbors executed s times is $O(sn)$. In summary, LI_BIFCM has a time complexity of $O(mnct + sn)$.

5 Experimental Results and Analyses

5.1 Experimental Settings

To evaluate the performance of the LI_BIFCM algorithm, twelve synthetic and real-world data sets are used in the experiment, which are from different fields and have different sizes, dimensions, and category numbers, as shown in Tables 1 and 2. These data sets are widely used in clustering tests, through which the clustering performance of LI_BIFCM in different application scenarios can be simulated. The experimental environment is a PC with an Intel (R) Core(TM) i7-7500 CPU @ 2.70GHz, 2.90 GHz, 12G RAM, Windows 10 64-bit OS, and the programming tool is Matlab 2015b.

Three classical clustering evaluation indexes are used in the experiment. The accuracy of clustering (ACC) describes the comparison results between the clustering labels and the real labels of the sample points [32]. The adjusted rand index (ARI) represents the overlap degree between clustering partition and actual partition [33]. The adjusted mutual information index (AMI) indicates the consistency between the clustering results and the real categories [34]. The specific calculating formulas of the three evaluation indexes are as follows:

$$\text{ACC} = \frac{1}{n} \sum_{i=1}^n \delta(u_i, v_i), \quad (14)$$

$$\delta(u_i, v_i) = \begin{cases} 1, & \text{if } (u_i = v_i) \\ 0, & \text{otherwise} \end{cases}$$

Table 1 Synthetic data sets

Data set	#Size	#Attribute	#Cluster
S1	5000	2	15
S2	5000	2	15
S3	5000	2	15

Table 2 Real-world data sets

Data set	#Size	#Attribute	#Cluster
WDBC	569	30	2
Wine	178	13	3
Segmentation	2310	19	7
Seeds	210	7	3
Dermatology	366	34	6
Parkinsons	195	23	2
Pima	768	8	2
Iris	150	4	3
Ionosphere	351	34	2

where n represents the total number of samples, u_i, v_i are the clustering labels and the real labels, respectively, and $\delta(u_i, v_i)$ is a delta function.

$$\text{ARI} = \frac{\text{RI} - E[\text{RI}]}{\max(\text{RI}) - E[\text{RI}]}, \quad (15)$$

$$\text{RI} = \frac{a + b}{C_2^n},$$

where a means the sample logarithm of the clustering result and the real category in the same category, b means that the clustering result and the real category are the sample logarithm of different categories, and $E[\text{RI}]$ means the expectation of rand index RI.

$$\text{AMI} = \frac{\text{MI}(U, V) - E\{\text{MI}(U, V)\}}{\sqrt{H(U)H(V)} - E\{\text{MI}(U, V)\}},$$

$$\text{MI}(U, V) = \sum_{i=1}^R \sum_{j=1}^C P(i, j) \log \frac{P(i, j)}{P(i)P(j)}, \quad (16)$$

$$P(i, j) = \frac{|U_i \cap V_j|}{n}, P(i) = \frac{|U_i|}{n}, P(j) = \frac{|V_j|}{n},$$

$$H(U) = - \sum_{i=1}^R P(i) \log(P(i)),$$

where U, V represent the real labels vector and clustering labels vector, respectively, R, C represent the number of real clusters and clustering clusters, respectively, and $P(\cdot)$ stands for probability, and $H(\cdot)$ stands for information entropy.

To eliminate the difference between data dimensions, the data sets need to be standardized before the experiment and the calculation method is shown as the following equation:

$$x_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)}, \quad (17)$$

where x_{ij} represents the attribute value of the sample points x_i in j th column, and $\max(x_j)$ and $\min(x_j)$ are the maximum and minimum values of the j th attribute column, respectively.

5.2 The Comparison with the Original FCM Algorithm on Synthetic Data Sets

To test the effectiveness of LI_BIFCM in processing boundary points, three two-dimensional synthetic data sets are chosen for experimental comparison with FCM. The size of these three data sets is 5000, with 15 clusters. The degree of contact between border points varies, and misclassification is common. Figures 2, 3 and 4 depict the clustering findings. Figure 2 shows that LI_BIFCM can handle the boundary points of the S1 data set successfully, whereas FCM has two issues: one is an error in the cluster center, and the other is an error in the boundary points. This is because when a cluster center error occurs in FCM, it sets off a chain reaction. Black boxes 1 and 2 are independent clusters, as illustrated in Figs. 2c, but because black box 1 has two cluster centers, the clusters in black box 2 can only be assigned to the adjacent cluster. Figure 3 depicts the clustering result of S2. The problem of boundary point misdivision and cluster center error exists in FCM as well, but not in our technique. Figure 4 depicts the experimental results of S3. Similarly, FCM splits a cluster into many clusters by mistake, but LI_BIFCM maintains excellent efficiency.

