Accelerated Fuzzy C-Means Clustering Based on New Affinity Filtering and Membership Scaling

Dong Li, Shuisheng Zhou, and Witold Pedrycz, Life Fellow, IEEE.

Abstract—Fuzzy C-Means (FCM) is a widely used clustering method. However, FCM and its many accelerated variants have low efficiency in the mid-to-late stage of the clustering process. In this stage, all samples are involved in the update of their non-affinity centers, and the fuzzy membership grades of the most of samples, whose assignment is unchanged, are still updated by calculating the samples-centers distances. All those lead to the algorithms converging slowly. In this paper, a new affinity filtering technique is developed to recognize a complete set of the non-affinity centers for each sample with low computations. Then, a new membership scaling technique is suggested to set the membership grades between each sample and its non-affinity centers to 0 and maintain the fuzzy membership grades for others. By integrating those two techniques, FCM based on new affinity filtering and membership scaling (AMFCM) is proposed to accelerate the whole convergence process of FCM. Many experimental results performed on synthetic and real-world data sets have shown the feasibility and efficiency of the proposed algorithm. Compared with the state-of-the-art algorithms, AMFCM is significantly faster and more effective. For example, AMFCM reduces the number of the iteration of FCM by 80% on average.

Index Terms—Fuzzy C-Means, affinity filtering, triangle inequality, non-affinity center, non-affinity sample, membership scaling.

1 INTRODUCTION

C LUSTERING analysis is one of the important topics in machine learning [1], which has been widely applied in many fields, including data mining [2], pattern recognition [3], image processing [4], etc. The clustering algorithm, which is an unsupervised learning approach, aims to divide the data sets into multiple clusters by similarity measure, among which the data points in the same cluster are similar.

In general, the clustering methods are divided into the hard and soft clustering schemes [5], [6]. The representative clustering algorithms are C-Means [7] and Fuzzy C-Means (FCM) [8]. The hard clustering scheme, in which a sample only belongs to a single cluster, assigns the membership grades between the samples and the clusters as 0 or 1. The hard clustering scheme is very simple and efficient. Inevitably, the hard clustering scheme lacks other distance information except for the closest distance information in the update of the cluster centers, which makes the algorithm more likely to fall into bad local minimum. The soft clustering scheme, in which a sample does not exclusively belong to a single cluster, allows the membership grades to vary between 0 and 1. The soft clustering scheme has better

clustering quality because of its flexibility and robustness [9].

The C-Means algorithm (Lloyd algorithm) [7] is the most representative method in the hard clustering scheme. However, the computational complexity of all samples-centers distances is very high in C-Means. Thus, many improved methods have been proposed. Both of these algorithms [10], [11] were proposed to speed up C-Means by applying a triangle inequality, which are effectively to avoid unnecessary distance calculations, and achieve higher efficiency. Another trick to deal with this challenge is region division of clusters in clustering. Some related research has been done [12], [13], [14], [15]. In recent, ball C-Means [16] has been proposed to focus on the efficiency of C-Means by reducing the samplescenters distance computations. Significantly, the concept of the neighbor clusters and the partition of cluster are designed to attain the same performance in less time by the multiple novel schemes.

As one of the most typical soft clustering methods, Fuzzy C-Means (FCM) [8] is to divide n samples into c clusters by membership grade matrix U, in which u_{ij} represents the grade jth sample belongs to ith cluster. FCM is successful in finding and describing overlapped clusters that are ubiquitous in the complex real-world data (see [2], [4], [17] and the references therein). However, all samples-centers distance computations also leads to high computing cost. Meanwhile, all samples are involved in the update of all centers by the memberships, which leads to low efficiency of FCM in the clustering process (see [18], [19])

In theory, the convergence rate theorem for FCM [20] is proved, which is that FCM converges linearly to the local minima. Meanwhile, based on the analysis of C-Means [21], [22], [23], it can be found that when C-Means is close to

D. Li, S. Zhou are with School of Mathematics and Statistics, Xidian University, Xi'an 710071, China (e-mail: lidong_xidian@foxmail.com; sszhou@mail.xidian.edu.cn).

[•] W. Pedrycz is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton T6R 2V4 AB, Canada, and with the Systems Research Institute, Polish Academy of Sciences, 00-901 Warsaw, Poland, and also with the Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia (e-mail: wpedrycz@ualberta.ca).

Manuscript received xxxx, 2022; revised xxxx, 2022. This work was supported by the National Natural Science Foundation of China under Grants No. 61772020. (Corresponding author: Shuisheng Zhou.)

the local minima, the convergence rate of C-Mean drops from an exponential rate to a linear rate. Since both C-Means and FCM are alternating optimization algorithm (AO), Therefore, likewise, the convergence rate of FCM also drops, when FCM is close to a local minima. Meanwhile, the convergence rate of FCM is slower than that of C-Means in the clustering process.

Many researchers have managed to tackle this issue based on the new update of the centers. Mitra et al. [24] have designed a Rough-Fuzzy C-Means (RFCM) clustering algorithm, which absorbs the advantages of fuzzy set and rough set and enhances the robustness and efficiency of the fuzzy clustering. Roy and Maji [25] proposed a spatially constrained Rough-Fuzzy C-Means (sRFCM), which wisely applies the advantages of rough-fuzzy clustering and local neighborhood information together. Furthermore, each cluster was divided into the prossibilistic core region and probabilistic boundary region by sRFCM, which improves the performance of the algorithm. Shadowed sets in the characterization of rough-fuzzy clustering (SRFCM) [26] was introduced to improve the clustering quality and efficiency by optimizing the threshold parameters based on the concept of shadowed set that affect the lower bound and boundary region of each cluster automatically. Similar research can be seen [27], [28], [29], [30], [31], [32] and the references therein. Unfortunately, unreasonable partition thresholds will result in undesired clustering results. Therefore, the partition parameters need to be optimized per iteration. Inevitably, the computational cost of the selection of the parameters is very high for the region partition.

To solve it, many research improve the performance of FCM by constraining the update of the memberships so that the centers can be updated to their target position more efficiently [33], [34], [35], [36], [37]. Recently, membership scaling Fuzzy C-Means clustering algorithm (MSFCM) [19] has been presented to accelerate the convergence of FCM and maintain high clustering quality, where the in-cluster and out-of-cluster samples are identified by a triangle inequality. Then, the membership grades are scaled to boost the effect of the in-cluster samples and weaken the effect of the out-of-cluster samples in the clustering process.

Although the above-mentioned FCM variants usually improve the efficiency and effectiveness of the algorithms, they ignore the low efficiency in the mid-to-late stage of the clustering process. There are three reasons: 1) the convergence rate of the alternating optimization algorithm (AO) drops, when the algorithms are in the mid-to-late stage [22]; 2) the FCM variants still need to do a full inverse-distance weighting [18]; 3) all samples are still involved in the update of all centers [19]. (see detail analysis in Subsection 3.2).

In this study, we first delve into the relationship between the samples and the centers, and further investigate the characteristic of the clustering process by dividing it into the early stage and the mid-to-late stage. Stemming from those findings, we propose a new accelerated FCM clustering algorithm called **AMFCM** (affinity filtering and **m**embership scaling based **FCM**). In the proposed algorithm, a new affinity filtering technique is put forward to precisely identify the complete set of the non-affinity centers of each sample (see **Definition** 1 in Section 3), and a new membership scaling method is suggested to accelerate the whole convergence process of the algorithm.

The main contributions of this paper are as follows:

- 1) We design a new affinity filtering scheme, which is composed of *c* triangle inequalities, to discover all samplescenters affinities. Compared with the previous methods by the triangle inequality in [11] and [19], the designed scheme can identify the complete set of the non-affinity center set of each sample more precisely with very low computational complexity. Compared with the works in [12], [24], [26], [32], the new affinity filtering scheme is parameter-free.
- 2) We propose a new membership scaling scheme to accelerate FCM convergence, especially in the mid-to-late stage. The new membership scaling scheme sets the membership grades to 0 in the update of the non-affinity centers, which eliminates the effect of the sample on the update of its non-affinity centers and reduces the burden of the fuzzy clusterings in efficiency, and maintains the original update of the remaining centers for each sample.
- 3) By integrating those schemes with FCM clustering, we propose a new accelerated clustering algorithm called AMFCM, which is the first work that focuses on accelerating the whole convergence process of the FCM-type clusterings.
- 4) Several experimental results on synthetic and realworld data sets illustrate that the proposed AMFCM outperforms the state-of-the-art algorithms in efficiency. For example, AMFCM reduces the number of the iteration of FCM by 80% on average.

The paper is organized as follows. Section 2 presents some preliminaries including notations, C-Means, FCM, and the related clustering algorithm. The research motivation is described in Section 3 and a new algorithm is presented in Section 4. The experimental results with discussion are reported in Section 6 and Section 5 concludes the paper.

2 PRELIMINARIES

In this section, some related clusterings are briefly relisted for the convenience of the following discussion.

2.1 Notations

Let a data set be $\mathbf{X} = {\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n}$ with $\mathbf{x}_j \in \mathbb{R}^p$, and the cluster centers be $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_c]$, where $\mathbf{v}_i \in \mathbb{R}^p$ is the centroid of the cluster C_i for i = 1, 2, ..., c. t is the number of iterations. The distances between \mathbf{x}_j and the cluster centers \mathbf{V} are $d_{ij} = ||\mathbf{x}_j - \mathbf{v}_i||(i = 1, \dots, c)$ and they are rearranged in ascending order as $D_j^{(1)} \leq D_j^{(2)} \leq \cdots \leq D_j^{(c)}$. Displacement of the center \mathbf{v}_i after one update is denoted by $\delta_i^{(t)} = d(\mathbf{v}_i^{(t+1)}, \mathbf{v}_i^{(t)})$. The membership grade matrix is denoted by $\mathbf{U} = [u_{ij}] \in \mathbb{R}^{c \times n}$, where u_{ij} represents the grade of *j*th sample belonging to *i*th cluster.

2.2 C-Means

C-Means clustering [7], as the most representative algorithm in the hard clustering, aims to find the c partitions of **X** by minimizing the within-cluster sum of the distance

from each sample to its nearest center. The underlying objective function is expressed as follows:

$$\min_{\mathbf{U},\mathbf{V}} J_{\text{Hard}}(\mathbf{U},\mathbf{V}) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij} \|\mathbf{x}_{j} - \mathbf{v}_{i}\|^{2},$$

$$s.t. \quad \sum_{i=1}^{c} u_{ij} = 1, u_{ij} = 0 \text{ or } 1,$$
(1)

To solve problem (1), which is NP-hard, C-Means [7] consists of two steps: the assignment step assigns each sample to its closest cluster and the update step renews each of the c cluster centers with the centroid of the samples assigned to that cluster. The algorithm repeats those two steps until convergence.

2.3 Fuzzy C-Means

FCM clustering [8], which is a soft clustering, allows a sample to have membership grades in all clusters instead of exclusively belonging to one single cluster. FCM partitions \mathbf{X} into *c* clusters by the cluster centers. The objective function is expressed as follows:

$$\min_{\mathbf{U},\mathbf{V}} J_{\mathbf{Fuzzy}}(\mathbf{U},\mathbf{V}) = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} \|\mathbf{x}_{j} - \mathbf{v}_{i}\|^{2},$$

$$s.t. \quad \sum_{i=1}^{c} u_{ij} = 1, u_{ij} \ge 0,$$
(2)

with the fuzziness weighting exponent m > 1.