Figure 5 summarizes the clustering index results from Figs. 2, 3 and 4, showing that LI_BIFCM beats FCM in all three indicators on the synthetic data sets. FCM is easily influenced by the initial clustering center, falls into a local optimum, and is sensitive to border points for this reason. By using a vertical ensemble, our approach can discover the best clustering center. The proposed algorithm uses vertical ensemble to determine the best clustering center, and horizontal ensemble to efficiently assign boundary points, which increases clustering quality.

5.3 The Comparison with the Basic Algorithms and the Clustering Ensemble Algorithms on Real-World Data Sets

To test the LI_BIFCM algorithm's clustering performance further, seven representative algorithms are selected for comparison, which includes basic algorithms (K-means, DBSCAN, AP, and FCM), soft clustering ensemble algorithm (CSPA [35, 36]), and hard clustering ensemble algorithms (HGPA [35, 36], and MCLA [37]).

K-means is a clustering technique based on partitions. It must determine the number of clusters to be created, choose initial cluster centers at random, and calculate the distance between each sample and each cluster center. The cluster centers are recalculated once each sample is assigned to the cluster that is closest to it. This step is repeated until the algorithm reaches a particular termination condition.

DBSCAN is a clustering algorithm that is based on density. The number of clusters does not need to be specified in advance, but it does have two parameters: domain radius (Eps) and core point threshold ($MinPts$). The clustering process starts from a selected core point and continuously expands to the area where the density is reachable to obtain a maximum area containing the core point and the boundary point, and any two points in the area are connected by density.

AP is a clustering algorithm that works by transferring information between sample points. It uses all samples as network nodes, iteratively calculating the information

Fig. 2 Clustering results of LI_BIFCM and FCM on S1 data set

Fig. 3 Clustering results of LI_BIFCM and FCM on S2 data set

Fig. 4 Clustering results of LI_BIFCM and FCM on S3 data set

Fig. 5 Cluster evaluation index of LI_BIFCM and FCM on synthetic data sets

(responsibility and availability) of each network edge until several high-quality exemplars are obtained and the remaining points are assigned to the appropriate clusters.

CSPA is a cluster-based similarity partition algorithm that starts by defining a new similarity between any two sample points, then calculates a $n \times n$ similarity matrix, and finally clusters the data using a pairwise similarity-based clustering algorithm.

HGPA is a kind of hypergraph partition algorithm. It starts by creating a hypergraph with sample points at the vertices. Each cluster in each cluster member is a closed hyperedge that includes all of the cluster's vertices, and the clustering results are obtained using the hypergraph partition procedure.

MCLA is a meta-clustering algorithm. It measures the similarity of two clusters using the Jaccard coefficient, then obtains meta-clusters using the METIS method, and lastly allocates samples to the most relevant meta-clusters.

As a result, using these algorithms to conduct comparative trials is extremely sensible. Each basic algorithm experiment is conducted ten times to get the average value of the clustering findings, and each clustering ensemble algorithm is run until 10 ideal cluster member sets are produced. The clustering accuracy of eight algorithms on nine test data sets is shown in Table 3, in which the best clustering evaluation index among eight clustering algorithms is indicated in bold. Table 3 reveals the average ACC of LI_BIFCM ranks first among eight algorithms, achieves the highest clustering accuracy in six of the nine data sets and ranking top two in the other data sets (Dermatology and Iris). Only in the Ionosphere data set, LI_BIFCM ranks 4th.

To see if there are statistically significant differences between the LI_BIFCM method and the other seven algorithms in Table 3. With a confidence level of 0.95, the Aligned Friedman test [38] is used. The specific method is the clustering accuracy of a certain algorithm on a certain data set minus the average clustering accuracy of all algorithms on this data set to participate in the sorting. The clustering accuracy of eight algorithms on nine data sets ranging from 1 to 72 is shown in Table 4, and the numbers in bold indicate the best ACC sorting. Table 4 shows that LI_BIFCM achieved the best-ranking result with an average of 16.8, MCLA ranked second with an average ranking value of 24.1, FCM ranked third with an average ranking value of 28.6, and the ranking of the remaining algorithms are as follows: AP, K-means, CSPA, DBSCAN, and HGPA.

The Friedman aligned rank test [38] is then applied, with the calculation method depicted as the following equation:

$$\begin{aligned} \sum_{j=1}^k \widehat{R}_{\cdot j}^2 &= 151^2 + 342^2 + 468^2 + 289^2 + 257.5^2 + 364.5^2 \\ &\quad + 539.5^2 + 216.5^2 = 979409, \\ \sum_{i=1}^k \widehat{R}_{i \cdot}^2 &= 218^2 + 263^2 + 273^2 + 261.5^2 + 275^2 \\ &\quad + 358^2 + 343^2 + 279^2 + 357^2 = 786689.5, \\ T &= \frac{(8-1) \times (979409 - (8 \times 9^2/4) \times (8 \times 9 + 1)^2)}{8 \times 9 \times (8 \times 9 + 1) \times (2 \times 8 \times 9 + 1)/6 - 786689.5/8} \\ &= 28.3357. \end{aligned} \quad (18)$$

In the eight algorithms and nine test data sets, the statistic T obeys the Chi-square distribution with seven degrees of freedom. We can observe that the p value of $\chi^2(7)$ is $1.91\text{E}-04$, which is substantially less than 0.05, by searching up the Chi-square distribution table. Therefore, the null hypothesis can be rejected. The results obtained by all algorithms on the nine data sets are considered to be significantly different.

Similarly, Tables 5 and 6 show that the average ARI and AMI of LI_BIFCM rank first among eight algorithms, and achieve the highest ARI and AMI in at least seven data sets, indicated by bold numbers. Then, the Friedman Aligned Rank test [38] is used in conjunction with Table 7 to compare the clustering index ARI, and calculated $T = 25.4161$. The statistical variable T obeys the chi-square distribution with seven degrees of freedom, the p value of $\chi^2(7)$ is $6.40\text{E}-04$, which is much less than 0.05. Therefore, the null hypothesis can be rejected, and the ARI value of the LI_BIFCM algorithm is statistically significantly better than the other seven algorithms in the ranking results. On the other clustering index AMI, the ranking results of AMI are shown in Table 8, the p value of $\chi^2(7)$ is $4.25\text{E}-05$, and its value is also much smaller than 0.05. The bold numbers in Tables 7 and 8 represent the best ARI and AMI rankings of the algorithm, respectively. Therefore, it is concluded that the proposed algorithm is superior to other seven algorithms.

Table 3 ACC of 8 algorithms on 9 data sets

Data set	LI_BIFCM	K-means	DBSCAN	AP	FCM	CSPA	HGPA	MCLA
WDBC	0.9385	0.9279	0.8471	0.9240	0.9279	0.8102	0.5149	0.9279
Wine	0.9607	0.9494	0.8146	0.8540	0.9494	0.9157	0.6517	0.9494
Segmentation	0.7333	0.6000	0.4143	0.6700	0.6819	0.7095	0.7143	0.7143
Seeds	0.9238	0.8905	0.6905	0.8950	0.9000	0.8952	0.6667	0.9000
Dermatology	0.8073	0.4553	0.5894	0.8140	0.7414	0.6816	0.6480	0.7626
Parkinsons	0.6923	0.6308	0.5949	0.6720	0.6718	0.6513	0.5744	0.6718
Pima	0.6771	0.6680	0.6549	0.6240	0.6667	0.6250	0.5026	0.6667
Iris	0.9000	0.8867	0.7400	0.9070	0.8933	0.8800	0.6467	0.8933
Ionosphere	0.7094	0.7123	0.9145	0.7090	0.7009	0.6781	0.5840	0.7123
Average	0.8158	0.7468	0.6956	0.7854	0.7926	0.7607	0.6115	0.7998

In summary, LI_BIFCM achieved the best clustering results on the most real-world data sets for the three reasons listed below. First, numerous membership matrices can correctly locate cluster centers and ensure rather stable clustering results in the vertical ensemble. Second, the method of multiple K-nearest neighbors is employed in the horizontal ensemble process to fully use the local information for the sample points to find the optimal category for border points. Third, the attribution of sample points is clarified after the bi-directional

clustering ensembles, and clustering performance is effectively improved.