To optimize problem (2), FCM usually initializes $U^{(0)}$, which is a randomly initialized partition matrix, and updates **V** and **U** iteratively by

$$\mathbf{v}_{i}^{(t+1)} = \frac{\sum_{j=1}^{n} \left(u_{ij}^{(t)}\right)^{m} \mathbf{x}_{j}}{\sum_{j=1}^{n} \left(u_{ij}^{(t)}\right)^{m}},$$
(3)

$$u_{ij}^{(t+1)} = \left[\sum_{k=1}^{c} \left(\frac{\|\mathbf{x}_j - \mathbf{v}_i^{(t+1)}\|}{\|\mathbf{x}_j - \mathbf{v}_k^{(t+1)}\|}\right)^{m-1}\right] \quad , \qquad (4)$$

until convergence.

2.4 Membership Scaling Fuzzy C-Means

Membership scaling Fuzzy C-Means clustering algorithm (MSFCM) [19] accelerates the clustering convergence and maintains high clustering quality by using a triangle inequality and membership scaling. Specifically, the triangle inequality [10], [11], which is a tool for mining the samplescenters affinities, is as follows:

Lemma 1. A sample \mathbf{x}_j cannot change its nearest cluster after one update, if

$$D_{j}^{(2)} - \max_{1 \le i \le c} \delta_{i} \ge D_{j}^{(1)} + \delta_{I_{j}^{*}},$$
(5)

where $I_j^* = \arg\min_{1 \le i \le c} \{d_{ij}\}.$

The samples whose closeness relationships do not change after one update are filtered out by using the triangle inequality (5) and Q is taken as the index set of the filtered

samples. The membership grades of the filtered samples are scaled to accelerate the convergence of FCM. Therefore, the new update scheme for $\mathbf{U}^{(t+1)}$ is as follows:

$$u_{i,j}^{(t+1)} = \begin{cases} M_j^{(t)}, & j \in Q^{(t)}, i = I_j^{*(t)}, \\ \beta_j^{(t)} u_{i,j}^{(t)}, & j \in Q^{(t)}, i \neq I_j^{*(t)}, \\ u_{i,j}^{(t)}, & j \notin Q^{(t)}, 1 \le i \le c, \end{cases}$$
(6)

where $M_j^{(t)} = \left[1 + (c-1)\left(D_j^{(1)}/D_j^{(c)}\right)^{\frac{2}{m-1}}\right]^{-1}, \beta_j^{(t)} = {}^{1-M_j^{(t)}}$ Then the undete of *V* in MCECM is also Eq. (2) In

 $\frac{1-M_j^{(t)}}{1-u_{l_{j,j}}^{(t)}}$. Then the update of **V** in MSFCM is also Eq. (3). In

[19], MSFCM also reduces the participation of the filtered samples in the update of their non-affinity centers and increases the participation of the filtered samples in the update of the remaining centers by scaling the memberships. Therefore, MSFCM has good properties, such as fewer number of iterations, lower time consumption, and higher clustering quality.

3 MOTIVATION

In this section, the relationship between the samples and the centers is first described by the following new **Definition** 1 and 2.

Definition 1. A cluster center \mathbf{v}_i is the **non-affinity center** of a sample \mathbf{x}_j , if \mathbf{v}_i cannot be the nearest center of \mathbf{x}_j in next iteration. Let \mathcal{P}_j be the set of the non-affinity centers of \mathbf{x}_j , j = 1, 2, ..., n.

Definition 2. A sample \mathbf{x}_j is the non-affinity sample of a cluster center \mathbf{v}_i , if \mathbf{x}_j cannot belong to \mathbf{v}_i in next iteration. Let C_{i-} be the set of the non-affinity samples of \mathbf{v}_i , and $\overline{C_{i-}}$ be the set of the remaining samples, i = 1, 2, ..., c.

Note that if a sample \mathbf{x}_j is the non-affinity sample of \mathbf{v}_i , \mathbf{v}_i is the non-affinity center of \mathbf{x}_j , $(i \in \mathcal{P}_j, \text{ if } j \in \mathcal{C}_{i-})$, and vice versa. The current nearest center of \mathbf{x}_j can not be the non-affinity center of \mathbf{x}_j after one iteration. That is, $|\mathcal{P}_j| \leq c-1$. Similarly, \mathbf{x}_j can not be the non-affinity sample of the current nearest center of \mathbf{x}_j .

For improving the convergence speed and clustering quality, the hierarchy of information granules [38], [39] guides the algorithms to reduce the contributions of the samples in the update of their non-affinity centers and increase the contributions of the samples in the update of the remaining centers. However, two problems need to be solved. One is how to get the set of the non-affinity centers of each sample efficiently. The other is how to formulate the modification benchmarks for the contributions of the samples in in the update of centers. These two problems motivate the proposal of two new schemes.

3.1 Searching Non-Affinity Centers by A New Affinity Filtering

In MSFCM, the affinity filtering scheme (5) can produce the non-affinity centers of the samples with low computations. For any sample \mathbf{x}_j , (5) can identify the complete nonaffinity centers of \mathbf{x}_j if $|\mathcal{P}_j| = 0$ or c - 1. However, a set of the non-affinity centers of the sample identified by (5) is



Fig. 1. Geometric explanation of identifying the non-affinity centers of \mathbf{x}_j by the affinity filtering scheme (5). For \mathbf{x}_j , centers \mathbf{v}_1 , \mathbf{v}_2 , \mathbf{v}_3 are its nearest, second-nearest, and third-nearest centers, respectively. The brown star points are the possible position of \mathbf{v}_i in next iteration. The radius of the black and gray dot-circles are δ_i and $\max_{1 \le i \le c} \delta_i$, respectively. In case (1a), \mathbf{v}_2 and \mathbf{v}_3 are identified as the complete non-affinity centers of \mathbf{x}_j by (5), where $|\mathcal{P}_j| = 2$. In case (1b) and (1c), the set of the non-affinity centers of \mathbf{x}_j is considered empty by (5). Actually, the complete set of the non-affinity centers of \mathbf{x}_j cannot be accurately identified by (5) when $|\mathcal{P}_j| \neq 0, 2$. For example, \mathbf{v}_2 is the non-affinity center of \mathbf{x}_j in case (1d), the set of the non-affinity centers of \mathbf{x}_j in case (1c). In case (1d), the set of the non-affinity centers of \mathbf{x}_j is complete because $|\mathcal{P}_j| = 0$.

incomplete when $|\mathcal{P}_j| \neq 0, c-1$. In order to illustrate this situation, a geometric interpretation is shown in Fig. 1.

In Fig. 1, centers $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are its nearest, second-nearest, and third-nearest centers for \mathbf{x}_j , respectively. Here, c = 3. The radius of red dot-arcs is the upper bound of $\mathbf{v}_1, d_{1,j} + \delta_1$, the radius of green dot-arcs is the lower bound of \mathbf{v}_2 , $d_{2,j} - \max_{1 \le i \le 3} \delta_i$, and the radius of blue dot-arcs is the lower bound of $\mathbf{v}_3, d_{3,j} - \max_{1 \le i \le 3} \delta_i$, for \mathbf{x}_j . Fig. 1a shows the situation, where $|\mathcal{P}_j| = 2$ for \mathbf{x}_j . At this time, (5) ensures that \mathbf{v}_2 and \mathbf{v}_3 are the complete non-affinity centers of \mathbf{x}_j . In Fig. 1b and 1c, the set of the non-affinity centers of \mathbf{x}_j is considered empty by (5). Actually, the complete set of the non-affinity center of \mathbf{x}_j in case (1b), and \mathbf{v}_3 is the non-affinity center of \mathbf{x}_j in case (1c). In Fig. 1d, the set of the non-affinity centers of \mathbf{x}_j identified by (5) is complete because $|\mathcal{P}_j| = 0$.

For $c \geq 3$, the affinity filtering scheme (5) always recognizes the incomplete set of non-affinity centers of the sample \mathbf{x}_j when $|\mathcal{P}_j| \neq 0, c-1$, because (5) only employs the lower bound of the second closest center of \mathbf{x}_j , $D_j^{(2)} - \max_{1 \leq i \leq c} \delta_i$, to screen all samples-centers affinities. More specifically, the lower bound of its second closest center is used to determine the affinity between \mathbf{x}_j and its second closest center, which is also inaccurate. For example, in the case of the same lower

bound of \mathbf{v}_2 , the affinity between \mathbf{x}_j and \mathbf{v}_2 is different according to **Definition** 1 in Fig. 1b and Fig. 1c. Therefore, the precise identification of all samples-centers affinities should involve the *c* new lower bounds formed by replacing $\max_{1 \le i \le c} \delta_i$ with δ_i for i = 1, 2, ..., c.

In this paper, a new affinity filtering scheme is proposed, which is composed of c new triangle inequalities. The lower bound of *i*th triangle inequality is $d_{ij} - \delta_i$ for i = 1, 2, ..., c, as shown in **Lemma** 2. In conclusion, the new affinity filtering scheme is manipulated to search the complete set of the non-affinity centers of each sample \mathbf{x}_j more precisely in any situation, where $0 \leq |\mathcal{P}_j| \leq c - 1$. To the best of our knowledge, there is no other effort that contributes to discussing the capture of the complete set of the non-affinity centers of each sample in this novel way.

3.2 Refining the Convergence of FCM

In this part, six real-world data sets are clustered by FCM with random initializations. The details of the data sets are shown in Section 6. The iterative fuzzy objective value is applied to analyze the convergence of FCM. The curves of the convergence of FCM with the iteration t are shown in Fig. 2, where the y-axis is the ratio of objectives to the initial value.



Fig. 2. Plots of $\frac{J_{\text{Fuzzy}}(\mathbf{U}^{(t)}, \mathbf{V}^{(t)})}{J_{\text{Fuzzy}}(\mathbf{U}^{(0)}, \mathbf{V}^{(0)})}$ for the iteration *t* on six real-world data sets with FCM. The initializations is selected randomly for each data set. The plots clearly show that the clustering process of FCM can be divided into stages [A] and [B], where [A] represents the early stage and [B] represents the mid-to-late stage.

In Fig. 2, it can be observed that the curves of the fuzzy objectives can be divided into stages [A] and [B], where [A] represents the early stage and [B] represents the mid-to-late stage. At the beginning of the iteration, with random initializations, the belongings of the samples are temporary and uncertain. Therefore, the membership grades are modified for good clustering results, which enlarges the displacement length of the cluster centers to reduce the objective value, as shown in stage [A]. However, the samples in other clusters always continuously interfere with the update of the centers given Eq. (3). Once FCM enters stage [B], the convergence efficiency of FCM drops rapidly [22], which can also be observed that the curves of the fuzzy objectives of FCM are quite stable and long in Fig. 2. According to this observation, it can be inferred that the cluster centers near those target positions and move in a small step in stage [B], where the assignment of most samples will not change, except for the boundary samples between clusters.

To take this question a step further, the update of \mathbf{v}_i is rewritten as $\mathbf{v}_i = (\sum_j u_{ij}^m)^{-1} (\sum_{j \in C_i} u_{ij}^m \mathbf{x}_j + \sum_{j \notin C_i} u_{ij}^m \mathbf{x}_j)$ in FCM. Clearly, $(\sum_j u_{ij}^m)^{-1} \sum_{j \in C_i} u_{ij}^m \mathbf{x}_j$ promotes \mathbf{v}_i to approach its final position, and $(\sum_j u_{ij}^m)^{-1} \sum_{j \notin C_i} u_{ij}^m \mathbf{x}_j$ prevents \mathbf{v}_i from approaching its final position, which causes the cluster centers still keep fluctuating slightly around the final targets in stage [B]. In this case, although FCM has not converged, the assignment of most samples has been determined. Actually, the fuzzy membership grades are redundant to assess uncertainty for the samples with the unchanging assignment in the current situation. Those membership grades are not only expensive to calculate, but more seriously, the update of those membership grades does not improve clustering quality. To illustrate this problem, a geometric explanation of stage [B] of the convergence process of FCM is presented, as shown in Fig.3.