5.4 Parameter Analysis

5.4.1 The Number of Vertical Ensembles Parameters m

Each additional vertical ensemble will increase the time cost using the algorithm of this paper. Experiments are

Table 4 ACC sorting results of 8 algorithms on 9 data sets

Data set	LI_BIFCM	K-means	DBSCAN	AP	FCM	CSPA	HGPA	MCLA	Total
WDBC	0.086(4)	0.076(9)	− 0.005(46)	0.072(12)	0.076(9)	− 0.042(57)	− 0.337(72)	0.076(9)	218
Wine	0.08(5)	0.069(14)	− 0.066(60)	− 0.027(54)	0.069(14)	0.035(33)	− 0.229(69)	0.069(14)	263
Segmentation	0.079(6.5)	− 0.055(59)	− 0.24(71)	0.015(42)	0.027(38.5)	0.055(21)	0.06(17.5)	0.06(17.5)	273
Seeds	0.079(6.5)	0.045(29)	− 0.155(66)	0.05(25.5)	0.055(21)	0.05(25.5)	− 0.179(67)	0.055(21)	261.5
Dermatology	0.12(3)	− 0.232(70)	− 0.098(62)	0.127(2)	0.054(23)	− 0.006(48)	− 0.039(56)	0.075(11)	275
Parkinsons	0.047(28)	− 0.014(52.5)	− 0.05(58)	0.027(38.5)	0.027(38.5)	0.006(43)	− 0.071(61)	0.027(38.5)	358
Pima	0.041(31)	0.032(34)	0.019(41)	− 0.012(51)	0.031(35.5)	− 0.011(50)	− 0.133(65)	0.031(35.5)	343
Iris	0.057(19)	0.043(30)	− 0.103(63)	0.064(16)	0.05(25.5)	0.037(32)	− 0.197(68)	0.05(25.5)	279
Ionosphere	− 0.006(48)	− 0.003(44.5)	0.199(1)	− 0.006(48)	− 0.014(52.5)	− 0.037(55)	− 0.131(64)	− 0.003(44.5)	357.5
Total	151.0	342.0	468.0	289.0	257.5	364.5	539.5	216.5	2628
Average rank	16.8	38.0	52.0	32.1	28.6	40.5	59.9	24.1	292

Table 5 ARI of 8 algorithms on 9 data sets

Data set	LI_BIFCM	K-means	DBSCAN	AP	FCM	CSPA	HGPA	MCLA
WDBC	0.7672	0.7302	0.4786	0.7180	0.7305	0.3839	− 0.0008	0.7305
Wine	0.8837	0.8471	0.5292	0.6160	0.8498	0.7597	0.4425	0.8498
Segmentation	0.5424	0.5106	0.2129	0.5020	0.5056	0.5194	0.5239	0.5239
Seeds	0.7880	0.7049	0.5291	0.7150	0.7266	0.7133	0.4179	0.7266
Dermatology	0.7447	0.3542	0.4106	0.7170	0.6537	0.5306	0.5213	0.6867
Parkinsons	0.1421	0.0520	0.0252	0.1270	0.1109	0.0880	0.1695	0.1109
Pima	0.1186	0.1024	0.0118	0.0890	0.1069	0.0614	− 0.0012	0.1069
Iris	0.7445	0.7163	0.6120	0.7570	0.7287	0.7009	0.4096	0.7287
Ionosphere	0.1728	0.1776	0.6835	0.1730	0.1589	0.1245	0.0257	0.1776
Average	0.5449	0.4661	0.3881	0.4904	0.5080	0.4313	0.2787	0.5157

Table 6 AMI of 8 algorithms on 9 data sets

Data set	LI_BIFCM	K-means	DBSCAN	AP	FCM	CSPA	HGPA	MCLA
WDBC	0.6524	0.6110	0.3581	0.6020	0.6076	0.3319	− 0.0006	0.6076
Wine	0.8598	0.8301	0.5484	0.6860	0.8281	0.7326	0.4504	0.8281
Segmentation	0.6309	0.6076	0.3693	0.6050	0.6097	0.6174	0.6207	0.6207
Seeds	0.7441	0.6705	0.5302	0.6850	0.6877	0.6559	0.4340	0.6877
Dermatology	0.7344	0.6049	0.5779	0.7710	0.6592	0.5145	0.5396	0.6838
Parkinsons	0.2556	0.2129	0.0071	0.1470	0.2090	0.0927	0.1325	0.2090
Pima	0.0654	0.0503	0.0029	0.0450	0.0625	0.0489	− 0.0009	0.0625
Iris	0.7667	0.7331	0.5692	0.7560	0.7372	0.7107	0.4659	0.7372
Ionosphere	0.1266	0.1294	0.5520	0.1270	0.1184	0.1004	0.0205	0.1294
Average	0.5373	0.4944	0.3906	0.4916	0.5022	0.4228	0.2958	0.5073

Table 7 ARI sorting results of 8 algorithms on 9 data sets

Data set	LI_BIFCM	K-means	DBSCAN	AP	FCM	CSPA	HGPA	MCLA	Total
WDBC	0.2(2)	0.163(5)	-0.089(60)	0.151(8)	0.163(5)	-0.183(64)	-0.568(72)	0.163(5)	221
Wine	0.161(7)	0.125(12)	-0.193(66)	-0.106(61)	0.128(10.5)	0.037(33)	-0.28(71)	0.128(10.5)	271
Segmentation	0.062(19)	0.031(36)	-0.267(70)	0.022(41)	0.026(38.5)	0.039(31.5)	0.044(27)	0.044(27)	290
Seeds	0.123(13)	0.04(30)	-0.136(62)	0.05(24)	0.061(20.5)	0.048(25)	-0.247(68)	0.061(20.5)	263
Dermatology	0.167(3)	-0.223(67)	-0.167(63)	0.14(9)	0.076(16)	-0.047(51)	-0.056(54)	0.109(14)	277
Parkinsons	0.039(31.5)	-0.051(52)	-0.078(58)	0.024(40)	0.008(43.5)	-0.015(46)	0.066(18)	0.008(43.5)	332.5
Pima	0.044(27)	0.028(37)	-0.063(55.5)	0.015(42)	0.032(34.5)	-0.013(45)	-0.076(57)	0.032(34.5)	332.5
Iris	0.07(17)	0.042(29)	-0.063(55.5)	0.082(15)	0.054(22.5)	0.026(38.5)	-0.265(69)	0.054(22.5)	269
Ionosphere	-0.039(49.5)	-0.034(47.5)	0.472(1)	-0.039(49.5)	-0.053(53)	-0.087(59)	-0.186(65)	-0.034(47.5)	372
Total	169.0	315.5	491.0	289.5	244.0	393.0	501.0	225.0	2628
Average Rank	18.8	35.1	54.6	32.2	27.1	43.7	55.7	25.0	292

Table 8 AMI sorting results of 8 algorithms on 9 data sets

Data set	LI_BIFCM	K-means	DBSCAN	AP	FCM	CSPA	HGPA	MCLA	Total
WDBC	0.181(2)	0.14(3)	-0.113(61)	0.131(8)	0.136(5.5)	-0.139(64)	-0.472(72)	0.136(5.5)	221
Wine	0.139(4)	0.11(9)	-0.172(67)	-0.034(49)	0.108(10.5)	0.012(41)	-0.27(71)	0.108(10.5)	262
Segmentation	0.046(27)	0.022(36)	-0.216(69)	0.02(38)	0.025(33)	0.032(31)	0.036(28.5)	0.036(28.5)	291
Seeds	0.107(12)	0.034(30)	-0.107(60)	0.048(25.5)	0.051(21.5)	0.019(40)	-0.203(68)	0.051(21.5)	278.5
Dermatology	0.099(13)	-0.031(47)	-0.058(56)	0.135(7)	0.024(34)	-0.121(63)	-0.096(59)	0.048(25.5)	304.5
Parkinsons	0.097(14)	0.055(17)	-0.151(66)	-0.011(45)	0.051(21.5)	-0.066(58)	-0.026(46)	0.051(21.5)	289
Pima	0.023(35)	0.008(42)	-0.039(53)	0.003(44)	0.02(38)	0.007(43)	-0.043(54)	0.02(38)	347
Iris	0.082(15)	0.049(24)	-0.115(62)	0.072(16)	0.053(18.5)	0.026(32)	-0.219(70)	0.053(18.5)	256
Ionosphere	-0.036(51.5)	-0.034(49)	0.389(1)	-0.036(51.5)	-0.045(55)	-0.063(57)	-0.142(65)	-0.034(49)	379
Total	173.5	257.0	495.0	284.0	237.5	429.0	533.5	218.5	2628
Average Rank	19.3	28.6	55.0	31.6	26.4	47.7	59.3	24.3	292