Fig. 3. Geometric explanation of stage [B] of the convergence process of FCM. The thin red and blue dotted lines are the bisectors between the clusters in these two iterations, respectively. After one iteration in [B] stage, centers still have slight changes around the final targets. Note that the fluctuation of the centers, which hardly affects the convergence result, will occur many times in stage [B]. In this case, the assignment of most samples remains unchanged, and the assignment of some boundary samples, near the thin red and blue dotted lines, is vulnerable to the slight changes of the centers.

In Fig. 3, the assignment of most samples remains unchanged in stage [B]. However, centers always have slight changes around the final targets after one iteration in stage [B] because of the contributions of the samples in other clusters in the update of the centers. Note that the fluctuation of the centers, which hardly affects the convergence result, will occur many times in stage [B], resulting in the slow convergence of stage [B]. In this case, the assignment of some boundary samples, near the thin red and blue dotted lines, is vulnerable to the slight changes of the centers. Therefore, this is the main reason for the slow convergence of stage [B].

Regardless of the defects of the alternating optimization algorithm (AO), the reasons for the slow convergence of FCM can be summarized as the following two points:

- The contributions of the samples in other clusters in the update of the cluster centers continuously delay the whole stages of the convergence process of FCM;
- The fuzzy membership grades are redundant to assess uncertainty for the samples with the unchanging assignment in stage [B].

As analyzed above, the early stopping in stage [B] can improve the efficiency of the algorithm. An ideal way to do this is to pick an appropriate FCM convergence threshold, $\|\mathbf{V}^{(t+1)} - \mathbf{V}^{(t)}\|$. In fact, the selection of convergence threshold for the early stopping in stage [B] is an extremely difficult task in unsupervised learning. To illustrate this problem, $\|\mathbf{V}^{(t+1)} - \mathbf{V}^{(t)}\|$ for the iteration *t* on six realworld data sets with FCM are shown in Fig. 4, where the green point is the beginning of stage [B] of FCM. It is found that for any data, the appropriate convergence threshold for the early stopping in stage [B] is difficult to determine without prior information. Therefore, this paper adopts a new feasible approach to effectively accelerate the convergence process of FCM.



Fig. 4. Plots of $\|\mathbf{V}^{(t+1)} - \mathbf{V}^{(t)}\|$ for the iteration t on six real-world data sets with FCM. The plots show that the appropriate convergence threshold for the early stopping in stage [B] is difficult to determine without prior information.

This paper hopes to design a FCM-type clustering algorithm that is a good trade-off between clustering efficiency and quality. According to **Definition 1** and 2, the update of \mathbf{v}_i can be equivalently written in a novel way as $\mathbf{v}_i = (\sum_j u_{ij}^m)^{-1} (\sum_{j \in C_i-} u_{ij}^m \mathbf{x}_j + \sum_{j \in \overline{C_{i-}}} u_{ij}^m \mathbf{x}_j)$. Obviously, in the whole convergence stages of FCM, the contribution of the samples in C_{i-} , $(\sum_j u_{ij}^m)^{-1} (\sum_{j \in C_{i-}} u_{ij}^m \mathbf{x}_j)$, should be eliminated in the update of \mathbf{v}_i for the efficiency and quality of the algorithm. This behavior is very significant in improving the efficiency of the algorithm, especially in stage [B], where $\sum_{j \in C_{i-}} u_{ij}^m \mathbf{x}_j$ has a large proportion because the assignment of most samples in other clusters does not change. While the contributions of the samples in $\overline{C_{i-}}$, $(\sum_j u_{ij}^m)^{-1} (\sum_{j \in \overline{C_{i-}}} u_{ij}^m \mathbf{x}_j)$, should be preserved in the update of \mathbf{v}_i , which promotes algorithm convergence and ensures clustering quality. Reasonably, fuzzy membership grades are still meaningful for the samples in $\overline{C_{i-}}$ in the update of \mathbf{v}_i . The details are presented in Section 4.

4 ACCELERATED FCM BASED ON NEW AFFINITY FILTERING AND MEMBERSHIP SCALING

In this section, FCM based on new affinity filtering and membership scaling (AMFCM) is proposed to accelerate the whole convergence process of FCM and improve clustering performance. In the proposed AMFCM, a new affinity filtering method is designed to obtain the complete set of the non-affinity information more precisely by a new set of triangle inequalities. Then, a new membership scaling scheme improves the efficiency of FCM convergence in stages [A] and [B]. The operation of the new membership scaling update is two-fold, including the elimination of the contributions of the samples in the update of their non-affinity centers and enhancing the contributions of the samples in the update of the remaining centers.

4.1 New Affinity Filtering Scheme

In the modified C-Means algorithms [10], [11], the triangle inequality that appears in the affinity filtering scheme (5) is applied to identify all samples-centers affinities. However, the triangle inequality is insufficient for FCM according to the analysis in Section 3.1. Therefore, a new lemma, which is better applicable to FCM, is presented as follows:

Lemma 2. A cluster center \mathbf{v}_i is the non-affinity center of a sample \mathbf{x}_i after one update, if

$$d_{ij}^{(t)} - \delta_i^{(t)} \ge D_j^{(1)}{}^{(t)} + \delta_{I_j^*}^{(t)}, \quad i \in \{1, 2, ..., c\},$$
(7)

where $I_j^* = \arg \min_{1 \le i \le c} \{d_{ij}^{(t)}\}.$

Proof. In virtue of the triangle inequality for $i \in \{1, 2, ..., c\}$, there is that

$$d_{ij}^{(t)} - \delta_i^{(t)} = \|\mathbf{x}_j - \mathbf{v}_i^{(t)}\| - \|\mathbf{v}_i^{(t+1)} - \mathbf{v}_i^{(t)}\| \le d_{ij}^{(t+1)}.$$

Similarly, there is that

$$D_{j}^{(1)(t)} + \delta_{I_{j}^{*}}^{(t)} = \|\mathbf{x}_{j} - \mathbf{v}_{I_{j}^{*}}^{(t)}\| + \|\mathbf{v}_{I_{j}^{*}}^{(t)} - \mathbf{v}_{I_{j}^{*}}^{(t+1)}\| \ge d_{I_{j}^{*}j}^{(t+1)}.$$

If $d_{ij}^{(t)} - \delta_{i}^{(t)} \ge D_{j}^{(1)(t)} + \delta_{I_{j}^{*}}^{(t)}$ holds, we have $d_{ij}^{(t+1)} \ge d_{I_{j}^{*}j}^{(t+1)}.$
Therefore, \mathbf{v}_{i} cannot be the nearest center of \mathbf{x}_{i} after

 $d_{I_j^*j}^{(i+1)}$. Therefore, \mathbf{v}_i cannot be the nearest center of \mathbf{x}_j after one update. According to **Definition** 1, we can conclude that \mathbf{v}_i is the non-affinity center of \mathbf{x}_j .

Compared with the previous **Lemma 1**, which can filter the complete set of the non-affinity centers only when $|\mathcal{P}_j| = 0$ or c - 1 for each sample \mathbf{x}_j , **Lemma 2** provides a more efficient and precise affinity filtering scheme for exploring any situation, where $0 \le |\mathcal{P}_j| \le c - 1$.

According to **Lemma** 2, a new affinity filtering scheme, which consists of c new triangular inequalities that are the same as (7), is proposed to more precisely search the complete set of the non-affinity centers of each sample \mathbf{x}_j by employing the lower bounds of all centers in any situation, where $0 \leq |\mathcal{P}_j| \leq c - 1$. Furthermore, the new affinity filtering scheme is parameter-free, which does not need to seek extra thresholds, which determine the non-affinity centers of the samples. A geometric interpretation for the new affinity filtering scheme (7) is given in **Appendix** A.

In this part, **Lemma 2** provides a new affinity filtering scheme, which can search the complete non-affinity centers of each sample in any situation, where $0 \le |\mathcal{P}_j| \le c - 1$. As analyzed above, the computational cost of the new affinity filtering condition is very low, because the additional calculations concern only the displacements of the centers. Next, a new membership scaling scheme is proposed to accelerate the whole convergence stages of FCM in Subsection 4.2.

4.2 New Membership Scaling Scheme

As mentioned above in Subsection 3.2, the update of \mathbf{v}_i is written as $(\sum_j u_{ij}^m)^{-1} (\sum_{j \in \mathcal{C}_{i-}} u_{ij}^m \mathbf{x}_j + \sum_{j \in \overline{\mathcal{C}_{i-}}} u_{ij}^m \mathbf{x}_j)$. Based on the new affinity filtering scheme (7), which can autonomously identify the complete set of the non-affinity center of each sample \mathbf{x}_j , \mathcal{P}_j , per iteration without much computational cost and extra thresholds, it can be concluded that $\mathbf{x}_j \in \mathcal{C}_{i-}$, if $i \in \mathcal{P}_j$; otherwise, $\mathbf{x}_j \in \overline{\mathcal{C}_{i-}}$, i = 1, 2, ..., c.

In this paper, the contributions of the samples in the update of their non-affinity centers should be eliminated and the contributions of the samples in the update of the remaining centers should be increased by the membership grades for the clustering efficiency and quality of the algorithms. Therefore, a new membership scaling technique is cleverly designed to better integrate all samples-centers affinities. In the new membership scaling scheme, the membership grades are set to 0 to eliminate the contributions of the samples in the update of their non-affinity centers, which reduces the burden of the fuzzy clusterings in efficiency. And the fuzzy membership grades still be applied for the remaining centers. Here, the new membership scaling scheme for $\tilde{\mathbf{U}}^{(t)}$ is

$$\tilde{u}_{ij}^{(t)} = \begin{cases} \left[\sum_{k \notin \mathcal{P}_j^{(t)}} \left(\frac{d_{ij}^{(t)}}{d_{kj}^{(t)}} \right)^{\frac{2}{m-1}} \right]^{-1}, & i \notin \mathcal{P}_j^{(t)}, \\ & & & & \\ & & & \\ & & & & \\ & & & \\ & &$$

Note that the update of \tilde{u}_{ij} is equivalent to the normalization of the membership grades, $\tilde{u}_{ij}^{(t)} = u_{ij}^{(t)} / \sum_{k \notin \mathcal{P}_j^{(t)}} u_{kj}^{(t)}$, for $i \notin \mathcal{P}_j$. The update of **V** is the same way as FCM (Eq. (3)). Obviously, this scheme is simple and efficient. The required distances between the samples and the current centers, $d_{ij}^{(t)}$, have been calculated. Therefore, the additional calculations concern only the displacements of the centers in the determination of \mathcal{P}_j based on **Lemma** 2. The complexity analysis will be shown in Subsection 4.3.

4.3 The Proposed Algorithm

Accelerated FCM based on new affinity filtering and membership scaling (**AMFCM**) integrating with traditional one is herein proposed. In this new algorithm, after a traditional FCM iteration, the current **U** is adjusted by using the new affinity filtering and membership scaling scheme. The proposed algorithm is presented as follows in Algorithm 1.

For simplicity, the flowchart of AMFCM is as follows:

$$\mathbf{V}^{(t)} \xrightarrow{\overset{(d)}{\longrightarrow} \mathbf{U}^{(t)} \xrightarrow{(3)}} \bar{\mathbf{V}}^{(t+1)}, \delta_i^{(t)} = \| \bar{\mathbf{v}}_i^{(t+1)} - \mathbf{v}_i^{(t)} \|}_{(7), (8)} \tilde{\mathbf{U}}^{(t)} \xrightarrow{(3)} \mathbf{V}^{(t+1)}.$$

There are the following comments for AMFCM.