carried out to test the number of vertical ensembles parameters in the algorithm on clustering performance, in which m fuzzy membership matrices can be obtained by running the FCM algorithm m times. In other words, the core of LI_BIFCM is to carry out m times of FCM to get the vertical ensemble and combine multiple K-nearest neighbors ($s = 5$) to get the final clustering result. The average clustering evaluation index value is obtained by repeating the algorithm 10 times. The values of m are 5, 10, 15, 20, 25, and 30, respectively, and the experimental results are shown in Fig. 6, which shows that with the number of vertical ensembles changing from 5 to 10 to 15... and to 30, no obvious fluctuation is observed in all clustering evaluation indexes.

5.4.2 The Number of Horizontal Ensembles Parameters s

In a horizontal clustering ensemble, the most critical parameter s governs the number of executions of multiple

K-nearest neighbors. Therefore, to study the influence of the parameter s on the clustering performance, the following experiments are carried out. The average clustering index of the algorithm conducted 10 times is utilized as the experimental result, assuming that the parameter m of the vertical clustering ensemble of LI_BIFCM is five. The s values are 4, 8, 12, 16, and 20. Figure 7 shows the experimental result, with the horizontal ensemble parameter s increasing from 4 to 8, then to 12, ..., and 20. The three clustering indexes did not show any noteworthy changes.

As a result, the experimental results of Figs. 6 and 7 can be summarized as follows. For the following reasons, LI_BIFCM is unaffected by the parameters of vertical and horizontal ensembles: FCM and KNN are reasonably stable; increasing the number of vertical and horizontal ensembles will only result in homogenized clustering; and the additional number of vertical and horizontal ensembles will have no effect on LI_BIFCM's performance.

Table 9 Run time of 4 algorithms on 9 data sets (time is measured in seconds)

Data set	LI_BIFCM	CSPA	HGPA	MCLA
WDBC	0.5923	0.8782	0.9151	0.8109
Wine	0.1859	0.8659	0.8110	0.8268
Segmentation	1.2934	1.8300	1.7914	1.7805
Seeds	0.1852	0.7894	0.6923	0.7042
Dermatology	0.5503	0.9615	0.968	0.948
Parkinsons	0.1716	0.7353	0.6925	0.7374
Pima	0.8971	1.0085	0.7533	0.8649
Iris	0.1473	0.7188	0.7675	0.7165
Ionosphere	0.3610	0.7711	0.7186	0.7283
Average	0.4871	0.9510	0.9011	0.9019

5.5 Run Time Comparison of 4 Algorithms

The following experiment was done to see how long LI_BIFCM takes to run. The CSPA, HGPA, and MCLA algorithms were chosen for experimental research since

the single K-means, DBSCAN, AP, and FCM algorithms are much faster than the clustering ensemble method. To acquire the final clustering results, the LI_BIFCM, CSPA, HGPA, and MCLA algorithms are used to integrate the cluster members produced by running the FCM algorithm five times. Each experiment was performed 10 times under the same conditions, with the average of the 10 running times used as the result. Table 9 shows the experimental result, with the minimal running time for each data set highlighted in bold.

Table 9 shows that LI_BIFCM takes substantially less time to run than the other three algorithms in all data sets (excluding Pima). Because the CSPA method must calculate the similarity of all sample points, the HGPA algorithm must produce a hypergraph, and the MCLA must also compute the similarity of any two clusters. The execution times of HGPA and MCLA are nearly same. LI_BIFCM not only provides a better clustering effect than other clustering ensemble algorithms when combined with the preceding experimental analysis, but it also has a shorter run time.