- As shown above, the difference between AMFCM and FCM is the calculation of U^(t), which is the novelty of AMFCM. This inserted part can improve the efficiency and quality of the clustering algorithms.
- The extra cost of AMFCM over FCM is Step 4-9. The cost in Step 4 is O(ncp), the cost of computing δ_i, (1 ≤ i ≤ c) in Step 5 is only O(cp), and the new cluster filtering technique (7) needs O(n(c-1)),

Algorithm 1 AMFCM

Input: Date set $\mathbf{X} = {\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n}$, cluster number c, fuzzy exponent m, and convergence threshold ε ; Output: Cluster center V. 1: Initialize cluster centers $\mathbf{V}^{(0)}$ and set t := 0; 2: Compute $d_{ij}^{(t)} = \|\mathbf{x}_j - \mathbf{v}_i^{(t)}\|, i = 1, ..., c, j = 1, ..., n;$ 3: Compute $\mathbf{U}^{(t)}$ with $u_{ij}^{(t)} = \left[\sum_{k=1}^{c} \left(d_{ij}^{(t)} / d_{kj}^{(t)} \right)^{\frac{2}{m-1}} \right]^{-1}$; 4: Compute $\bar{\mathbf{V}}^{(t+1)}$ with $\bar{\mathbf{v}}_{i}^{(t+1)} = \frac{\sum_{j=1}^{n} (u_{ij}^{(t)})^{m} \mathbf{x}_{j}}{\sum_{j=1}^{n} (u_{ij}^{(t)})^{m}}$; 5: Compute $\delta_{i}^{(t)} = \|\bar{\mathbf{v}}_{i}^{(t+1)} - \mathbf{v}_{i}^{(t)}\|$ for i = 1, 2, ..., c; 6: **for** j = 1 to n **do** $\mathcal{P}_{j}^{(t)} = \{ 1 \le i \le c \mid d_{ij}^{(t)} - \delta_{i}^{(t)} \ge D_{j}^{(1)}^{(1)} + \delta_{I_{i}^{*}}^{(t)} \};$ 7: Compute $\tilde{u}_{ij}^{(t)}$ according to Eq. (8); 8: 9: end for 9: end for 10: Compute $\mathbf{V}^{(t+1)}$ with $\mathbf{v}_i^{(t+1)} = \frac{\sum_{j=1}^n (\tilde{u}_{ij}^{(t)})^m \mathbf{x}_j}{\sum_{j=1}^n (\tilde{u}_{ij}^{(t)})^m};$ 11: if $\|\mathbf{V}^{(t+1)} - \mathbf{V}^{(t)}\| \ge \varepsilon$ then 12: Set t := t + 1; 13: Goto Step 2; 14: else return $\mathbf{V} = \mathbf{V}^{(t+1)}$; 15: 16: end if

where $d_{ij}^{(t)}$ in (7) and (8) has been calculated in Step 2. Therefore, the cost of AMFCM is $\mathcal{O}(3ncp)$ per iteration (the cost of FCM is $\mathcal{O}(2ncp)$ per iteration).

- The time complexity of FCM is $\mathcal{O}(nc^2pt_{\text{FCM}})$ [40], [41], where t_{FCM} is the iteration of FCM. For the time complexity of AMFCM, the update of the centers $\mathbf{V}_{\text{AMFCM}}$ requires $\mathcal{O}((\sum_{j=1}^{n} |\mathcal{P}_j|)p)$ per iteration, where $1 \leq |\mathcal{P}_j| \leq c$ for j = 1, 2, ..., n. With the update of the membership grade matrix $\tilde{\mathbf{U}}$, the time complexity of AMFCM is $\mathcal{O}(nc^2pt_{\text{AMFCM}})$, where t_{AMFCM} is the iteration of AMFCM. Based on **Theorem** 1, $t_{\text{AMFCM}} < t_{\text{FCM}}$. As a result, AMFCM can save the running time.
- For any membership grade of AMFCM, ũ_{ij} ∈ [0, 1] for i = 1, 2, ..., c and j = 1, 2, ..., n, based on Eq. (8). Obviously, AMFCM is a combination of the hard and soft clustering algorithms, which sets the membership grades to 0 for eliminating the contributions of the samples in the update of their non-affinity centers. Meanwhile, the fuzzy-type update way is performed for the remaining centers. In particular, AMFCM is a parameter-free and adaptive clustering algorithm, which can autonomously determine all samples-centers affinities and the update method.

5 THEORETICAL ANALYSIS

In this section, the convergence properties of AMFCM are provided. First, the flowchart of FCM and AMFCM in *t* iteration are as follows:

$$\begin{aligned} \mathbf{FCM}: \quad \mathbf{V}^{(t)} &\xrightarrow{(4)} \mathbf{U}_{\mathbf{FCM}}^{(t)} \xrightarrow{(3)} \mathbf{V}_{\mathbf{FCM}}^{(t+1)}.\\ \mathbf{AMFCM}: \quad \mathbf{V}^{(t)} &\xrightarrow{\overset{(4)}{\longrightarrow} \mathbf{U}_{\mathbf{FCM}}^{(t)} \xrightarrow{(3)} \mathbf{V}_{\mathbf{FCM}}^{(t+1)}}{\delta_i^{(t)} = \|\mathbf{v}_{\mathbf{FCM}}^{(t+1)} - \mathbf{v}_i^{(t)}\|, (7), (8)} & \tilde{\mathbf{U}}^{(t)} \xrightarrow{(3)} \mathbf{V}_{\mathbf{AMFCI}}^{(t+1)}. \end{aligned}$$

Theorem 1. The number of the iteration of AMFCM is smaller than that of FCM in the clustering process.

Proof. The proof is given in **Appendix B**.
$$\Box$$

Theorem 2. *AMFCM does not converge precociously in the mid stage of the clustering process.*

Proof. The proof is given in **Appendix**
$$\mathbb{C}$$
.

Theorem 3. For the samples \mathbf{x}_j with $|\mathcal{P}_j^{(t)}| = 1$, the corresponding hard objective value of AMFCM is smaller than that of FCM in t iteration.

Proof. The proof is given in **Appendix D**.
$$\Box$$

In the next section, several experiments are performed to illustrate the efficiency of the proposed algorithm.

6 **EXPERIMENTAL RESULTS**

To verify the effectiveness and efficiency of the proposed algorithm, experimental studies are carried out on synthetic and real-world data sets, respectively. AMFCM is compared with another seven clustering algorithms, including:

- 1) Fuzzy C-Means (FCM) [8],
- Rough Fuzzy C-Means (RFCM) [24],
- Shadowed Set-based Rough C-Means (SRCM), Shadowed Set-based Rough Fuzzy C-Means I (SRFCM I), and Shadowed Set-based Rough Fuzzy C-Means II (SRFCM II) [26],
- Rough-Fuzzy Clustering based on Two-stage Threeway Approximations (ARFCM) [32],
- 5) Membership Scaling Fuzzy C-Means (MSFCM) [19].

These algorithms are chosen because they use different techniques to reduce the contributions of the samples in the update of their non-affinity centers and increase the contributions of the samples in the update of the remaining centers for good clustering quality and fast convergence.

All experiments are run on a computer with an Intel Core i7-6700 processor and a maximum memory of 8GB for all processes; the computer runs Windows 7 with MATLAB R2017a. The experimental setup and the evaluation metrics used for clustering performance are first described. The fuzziness weighting exponent m = 2 and the termination parameter $\varepsilon = 10^{-6}$ for all algorithms. In addition, the weight exponent of the core region $w_l = 0.95$ and the weight exponent of the boundary region $w_b = 1 - w_l$ for RFCM [24], SRCM, SRFCM I and SRFCM II [26], ARFCM [32].

6.1 Evaluation Metrics

In order to evaluate the performances of the newly proposed clustering algorithms, three external metrics, including the overall F-measure for the entire data set (**F**^{*}), Normalized Mutual Information (**NMI**), and Adjusted Rand Index (**ARI**) [42], [43], [44], are used. All three metrics are used to measure the agreement of the ground truth and the clustering results produced by an algorithm. The metrics that do not require the labels of data are also used for **M**[•] performance evaluation, called the internal metrics. The



Fig. 5. The convergence trajectories of FCM, MSFCM and AMFCM with same initializations on data sets D1 and D2. The convergence trajectories of the three algorithms on D1 are put together, as shown in Fig. 5a. The convergence trajectories of the three algorithms on D2 are as shown in Fig. 5b, 5c and 5d, respectively. On D1, FCM, MSFCM and AMFCM are converged with 12 (t_A =2; t_B =10), 10 (t_A =2; t_B =8), and 8 (t_A =2; t_B =6) iterations and take 0.1294, 0.0781, and 0.054 seconds, respectively. On D2, FCM, MSFCM and AMFCM are converged with 30 (t_A =8; t_B =22), 18 (t_A =6; t_B =12), and 10 (t_A =3; t_B =7) iterations and take 0.2830, 0.1094, and 0.066 seconds, respectively.

three internal validity metrics are selected, including PC [45], DBI [46], and XB [47].

$$\mathbf{PC} = \frac{1}{n} \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{2},$$
(9)

$$\mathbf{DBI} = \frac{1}{c} \sum_{k=1}^{c} \max_{i \neq k} \frac{\frac{1}{|C_i|} \sum_{\mathbf{x}_j \in C_i} d_{ij}^2 + \frac{1}{|C_k|} \sum_{\mathbf{x}_j \in C_k} d_{kj}^2}{\|\mathbf{v}_i - \mathbf{v}_k\|^2}, \quad (10)$$

$$\mathbf{XB} = \frac{\sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2}}{n \min_{i \neq k} \|\mathbf{v}_{k} - \mathbf{v}_{i}\|^{2}}.$$
 (11)

Note that **Time** and **Iteration** are the remaining two evaluation metrics for expressing the efficiency of the algorithms.

6.2 Experiments on Synthetic Data Sets

In the first set of experiments, to test the efficiency of AMFCM in the whole convergence stages, two synthetic data sets in \mathbb{R}^2 are implemented to observe the convergence path. The first synthetic data set contains three clusters, which are generalized by the two-dimensional Gaussian distribution with mean vector μ_i and covariance matrix Σ_i , i = 1, 2, 3. The number of data in each cluster is 200, and the corresponding parameters μ_i and Σ_i are $\mu_1 = [10, 10]$, $\Sigma_1 = \begin{bmatrix} 0.3 & 0 \\ 0 & 0.3 \end{bmatrix}$, $\mu_2 = [13, 10]$, $\Sigma_2 = \begin{bmatrix} 0.8 & 0 \\ 0 & 0.8 \end{bmatrix}$, and $\mu_3 = [11, 4]$, $\Sigma_3 = \begin{bmatrix} 1.2 & 0 \\ 0 & 1.2 \end{bmatrix}$, respectively. To further reflect the effect of the contributions of the samples in other clusters on the convergence of the algorithms, the second synthetic data set is designed, in which some samples are added to the first synthetic data set. These two synthetic data sets are called D1 and D2, respectively.

To illustrate intuitively, the visualized figures of the convergence trajectories of FCM, MSFCM, and AMFCM on D1 and D2 for comparison in this part, where same initializations are selected, are shown as Fig. 5. Time and **Iterations** are selected to characterize the performance of the algorithms. Here, t_A and t_B are defined as the number of iteration of the algorithm in stages [A] and [B], respectively.

First of all, the convergence trajectories of FCM, MSFCM, and AMFCM on D1 are similar, where the total contributions of the samples in the update of their non-affinity centers are small and not enough to observe. Therefore, the convergence trajectories of the three algorithms on D1 are put together, as shown in Fig. 5a. The experimental results on D1 show that AMFCM performs best, as shown in Fig. 5. From the convergence trajectories on D1, it is observed that although $t_A=2$ for FCM, MSFCM and AMFCM, t_B of AMFCM is the least with only 6 iterations. On D2, the contributions of the new added samples in the update of their non-affinity centers mislead the displacement direction of the centers by the membership grades for FCM and MSFCM. Therefore, stages [A] and [B] of FCM and MSFCM have been extended, where $t_A=8$ and $t_B=22$ for FCM; $t_A=6$ and $t_{\rm B}$ =12 for MSFCM. However, AMFCM completely eliminates the misleading of the total contributions of the newly added samples in the update of their non-affinity centers, which achieves better performance. From the convergence trajectory of AMFCM on D2, t_A =3 and t_B =7 for AMFCM.

Secondly, it is observed that the efficiency of MSFCM is higher than that of FCM. But the performance of MSFCM is limited. As analyzed in Section 3.1, the previous affinity filtering (5) of MSFCM fails to obtain the complete set of the non-affinity centers of each sample on D1 and D2, where $|\mathcal{P}| = 1$ for some samples in this stage. Therefore, MSFCM does not always maintain high efficiency, because the membership scaling (5) is invalid. However, AMFCM makes up for this shortcoming.

AMFCM accelerates the whole convergence process of FCM under same initializations. Both stages [A] and [B] have been accelerated. Especially, AMFCM can save 67% of the number of the iteration of FCM on D2. As the previous complexity analysis, the running time of AMFCM is decreased. Thus, it can be concluded that the new affinity filtering scheme (7) is implemented with high efficiency, and the new membership scaling scheme (8) is outstanding in terms of efficiency and clustering quality.

6.3 Experiments on Real-World Data Sets

In this subsection, some experiments are done to verify the clustering efficiency and performance of AMFCM on real-world data sets.

6.3.1 Acceleration and Performance of AMFCM

To verify the acceleration of AMFCM in stages [A] and [B] on real-world data sets, the experiments in Fig. 2 are redone by AMFCM with the same settings, and the corresponding hard-objective of AMFCM are also displayed, which is to illustrate the clustering characteristics of AM-FCM, as shown in Fig. 6.



Fig. 6. Plots of $\frac{J_{\text{Fuzzy}}(\mathbf{U}^{(t)}, \mathbf{V}^{(t)})}{J_{\text{Fuzzy}}(\mathbf{U}^{(0)}, \mathbf{V}^{(0)})}$ and $\frac{J_{\text{Hard}}(\mathbf{U}^{(t)}, \mathbf{V}^{(t)})}{J_{\text{Hard}}(\mathbf{U}^{(0)}, \mathbf{V}^{(0)})}$ for iteration *t* on six real-world data sets with AMFCM. The initializations is selected randomly for each data set. The plots clearly show that the clustering process of AMFCM can be divided into stages [A] and [B], where [A] represents the early stage and [B] represents the mid-to-late stage.

Similar to FCM, the fuzzy-objective and the corresponding hard-objective of AMFCM also can be divided into stages [A] and [B]. However, compared with Fig. 2, the number of the iteration of AMFCM is much lower than that of FCM, which saves at least 76% of the total rounds of the iteration on these six real-world data sets. Meanwhile, stages [A] and [B] of the convergence process of AMFCM are terminated earlier than the corresponding FCM, which is attributed to the new affinity filtering and membership scaling schemes, as shown in Fig. 6.

Recently, Nie *et al.* [48] mentioned that bad local minimum makes the objective value not small enough, which limits the performance of the algorithms. According to this, the comparison results of the fuzzy and corresponding hard objective value of FCM and AMFCM on these six real-world data sets are displayed in TABLE 1.

TABLE 1 Comparison for the fuzzy and hard objective value with FCM and AMFCM. The values are averaged over 10 trials with random initializations. The best results are shown in boldface.

Data sets	Fuzzy Obje	ective Value	Hard Objective Value		
	FCM	AMFCM	FCM	AMFCM	
Arcene	4.297E+04	4.439E+04	6.635E+04	6.261E+04	
DrivFace	6.079E+04	6.362E+04	1.315E+05	1.101 ± 05	
Led7	4.302E+02	3.709E+02	3.468E+03	$1.468 \pm +03$	
Satimage	5.988E+02	6.474E+02	1.764E+03	1.389E+03	
Shuttle	1.383E+02	1.326E+02	2.925E+02	2.588E+02	
Sensorless	1.869E+03	1.919E+03	6.243E+03	4.558E+03	

In TABLE 1, the fuzzy objective value cannot achieve a small value because some values derived by the ordinary optimization theory are modified by AMFCM. The fuzzy objective is sacrificed for the efficiency of the algorithm. However, AMFCM can increase the membership of each sample to its nearest center through Eq. 8, so that the corresponding hard objective of AMFCM is continuously optimized. Therefore, the corresponding hard objective value of AMFCM is better than that of FCM. Furthermore, AMFCM not only greatly improves the efficiency of FCM, but also maintains better clustering performance.

In order to display the performance of AMFCM more comprehensively, AMFCM is compared with the seven chosen clustering algorithms on the above eight evaluation metrics. Moreover, all real-world data sets, which are selected from UCI Machine Learning Repository¹, are clustered by the chosen clustering algorithms. The detailed information on the data sets is given in each table title, where *n* is the number of training size, *p* is the dimensionality of samples, and *c* is the given number of clusters. The values are averaged over 10 trials with random initializations and the standard deviations are given after the means (linked with \pm), and the best results are shown in boldface. In addition, the corresponding **Iteration** and **Time** of the different algorithms on ten real-world data sets are shown in Fig. 7.

From the experimental results in TABLE 2 and Fig. 7, the following conclusions are obtained.

- Comparing the first and last columns of each data set, AMFCM has better performance than FCM on all data sets in terms of the above six evaluation metrics in TABLE 2. Moreover, it is worth mentioning that the efficiency of AMFCM has been improved in the whole stages, which reduces the number of the iteration of FCM by 80% on average without significant computational cost in Fig. 7. Therefore, AMFCM has also achieved significant savings in running time. As shown in Fig. 7, the total **Iteration** and **Time** of AMFCM is much lower than that of other algorithms.
- According to [19], it is found that the cost of AMFCM and MSFCM are both O(3ncp) per iteration. For the experimental results of MSFCM in the penultimate column of each table, the acceleration of MSFCM on the two data sets failed because the affinity filtering(5) fails to obtain the complete non-affinity information as analyzed in Section 3.1, and the low efficiency of the membership scaling (6), which will be explained in the next Subsection 6.3.2. In the remaining eight data sets, although MSFCM is effective, the acceleration performance of AMFCM in the whole stages is better than that of MSFCM. Thus, AMFCM is a successful generalization to MSFCM.
- The remaining five algorithms sometimes have better clustering quality than AMFCM, which is due to the good parameters for the division of each cluster. In this case, the clustering efficiency of the algorithms is reduced, as shown in Fig. 7. However, searching the complete set of the non-affinity centers is a parameter-free and autonomous process for AMFCM. In summary, AMFCM is a good trade-off between efficiency and quality.

According to the summary above, AMFCM can greatly improve clustering efficiency and quality on real-world data

TABLE 2

Experimental results on ten real-world data sets for different algorithms. The values are averaged over 10 trials with random initializations. The standard deviations are given after the means (linked with ±), and the best results are shown in boldface.