Fig. 6 Parameter analysis of the number of vertical ensembles **Fig. 7** Parameter analysis of the number of horizontal ensembles

6 Conclusion

In this paper, to increase clustering performance, we introduced a new clustering ensemble framework that considers local information (LI_BIFCM), dubbed the bi-directional FCM clustering ensemble algorithm. To achieve the best clustering results, LI_BIFCM combines vertical and horizontal ensembles. The properties of FCM random initial clustering centers are employed in the vertical ensemble, and FCM is run numerous times to acquire multiple cluster members for the vertical ensemble to ensure clustering stability. To avoid boundary point misdivision, multiple K-nearest neighbors are used to execute a horizontal ensemble after acquiring each cluster member. A comprehensive experimental analysis indicates: (1) LI_BIFCM surpasses FCM in dealing with boundary points. (2) On most data sets, LI_BIFCM outperforms a single clustering method as well as some clustering ensemble algorithms. (3) The vertical and horizontal ensembles parameters do not affect LI_BIFCM. (4) When compared to the clustering ensemble approach, the suggested algorithm takes less time to run. Base clustering optimization and its application to real-world scenarios, such as customer analytics, will receive greater attention in the future.

Acknowledgements This paper is supported by the National Key R&D Program of China (No. 2018YFB1701500 and No. 2018YFB1701502).

Author Contributions Literature and data collection: CR; research design: CR and LS; manuscript writing: CR; manuscript review and revise: CR and LS.

Declarations

Conflict of interest As far as we know, the author named in this paper has no financial or interest conflicts.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Xu, R., Wunsch, D.: Survey of clustering algorithms. *IEEE Trans. Neural Netw.* **16**, 645–678 (2005)
- Chaira, T.: A novel intuitionistic fuzzy C means clustering algorithm and its application to medical images. *Appl. Soft Comput.* **11**, 1711–1717 (2011)
- Kalyani, S., Swarup, K.S.: Particle swarm optimization based K-means clustering approach for security assessment in power systems. *Expert Syst. Appl.* **38**, 10839–10846 (2011)
- Hosseini, S., Maleki, A., Gholamian, M.R.: Cluster analysis using data mining approach to develop CRM methodology to assess the customer loyalty. *Expert Syst. Appl.* **37**, 5259–5264 (2010)
- Jain, A., Murty, M., Flynn, P.: Data clustering: a review. *ACM Comput. Surv.* **31**, 264–323 (1999)
- Jain, A.K.: Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **31**, 651–666 (2010)
- Dunn, J.C.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *J. Cybern.* **3**, 32–57 (1974)
- Bezdek, J.C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York (1981)
- Arqub, O.A., Al-Smadi, M.: Fuzzy conformable fractional differential equations: novel extended approach and new numerical solutions. *Soft Comput.* **24**, 12501–12522 (2020)
- Arqub, O.A., Al-Smadi, M., Momani, S., Hayat, T.: Numerical solutions of fuzzy differential equations using reproducing kernel Hilbert space method. *Soft Comput.* **20**, 3283–3302 (2016)
- Silva, T.M., Pimentel, B.A., Souza, R.M., Oliveira, A.L.: Hybrid methods for fuzzy clustering based on fuzzy c-means and improved particle swarm optimization. *Expert Syst. Appl.* **42**, 6315–6328 (2015)
- Yao, J., Dash, M., Tan, S.T., Liu, H.: Entropy-based fuzzy clustering and fuzzy modeling. *Fuzzy Set Syst.* **113**, 381–388 (2000)
- Ding, Y., Fu, X.: Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm. *Neurocomputing* **188**, 233–238 (2016)
- Zou, K., Wang, Z., Hu, M.: A new initialization method for fuzzy c means algorithm. *Fuzzy Optim. Decis. Mak.* **7**, 409–416 (2008)
- Shi, Y.L., Nana, J.Z.: Improved FCM algorithm based on initial center optimization method. *J. Intell. Fuzzy Syst.* **32**, 3487–3494 (2017)
- Ramathilaga, S., Leu, J.J., Huang, K.K., Huang, Y.M.: Two novel fuzzy clustering methods for solving data clustering problems. *J. Intell. Fuzzy Syst.* **26**, 705–719 (2014)
- Qamar, U.: A dissimilarity measure based fuzzy c-means (FCM) clustering algorithm. *J. Intell. Fuzzy Syst.* **26**, 229–238 (2014)
- Li, L., Wang, R.X., Li, X.C.: Double fuzzy C-means model and its application in the technology innovation of China. *J. Intell. Fuzzy Syst.* **31**, 2895–2901 (2016)
- Wang, X.Z., Wang, Y.D., Wang, L.J.: Improving fuzzy c-means clustering based on feature-weight learning. *Pattern Recognit. Lett.* **25**, 1123–1132 (2004)
- Haldar, N.A., Khan, F.A., Ali, A., Abbas, H.: Arrhythmia classification using Mahalanobis distance based improved Fuzzy C-Means clustering for mobile health monitoring systems. *Neurocomputing* **220**, 221–235 (2017)
- Arqub, O.A., Al-Smadi, M., Momani, S., Hayat, T.: Application of reproducing kernel algorithm for solving second-order, two-point fuzzy boundary value problems. *Soft Comput.* **21**, 7191–7206 (2016)
- Arqub, O.A.: Adaptation of reproducing kernel algorithm for solving fuzzy Fredholm–Volterra integrodifferential equations. *Neural Comput. Appl.* **28**, 1–20 (2015)
- Wu, Z.H., Wu, Z.C., Zhang, J.: An improved FCM algorithm with adaptive weights based on SA-PSO. *Neural Comput. Appl.* **28**, 3113–3118 (2017)
- Wu, Z.H., Wang, B.: DwfWfcm: an effective fuzzy c-means clustering framework considering the different data weights and feature weights. *J. Intell. Fuzzy Syst.* **37**, 4339–4347 (2019)