Data sots	Motrice 1	FCM	PECM	SPCM	SPECM I	SPECM II	ARECM	MSECM	AMECM
	Metrics -	rcivi	KrCivi	эксій	SKFCWII	SKICIVI II	AKICINI	WISPCIVI	AIVITCIVI
	$PC \uparrow$	0.655 ± 0.001	0.671 ± 0.001	0.663 ± 0.023	0.698 ± 0.001	0.652 ± 0.001	$0.710 {\pm} 0.001$	0.655 ± 0.001	0.658 ± 0.001
Arcene	DBI↓	1.015 ± 0.001	1.031 ± 0.001	1.042 ± 0.167	0.857 ± 0.001	1.124 ± 0.002	1.000 ± 0.001	1.015 ± 0.001	$0.831 {\pm} 0.001$
n=200	XB↓	0.379 ± 0.001	0.357 ± 0.001	0.386 ± 0.001	0.337 ± 0.001	0.405 ± 0.001	0.335 ± 0.001	0.379 ± 0.001	$0.316 {\pm} 0.001$
p=10000	$F^{*}\uparrow$	0.583 ± 0.001	0.641 ± 0.001	0.633 ± 0.004	0.586 ± 0.001	0.643 ± 0.001	0.649 ± 0.001	0.583 ± 0.001	$0.654{\pm}0.001$
<i>c</i> =2	ARI↑	0.027 ± 0.001	0.089 ± 0.001	0.074 ± 0.003	0.030 ± 0.001	0.090 ± 0.001	0.091 ± 0.001	0.027 ± 0.001	0.091 ± 0.001
	NMI↑	0.018 ± 0.001	0.091 ± 0.001	0.066 ± 0.001	0.020 ± 0.001	0.086 ± 0.001	0.085 ± 0.001	0.018 ± 0.001	0.087±0.001
	PC ↑	$0.420 {\pm} 0.001$	$0.489 {\pm} 0.001$	$0.479 {\pm} 0.010$	$0.487 {\pm} 0.006$	$0.477 {\pm} 0.009$	$0.445 {\pm} 0.001$	$0.461 {\pm} 0.001$	$0.490{\pm}0.001$
DrivFace	$DBI \downarrow$	15.7 ± 0.1	2.641 ± 0.011	3.638 ± 1.375	$2.258 {\pm} .0461$	$4.964{\pm}1.809$	2.666 ± 0.810	2.677 ± 0.001	$1.644{\pm}0.001$
n=606	$XB\downarrow$	5.354 ± 0.001	$0.897 {\pm} 0.004$	$1.685 {\pm} 0.425$	1.187 ± 0.267	1.732 ± 0.601	$0.936 {\pm} 0.072$	0.902 ± 0.001	$0.567{\pm}0.001$
p=6400	$F^* \uparrow$	$0.558 {\pm} 0.001$	$0.576 {\pm} 0.005$	$0.541 {\pm} 0.001$	$0.576 {\pm} 0.004$	$0.571 {\pm} 0.003$	$0.566 {\pm} 0.001$	$0.587 {\pm} 0.001$	$0.597{\pm}0.001$
c=3	ARI ↑	0.016 ± 0.001	0.021 ± 0.001	$0.008 {\pm} 0.001$	0.006 ± 0.001	0.020 ± 0.003	0.019 ± 0.001	0.019 ± 0.001	$0.024{\pm}0.001$
	NMI ↑	0.053 ± 0.001	0.054 ± 0.001	0.030 ± 0.002	0.035 ± 0.007	0.045 ± 0.002	0.054 ± 0.001	0.053 ± 0.001	$0.057 {\pm} 0.001$
	PC ↑	$0.007{\pm}0.001$	$0.007{\pm}0.001$	$0.006 {\pm} 0.003$	$0.007{\pm}0.001$	$0.007 {\pm} 0.002$	$0.005 {\pm} 0.001$	$0.007{\pm}0.001$	$0.007{\pm}0.001$
Feret	$DBI\downarrow$	2.729 ± 0.106	$2.630 {\pm} 0.186$	2.832 ± 0.389	$2.311 {\pm} 0.834$	7.267 ± 3.920	$1.963 {\pm} 0.345$	1.729 ± 0.259	0.927 ±0.122
n=1400	$XB\downarrow$	0.127 ± 0.001	0.292 ± 0.001	$0.156 {\pm} 0.041$	0.062 ± 0.010	$5.984{\pm}4.184$	$0.056 {\pm} 0.001$	$0.088 {\pm} 0.001$	0.053±0.009
p=1600	$F^* \uparrow$	$0.133 {\pm} 0.005$	$0.119 {\pm} 0.005$	$0.144{\pm}0.026$	$0.146 {\pm} 0.012$	$0.143 {\pm} 0.022$	$0.150 {\pm} 0.025$	$0.119 {\pm} 0.030$	$0.153 {\pm} 0.002$
c=200	ARI ↑	0.016 ± 0.034	$0.016 {\pm} 0.020$	0.022 ± 0.009	$0.024{\pm}0.011$	$0.019 {\pm} 0.005$	0.021 ± 0.006	0.021 ± 0.003	$0.025{\pm}0.003$
	NMI ↑	0.415 ± 0.009	0.382 ± 0.015	0.415 ± 0.028	0.419 ± 0.021	0.402 ± 0.019	0.425 ± 0.017	0.419 ± 0.058	0.478 ±0.034
	PC ↑	0.055 ± 0.001	$0.056 {\pm} 0.006$	$0.091 {\pm} 0.009$	$0.097 {\pm} 0.009$	$0.102 {\pm} 0.009$	$0.055 {\pm} 0.001$	$0.058 {\pm} 0.005$	0.108 ±0.018
COIL20	DBI↓	1.776 ± 0.225	$2.961 {\pm} 0.154$	42.3 ± 21.2	42.3 ± 1.7	37.9 ± 15.6	$1.276 {\pm} 0.074$	$1.787 {\pm} 0.001$	0.847±0.012
n=1440	$XB\downarrow$	1.077 ± 0.001	$1.099 {\pm} 0.014$	9.7±3.7	10.1 ± 3.0	14.2 ± 6.2	$0.297 {\pm} 0.001$	$0.970 {\pm} 0.003$	0.127±0.005
p=1024	$F^* \uparrow$	$0.242 {\pm} 0.028$	$0.228 {\pm} 0.001$	$0.282 {\pm} 0.041$	$0.401 {\pm} 0.032$	$0.397 {\pm} 0.010$	$0.271 {\pm} 0.001$	$0.260 {\pm} 0.015$	0.443 ±0.060
c=20	ARI ↑	$0.110 {\pm} 0.036$	$0.133 {\pm} 0.001$	$0.248 {\pm} 0.058$	$0.237 {\pm} 0.026$	$0.254{\pm}0.017$	$0.279 {\pm} 0.002$	$0.110 {\pm} 0.023$	0.286 ±0.068
	NMI ↑	0.299 ± 0.042	$0.372 {\pm} 0.001$	$0.339 {\pm} 0.045$	$0.385 {\pm} 0.020$	$0.317{\pm}0.004$	$0.301 {\pm} 0.003$	$0.374 {\pm} 0.017$	$0.574 {\pm} 0.059$
	PC ↑	0.229±0.027	$0.521 {\pm} 0.021$	$0.281 {\pm} 0.043$	$0.421 {\pm} 0.064$	$0.371 {\pm} 0.070$	0.593 ±0.026	0.351 ± 0.221	$0.509 {\pm} 0.051$
Led7	DBI↓	0.986 ± 0.131	1.145 ± 0.100	2.852 ± 0.575	1.454 ± 0.229	1.807 ± 0.229	$1.160 {\pm} 0.206$	$0.983 {\pm} 0.117$	0.857±0.105
n=3200	XB↓	$0.178 {\pm} 0.011$	$0.132 {\pm} 0.008$	$0.191 {\pm} 0.015$	$0.165 {\pm} 0.022$	$0.223 {\pm} 0.042$	$0.144{\pm}0.027$	$0.164{\pm}0.032$	$0.122{\pm}0.002$
<i>p</i> =7	$F^* \uparrow$	$0.424 {\pm} 0.001$	$0.614{\pm}0.049$	$0.591 {\pm} 0.055$	$0.618 {\pm} 0.111$	$0.586 {\pm} 0.018$	$0.694 {\pm} 0.025$	$0.487 {\pm} 0.165$	0.731±0.007
c=10	ARI ↑	0.232 ± 0.001	$0.415 {\pm} 0.043$	$0.376 {\pm} 0.063$	$0.405 {\pm} 0.101$	$0.356 {\pm} 0.034$	$0.438 {\pm} 0.021$	$0.436 {\pm} 0.152$	0.497±0.007
	NMI ↑	$0.366 {\pm} 0.001$	$0.498 {\pm} 0.038$	$0.465 {\pm} 0.052$	$0.494{\pm}0.077$	$0.466 {\pm} 0.033$	$0.510 {\pm} 0.016$	$0.473 {\pm} 0.104$	0.563±0.019
	PC ↑	0.390±0.001	$0.432 {\pm} 0.029$	$0.315 {\pm} 0.045$	$0.351 {\pm} 0.010$	$0.341{\pm}0.038$	$0.408 {\pm} 0.019$	$0.448 {\pm} 0.028$	0.476±0.001
Satimage	DBI↓	$4.880 {\pm} 0.001$	$2.464 {\pm} 0.862$	$39.4{\pm}12.4$	8.906 ± 3.432	$28.8 {\pm} 10.9$	3.681 ± 1.310	2.261 ± 1.048	$0.908 {\pm} 0.001$
n=6435	XB↓	$3.478 {\pm} 0.001$	1.112 ± 0.023	25.2 ± 3.1	$1.612 {\pm} 0.419$	8.032 ± 2.187	$1.180 {\pm} 0.186$	$0.493 {\pm} 0.036$	$0.455{\pm}0.001$
<i>p</i> =36	$F^* \uparrow$	$0.553 {\pm} 0.001$	$0.593 {\pm} 0.029$	$0.631 {\pm} 0.049$	$0.638 {\pm} 0.062$	$0.612 {\pm} 0.080$	$0.635 {\pm} 0.009$	$0.638 {\pm} 0.013$	$0.659 {\pm} 0.001$
c=6	ARI ↑	0.292 ± 0.001	$0.317 {\pm} 0.027$	$0.348 {\pm} 0.051$	$0.358 {\pm} 0.071$	$0.337 {\pm} 0.018$	$0.389 {\pm} 0.005$	$0.406 {\pm} 0.019$	$0.443 {\pm} 0.001$
	NMI ↑	0.450 ± 0.001	0.461 ± 0.023	0.457 ± 0.029	0.457 ± 0.034	0.446 ± 0.062	$0.471 {\pm} 0.007$	$0.493 {\pm} 0.006$	$0.515 {\pm} 0.001$
	PC ↑	0.656 ± 0.001	$0.720 {\pm} 0.001$	$0.733 {\pm} 0.001$	$0.717 {\pm} 0.001$	$0.702 {\pm} 0.001$	$0.679 {\pm} 0.036$	$0.656 {\pm} 0.001$	0.729±0.001
Magic	DBI↓	$1.886 {\pm} 0.001$	$1.549 {\pm} 0.001$	$1.542 {\pm} 0.001$	$1.539 {\pm} 0.001$	$1.539 {\pm} 0.001$	$1.490 {\pm} 0.005$	$1.886 {\pm} 0.001$	$1.050 {\pm} 0.001$
n=19020	$XB\downarrow$	$0.545 {\pm} 0.001$	$0.390 {\pm} 0.001$	$0.373 {\pm} 0.001$	$0.381 {\pm} 0.001$	$0.372 {\pm} 0.001$	$0.438 {\pm} 0.010$	$0.545 {\pm} 0.001$	$0.266 {\pm} 0.001$
p=10	$F^* \uparrow$	$0.582 {\pm} 0.001$	$0.627 {\pm} 0.001$	$0.622 {\pm} 0.001$	$0.633 {\pm} 0.001$	$0.612 {\pm} 0.001$	$0.591 {\pm} 0.003$	$0.582 {\pm} 0.001$	$0.641 {\pm} 0.001$
c=2	ARI ↑	0.007 ± 0.001	$0.013 {\pm} 0.001$	$0.018 {\pm} 0.001$	$0.014 {\pm} 0.001$	$0.018 {\pm} 0.001$	$0.008 {\pm} 0.001$	0.007 ± 0.001	$0.020{\pm}0.001$
	NMI ↑	0.020 ± 0.001	0.043 ± 0.001	0.051 ± 0.001	0.046 ± 0.001	0.053 ± 0.001	0.053 ± 0.001	0.020 ± 0.001	$0.057 {\pm} 0.001$
	PC ↑	$0.362 {\pm} 0.001$	$0.447{\pm}0.035$	$0.353 {\pm} 0.006$	$0.398 {\pm} 0.017$	$0.349 {\pm} 0.032$	$0.330 {\pm} 0.032$	$0.409 {\pm} 0.006$	0.513±0.049
Shuttle	$DBI\downarrow$	290.5 ± 0.8	157.8 ± 27.4	307.7±124.3	230.4 ± 30.2	244.6 ± 27.1	287.9 ± 63.9	153.6 ± 19.4	52.2±33.7
n=58000	$XB\downarrow$	98.7±1.6	$9.7{\pm}0.4$	13.5 ± 1.3	13.0 ± 3.9	12.2 ± 0.4	65.1 ± 8.41	15.4 ± 0.1	7.8 ±3.6
p=9	$F^* \uparrow$	$0.504 {\pm} 0.001$	$0.593 {\pm} 0.010$	$0.546 {\pm} 0.061$	$0.578 {\pm} 0.062$	$0.571 {\pm} 0.010$	$0.460 {\pm} 0.056$	$0.512 {\pm} 0.043$	0.667 ±0.071
<i>c</i> =7	ARI ↑	0.114 ± 0.001	$0.157 {\pm} 0.026$	$0.117 {\pm} 0.054$	$0.157 {\pm} 0.073$	$0.149 {\pm} 0.007$	$0.085 {\pm} 0.022$	$0.153 {\pm} 0.052$	0.190 ±0.062
	NMI ↑	0.218 ± 0.001	0.257 ± 0.027	0.201 ± 0.066	0.226 ± 0.023	0.206 ± 0.030	0.212 ± 0.036	$0.218 {\pm} 0.060$	0.260 ±0.016
	PC ↑	$0.263 {\pm} 0.016$	0.251 ± 0.023	0.202 ± 0.026	$0.191 {\pm} 0.024$	$0.112 {\pm} 0.004$	$0.241 {\pm} 0.016$	0.295 ± 0.023	$0.309{\pm}0.004$
Sensorless	DBI↓	3.206 ± 0.091	$5.304{\pm}0.951$	156.7 ± 32.4	31.1 ± 28.6	21.6 ± 9.4	5.968 ± 3.370	$1.869 {\pm} 0.507$	$0.814{\pm}0.054$
n=58509	$XB\downarrow$	$1.494{\pm}0.067$	$3.051 {\pm} 0.667$	$35.5 {\pm} 14.7$	$25.2{\pm}14.5$	$15.9 {\pm} 8.6$	$3.906{\pm}1.513$	$1.195 {\pm} 0.635$	$0.326{\pm}0.052$
p=48	$F^* \uparrow$	$0.307 {\pm} 0.014$	$0.282{\pm}0.021$	$0.281 {\pm} 0.017$	$0.271 {\pm} 0.020$	$0.288 {\pm} 0.023$	$0.275 {\pm} 0.028$	$0.311 {\pm} 0.023$	$0.325 {\pm} 0.005$
c=11	ARI ↑	0.142 ± 0.007	$0.122 {\pm} 0.014$	$0.106 {\pm} 0.013$	$0.009 {\pm} 0.001$	$0.103 {\pm} 0.018$	$0.100 {\pm} 0.018$	$0.143 {\pm} 0.023$	$0.147 {\pm} 0.001$
	NMI ↑	0.306 ± 0.006	0.278 ± 0.031	$0.265 {\pm} 0.012$	$0.243 {\pm} 0.025$	0.276 ± 0.039	$0.238 {\pm} 0.027$	0.325 ± 0.020	0.339 ±0.006
	PC ↑	0.420 ± 0.001	$0.521 {\pm} 0.001$	0.522 ± 0.033	$0.512 {\pm} 0.053$	$0.514{\pm}0.028$	$0.544{\pm}0.001$	$0.447 {\pm} 0.001$	$0.503 {\pm} 0.001$
Seismic	DBI↓	10.9 ± 0.1	$2.834{\pm}0.001$	3.055 ± 1.179	$2.569 {\pm} 0.490$	$3.186{\pm}1.057$	$2.008{\pm}0.001$	$7.034{\pm}0.001$	$3.364{\pm}0.005$
n=78823	XB↓	2.220 ± 0.001	$0.566 {\pm} 0.001$	$0.724 {\pm} 0.356$	$0.474{\pm}0.027$	$0.734{\pm}0.336$	$0.428{\pm}0.001$	$1.412 {\pm} 0.001$	$0.680 {\pm} 0.001$
<i>p</i> =30	$F^* \uparrow$	$0.448 {\pm} 0.001$	$0.460{\pm}0.001$	$0.493 {\pm} 0.006$	$0.492{\pm}0.011$	0.503 ±0.006	$0.491 {\pm} 0.001$	$0.451 {\pm} 0.001$	$0.471 {\pm} 0.001$
c=3	ARI ↑	$0.038 {\pm} 0.001$	$0.040{\pm}0.001$	0.069 ±0.001	$0.046{\pm}0.004$	$0.055 {\pm} 0.022$	$0.063 {\pm} 0.001$	$0.038 {\pm} 0.001$	$0.045 {\pm} 0.001$
	NMI↑	0.043 ± 0.001	$0.045 {\pm} 0.001$	$0.066 {\pm} 0.003$	$0.065 {\pm} 0.005$	0.081±0.024	$0.058 {\pm} 0.001$	$0.043 {\pm} 0.001$	$0.049 {\pm} 0.001$

¹ The superscript ' \uparrow ' sign of evaluation metrics represents that the larger evaluation metrics, the better the clustering performance. The superscript ' \downarrow ' sign of evaluation metrics represents that the smaller evaluation metrics, the better the clustering performance.





Fig. 7. Plot of the corresponding lteration and Time of the different algorithms on ten real-world data sets.

sets, which is based on the new affinity filtering (7) and membership scaling (8) schemes. The acquisition and elimination of the redundant contributions of the samples in the update of the centers is the key to the success of AMFCM.

6.3.2 Statistical Comparisons by Friedman Test

In order to compare the multiple algorithms systematically, the Friedman test [49] is applied to compare the clustering efficiency (Time and Iteration) and quality (F* and **ARI**) of the eight algorithms on the selected ten data sets. In detail, Friedman test at significance level $\alpha = 0.05$ rejects the null hypothesis of equal performance, which leads to the use of post-hoc tests to find out which algorithms are actually different. Next, Nemenyi test is used to where the performance of two algorithms is significantly different if their average ranks over all datasets differ by at least one critical difference. The critical difference is defined as $CD = q_{\alpha} \sqrt{\frac{K(K+1)}{6N}}$, where critical values q_{α} are based on the Studentized range statistic divided by $\sqrt{2}$, K is the number of the comparison algorithms, and N is the number of the data sets. In this part, F^\ast and \boldsymbol{ARI} are selected to evaluate the clustering quality, and the remaining metric have the similar results. The critical difference (CD) diagrams, as shown in Fig. 8, are presented to analyze the significance between AMFCM and the comparison algorithms on the ten data sets with F*, ARI, Iteration and Time, where the average rank of each algorithm is marked on the line and the axis. The axis is turned so that the lowest (best) ranks are to the right. Groups of algorithms that are not significantly different according to Nemenyi test are connected with a red line. The critical difference (CD = 3.3203 at 0.05 significance level) is also shown above the axis in each subfigure.

According to the CD diagrams, first, in the clustering efficiency and quality, AMFCM achieves the statistically superior performance than that of FCM on the ten data sets. Second, from the Fig. 8a and 8b, AMFCM presents statistically comparable clustering quality with ARFCM on the ten data sets. However, AMFCM statistically outperforms ARFCM in the clustering efficiency, as shown in Fig. 8c and 8d. Finally, it can be found that none of the algorithms can present statistically comparable performance with AMFCM in both efficiency and quality. Therefore, AMFCM statistically achieves a good trade-off between the clustering of quality and efficiency.



Fig. 8. CD diagrams of the eight comparison algorithms on the ten data sets with **F***, **ARI**, **Iteration**, and **Time**. It is clear that AMFCM statistically achieves a good trade-off between the clustering quality and efficiency.

6.3.3 Efficiency of the New Affinity Filtering Scheme

This set of experiments, carried out on the same data sets, is presented to test the filtering rate of the new affinity filtering (7), which is the key factor that determines the acceleration of AMFCM. The previous affinity filtering (5) and the new affinity filtering (7) are chosen with the same settings for comparison. Let $\hat{n}_t = |\{1 \le j \le n \mid |\mathcal{P}_j^{(t)}| \ne 0\}|$ denote the number of the samples that satisfy the affinity



Fig. 9. Plots of $\frac{\hat{n}_t}{n}$ in relation to iteration on ten data sets. The log-scale of the x-axis clearly shows the differences of the sample filtering efficiency in [A] stage of the convergence process of the algorithms.

filtering in *t* iteration. For ten data sets, the curves of $\frac{\hat{n}_t}{n}$ are plotted in relation to iteration *t* in Fig. 9, where the blue and red lines represent MSFCM and AMFCM, respectively. And the log-scale of the x-axis clearly shows the differences of the sample filtering efficiency in stage [A] of the convergence process of the algorithms.

Firstly, the filtering efficiency of the new affinity filtering (7) is higher than that of the previous affinity filtering (5) in stage [A]. Moreover, the new affinity filtering (7) has reached the highest filtering rate before 10 iterations, except for Magic and Seismic. Thus, the new affinity filtering (7) overcomes the inherent shortcomings of the previous affinity filtering (5) that is easy to be invalid in stage [A], as shown in Fig. 9.

Secondly, for data sets Magic and Seismic, the efficiency of the previous affinity filtering (5) is higher than that of AMFCM in stage [B] of the convergence process of the algorithms. However, the previous membership scaling (6) is not very effective in accelerating the algorithm in stage [B], which is because $\beta_j^{(t)}$ is very close to 1 in stage [B]. The redundant contributions of the samples in the update of the centers have not been reduced in MSFCM, resulting in a decrease in its clustering efficiency.

AMFCM always maintains a high-efficiency level for all data sets, which can be observed from the number of iteration on the x-axis. According to the above analysis, it can be seen that the new affinity filtering (7) and membership scaling (8) schemes are complementary to each other. Therefore, AMFCM is very efficient in stages [A] and [B] of the convergence process of the algorithms.

7 CONCLUSION

In this paper, FCM based on new affinity filtering and membership scaling (AMFCM) is proposed to accelerate the whole convergence stages of the traditional FCM clustering. In the proposed algorithm, a new affinity filtering is designed to obtain the complete non-affinity centers for each sample by a new set of triangle inequalities, which is more compatible with fuzzy clustering. Then, a new membership scaling is suggested to eliminate the contributions of the samples in the update of their non-affinity centers by setting the membership grades to 0 and promote the contributions of the samples in the update of the remaining centers by the fuzzy membership grades, which improves the performance and efficiency of the algorithm. Many experimental results have verified its effectiveness and efficiency on synthetic and real-world data sets. Therefore, AMFCM is a wellbalanced FCM-type algorithm in clustering efficiency and quality. For future work, AMFCM could be explored to enhance the performance of the FCM-type clustering algorithms on high-dimensional data sets. Another interesting possibility is to generalize the concept to nonlinear fuzzy clustering according to information granules.

REFERENCES

- M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.
- [2] C. Döring, M.-J. Lesot, and R. Kruse, "Data analysis with fuzzy clustering methods," *Computational Statistics & Data Analysis*, vol. 51, no. 1, pp. 192–214, 2006.
- [3] J. C. Bezdek, Pattern recognition with fuzzy objective function algorithms. Springer Science & Business Media, 2013.
- [4] M. R. Rezaee, P. M. J. van der Zwet, B. P. E. Lelieveldt, R. J. van der Geest, and J. H. C. Reiber, "A multiresolution image segmentation technique based on pyramidal segmentation and fuzzy clustering," *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1238–1248, 2000.
- [5] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, M. J. Er, W. Ding, and C.-T. Lin, "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–681, 2017.
- [6] A. K. Jain, "Data clustering: 50 years beyond k-means," Pattern recognition letters, vol. 31, no. 8, pp. 651–666, 2010.
- [7] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions* on *Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [8] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.
 [9] Z. Bian, F.-L. Chung, and S. Wang, "Fuzzy density peaks cluster-
- [9] Z. Bian, F.-L. Chung, and S. Wang, "Fuzzy density peaks clustering," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 7, pp. 1725– 1738, 2021.
- [10] C. Elkan, "Using the triangle inequality to accelerate k-means," in Proceedings of the 20th International Conference on Machine Learning (ICML-03), 2003, pp. 147–153.
- [11] Y. Ding, Y. Zhao, X. Shen, M. Musuvathi, and T. Mytkowicz, "Yinyang k-means: A drop-in replacement of the classic k-means with consistent speedup," in *International Conference on Machine Learning*, 2015, pp. 579–587.
- [12] P. Lingras and C. West, "Interval set clustering of web users with rough k-means," *Journal of Intelligent Information Systems*, vol. 23, no. 1, pp. 5–16, 2004.
- [13] K. Zhang, "A three-way c-means algorithm," Applied Soft Computing, vol. 82, p. 105536, 2019.