25. Strehl, A., Ghosh, J.: Cluster ensembles: a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **3**, 583–617 (2002)
26. Topchy, A., Jain, A.K., Punch, W.: A mixture model for clustering ensembles. In: *Proceedings of the Fourth SIAM International Conference on Data Mining*. Lake Buena Vista, FL, SIAM, USA, pp. 379–390 (2004)
27. Li, J., Gao, X.B., Tian, C.N.: FCM-based clustering algorithm ensemble for large data sets. *Fuzzy Syst. Knowl. Discov.* **4223**, 559–567 (2006)
28. Su, P., Shang, C.J., Shen, Q.: Link-based pairwise similarity matrix approach for fuzzy c-means clustering ensemble. In: *2014 IEEE International Conference on Fuzzy Systems*, Beijing, Peoples Republic of China. IEEE, USA, pp. 1538–1544 (2014)
29. Su, P., Shang, C.J., Shen, Q.: A hierarchical fuzzy cluster ensemble approach and its application to big data clustering. *J. Intell. Fuzzy Syst.* **28**, 2409–2421 (2015)
30. Ye, M., Liu, W.F., Wei, J.H., Hu, X.X.: Fuzzy c-means and cluster ensemble with random projection for big data clustering. *Math. Probl. Eng.* **2016**, 1–13 (2016)
31. Wan, X., Lin, H., Li, H., Liu, G.N., An, M.B.: Ensemble clustering via fuzzy c-means. In: *2017 14th International Conference on Services Systems And Services Management*, Dalian, Peoples Republic of China. IEEE, USA, pp. 1–6 (2017)
32. Wang, Z.C., Parvin, H., Qasem, S.N., Tuang, B.A., Pho, K.H.: Cluster ensemble selection using balanced normalized mutual information. *J. Intell. Fuzzy Syst.* **39**, 3033–3055 (2020)
33. Vinh, N., Epps, J., Bailey, J.: Information theoretic measures for clustering comparison: variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.* **11**, 2837–2854 (2010)
34. Boudane, F., Berrichi, A.: Gabriel graph-based connectivity and density for internal validity of clustering. *Prog. Artif. Intell.* **9**, 221–238 (2020)
35. Punera, K., Ghosh, J.: Consensus-based ensembles of soft clusterings. *Appl. Artif. Intell.* **22**, 780–810 (2008)
36. Strehl, A., Ghosh, J.: Cluster ensembles: a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **3**, 583–617 (2003)
37. Hu, J., Li, T.R., Luo, C., Fujita, H., Yang, Y.: Incremental fuzzy cluster ensemble learning based on rough set theory. *Knowl. Based Syst.* **132**, 144–155 (2017)
38. Garcia, S., Fernandez, A., Luengo, J., Herrera, F.: Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. *Inf. Sci.* **10**, 2044–2064 (2010)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.