- [14] J. Han, J. Xu, F. Nie, and X. Li, "Multi-view k-means clustering with adaptive sparse memberships and weight allocation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 2, pp. 816–827, 2022.
- [15] F. Nie, Z. Li, R. Wang, and X. Li, "An effective and efficient algorithm for k-means clustering with new formulation," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2022.
- [16] S. Xia, D. Peng, D. Meng, C. Zhang, G. Wang, E. Giem, W. Wei, and Z. Chen, "A fast adaptive k-means with no bounds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.
- [17] L. F. S. Coletta, L. Vendramin, E. R. Hruschka, R. J. G. B. Campello, and W. Pedrycz, "Collaborative fuzzy clustering algorithms: Some refinements and design guidelines," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 3, pp. 444–462, 2012.
- [18] J. Xu, J. Han, M. Xu, F. Nie, and X. Li, "Robust and efficient fuzzy c-means clustering constrained on flexible sparsity," arXiv preprint arXiv:1908.06699, 2019.
- [19] S. Zhou, D. Li, Z. Zhang, and R. Ping, "A new membership scaling fuzzy c-means clustering algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 9, pp. 2810–2818, 2021.
- [20] R. J. Hathaway and J. C. Bezdek, "Recent convergence results for the fuzzy c-means clustering algorithms," *Journal of Classification*, vol. 5, no. 2, pp. 237–247, 1988.
- [21] J. Kieffer, "Exponential rate of convergence for lloyd's method i," IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 205–210, 1982.
- [22] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms," SIAM review, vol. 41, no. 4, pp. 637–676, 1999.
- [23] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu, "The analysis of a simple k-means clustering algorithm," in *Proceedings of the sixteenth annual symposium on Computational geometry*, 2000, pp. 100–109.
- [24] S. Mitra, H. Banka, and W. Pedrycz, "Rough-fuzzy collaborative clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 4, pp. 795–805, 2006.
- [25] S. Roy and P. Maji, "Medical image segmentation by partitioning spatially constrained fuzzy approximation spaces," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 5, pp. 965–977, 2020.
- [26] J. Zhou, W. Pedrycz, and D. Miao, "Shadowed sets in the characterization of rough-fuzzy clustering," *Pattern Recognition*, vol. 44, no. 8, pp. 1738–1749, 2011.
- [27] T. William-West, A. F. D. Kana, and M. A. Ibrahim, "Shadowedset-based three-way clustering methods: An investigation of new optimization-based principles," *Information Sciences*, vol. 591, pp. 1–24, 2022.
- [28] F. Zhao, L. Cao, H. Liu, Z. Tang, and J. Fan, "Particle competitive mechanism based multiobjective rough clustering algorithm for image segmentation," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 10, pp. 4127–4141, 2022.
- [29] K. Zhang, J. Dai, and Z. Xu, "The criterion-oriented three-way ranking and clustering strategies in fuzzy decision environments," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 9, pp. 3841–3856, 2022.
- [30] D. Li, H. Zhang, T. Li, A. Bouras, X. Yu, and T. Wang, "Hybrid missing value imputation algorithms using fuzzy c-means and vaguely quantified rough set," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 5, pp. 1396–1408, 2022.
- [31] M. Du, J. Zhao, J. Sun, and Y. Dong, "M3w: Multistep threeway clustering," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2022.
- [32] J. Zhou, Z. Lai, C. Gao, X. Yue, and W. Wong, "Rough-fuzzy clustering based on two-stage three-way approximations," *IEEE Access*, vol. 6, pp. 27541–27554, 2018.
- [33] X. Zhao, F. Nie, R. Wang, and X. Li, "Robust fuzzy k-means clustering with shrunk patterns learning," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, 2021.
- [34] F. Nie, X. Zhao, R. Wang, X. Li, and Z. Li, "Fuzzy k-means clustering with discriminative embedding," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 3, pp. 1221–1230, 2022.
- [35] C. Liu, F. Nie, R. Wang, and X. Li, "Scalable fuzzy clustering with anchor graph," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–11, 2022.
- [36] X. Gu, Q. Ni, and G. Tang, "A novel data-driven approach to autonomous fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 6, pp. 2073–2085, 2022.

- [37] Z. Wang, S.-S. Wang, L. Bai, W.-S. Wang, and Y.-H. Shao, "Semisupervised fuzzy clustering with fuzzy pairwise constraints," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 9, pp. 3797–3811, 2022.
- [38] A. Bargiela and W. Pedrycz, "Toward a theory of granular computing for human-centered information processing," *IEEE Transactions on Fuzzy Systems*, vol. 16, no. 2, pp. 320–330, 2008.
- [39] J. T. Yao, A. V. Vasilakos, and W. Pedrycz, "Granular computing: Perspectives and challenges," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1977–1989, 2013.
- [40] J. F. Kolen and T. Hutcheson, "Reducing the time complexity of the fuzzy c-means algorithm," *IEEE Transactions on Fuzzy Systems*, vol. 10, no. 2, pp. 263–267, 2002.
- [41] A. Bhattacherjee, S. Sanyal, and A. Abraham, "Reducing time complexity of fuzzy c means algorithm," in *Innovations in Bio-Inspired Computing and Applications: Proceedings of the 12th International Conference on Innovations in Bio-Inspired Computing and Applications* (IBICA 2021) Held During December 16–18, 2021. Springer, 2022, pp. 332–347.
- pp. 332–347.
 [42] J. K. Parker and L. O. Hall, "Accelerating fuzzy-c means using an estimated subsample size," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 5, pp. 1229–1244, 2013.
- [43] J.-P. Mei, Y. Wang, L. Chen, and C. Miao, "Large scale document categorization with fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 25, no. 5, pp. 1239–1251, 2016.
- [44] L. Hubert and P. Arabie, "Comparing partitions," Journal of Classification, vol. 2, no. 1, pp. 193–218, 1985.
- [45] J. C. Bezdek, "Cluster validity with fuzzy sets," Journal of Cybernetics, vol. 3, no. 3, pp. 58–73, 1973.
- [46] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.
- [47] X. L. Xie and G. Beni, "A validity measure for fuzzy clustering," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 13, no. 8, pp. 841–847, 1991.
- [48] F. Nie, J. Xue, D. Wu, R. Wang, H. Li, and X. Li, "Coordinate descent method for k-means," *IEEE Transactions on Pattern Analysis* and Machine Intelligence, pp. 1–1, 2021.
- [49] J. Demiar and D. Schuurmans, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1–30, 2006.

APPENDIX A

A GEOMETRIC INTERPRETATION FOR THE NEW AFFINITY FILTERING SCHEME (7)



Fig. 10. The affinities between sample \mathbf{x}_j and all centers are identified by the new affinity filtering scheme (7). Let \mathbf{v}_1 be the nearest center of \mathbf{x}_j . The radius of the green circle and orange arc are the drift of \mathbf{v}_i and the lower bound of \mathbf{v}_i after one update, respectively. The radius of the black circle is the upper bound of \mathbf{v}_1 after one update. Here, the blue and red centers are the non-affinity and the remaining centers of \mathbf{x}_j , respectively.

Here, a geometric interpretation is given for the new affinity filtering scheme (7), as illustrated in Fig. 10. In Fig. 10, there is any sample \mathbf{x}_j and the cluster centers are $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_5]$. Let \mathbf{v}_1 be the nearest center of \mathbf{x}_j . The radius of green circle represent δ_i , the radius of orange arc represent the lower bound of \mathbf{v}_i , $d_{ij} - \delta_i$, and the radius of black circle is the upper bound of \mathbf{v}_1 , $d_{1,j} + \delta_1$. In Fig. 10, as long as the orange arc of \mathbf{v}_i is outside the black circle, it means that \mathbf{v}_i cannot be the nearest center of \mathbf{x}_j . Thus, \mathbf{v}_i is identified as the non-affinity center of \mathbf{x}_j according to Lemma 2. For example, \mathbf{v}_4 and \mathbf{v}_5 are the non-affinity centers of \mathbf{x}_j , which are marked in blue. The red centers are the remaining centers of \mathbf{x}_j . In this case, the set of the non-affinity centers of \mathbf{x}_j obtained by (7) is complete.

APPENDIX B PROOF OF THEOREM 1

As analyzed in Subsection 3.2, in the stage [B] of the convergence process, the assignment of most samples remains unchanged with the slight changes of the centers, where $\tilde{\mathbf{X}}^{(t)}$ is defined as the set of the samples whose the assignment does not change in t and t + 1 iteration. Therefore, we have: $\tilde{u}_{.j}^{(t)} = \tilde{u}_{.j}^{(t+1)}$ for $\mathbf{x}_j \in \tilde{\mathbf{X}}^{(t)}$ based on the Eq. (7) and (8). However, in FCM, we have: $u_{.j}^{(t)} \neq u_{.j}^{(t+1)}$ for $\mathbf{x}_j \in \tilde{\mathbf{X}}^{(t)}$ based on the Eq. (4). According to the membership grade matrix U and $\tilde{\mathbf{U}}$ of FCM and AMFCM, we have:

$$\|\tilde{\mathbf{U}}^{(t+1)} - \tilde{\mathbf{U}}^{(t)}\| < \|\mathbf{U}^{(t+1)} - \mathbf{U}^{(t)}\|.$$
(12)

Therefore, given the same termination parameter ε , the number of the iteration of AMFCM, t_{AMFCM} , is smaller than that of FCM, t_{FCM} .

APPENDIX C PROOF OF THEOREM 2

First, we list the objective function of FCM and AMFCM with the same initialization $\mathbf{V}^{(t)}$ from the *t* iteration to the t + 1 iteration:

$$J_{\text{FCM}}^{(t)} = \sum_{j=1}^{n} \sum_{i=1}^{c} (u_{\text{FCM}_{ij}}^{(t)})^{m} \|\mathbf{x}_{j} - \mathbf{v}_{i}^{(t)}\|^{2},$$

$$J_{\text{FCM}}^{(t+1)} = \sum_{j=1}^{n} \sum_{i=1}^{c} (u_{\text{FCM}_{ij}}^{(t)})^{m} \|\mathbf{x}_{j} - \mathbf{v}_{\text{FCM}_{i}}^{(t+1)}\|^{2},$$

$$J_{\text{AMFCM}}^{(t)} = \sum_{j=1}^{n} \sum_{i=1}^{c} (\tilde{u}_{ij}^{(t)})^{m} \|\mathbf{x}_{j} - \mathbf{v}_{i}^{(t)}\|^{2},$$

$$J_{\text{AMFCM}}^{(t+1)} = \sum_{j=1}^{n} \sum_{i=1}^{c} (\tilde{u}_{ij}^{(t)})^{m} \|\mathbf{x}_{j} - \mathbf{v}_{\text{AMFCM}_{i}}^{(t+1)}\|^{2}.$$
(13)

We assume that there must exist an iteration threshold T, when t < T, $\mathcal{P}_{j}^{(t)} = \emptyset$ for j = 1, 2, ..., n. Obviously, when t < T, $J_{\text{FCM}} = J_{\text{AMFCM}}$. When $t \ge T$, $\exists j \in \{1, 2, ..., n\}$, satisfying $\mathcal{P}_{j}^{(t)} \neq \emptyset$. Then, we derive lower and upper bounds of J_{AMFCM} .

For the lower bound of $J_{AMFCM}^{(t)}$, based on the Lagrangian multiplier method, we have: $J_{FCM}^{(t)} < J_{AMFCM}^{(t)}$. For the upper bound of $J_{AMFCM}^{(t)}$, we have:

$$J_{\text{AMFCM}}^{(t)} = \sum_{j=1}^{n} \sum_{i=1}^{c} (\tilde{u}_{ij}^{(t)})^{m} \|\mathbf{x}_{j} - \mathbf{v}_{i}^{(t)}\|^{2},$$

$$= \sum_{j=1}^{n} [\alpha_{j}^{(t)}]^{m} \cdot [\sum_{i \notin \mathcal{P}_{j}^{(t)}} (u_{\text{FCM}_{ij}^{(t)}})^{m} \|\mathbf{x}_{j} - \mathbf{v}_{i}^{(t)}\|^{2}],$$

$$< [\max_{1 \le j \le n} \{\alpha_{j}^{(t)}\}]^{m} \cdot [\sum_{j=1}^{n} \sum_{i=1}^{c} (u_{\text{FCM}_{ij}^{(t)}})^{m} \|\mathbf{x}_{j} - \mathbf{v}_{i}^{(t)}\|^{2}],$$

$$= [\max_{1 \le j \le n} \{\alpha_{j}^{(t)}\}]^{m} \cdot J_{\text{FCM}}^{(t)},$$
(14)

where $\tilde{\mathbf{U}}^{(t)}$ can be rewritten as $\alpha_j^{(t)} \cdot u_{\text{FCM}}_{ij}^{(t)}$ for $i \notin \mathcal{P}_j^{(t)}$, and 0 for $i \in \mathcal{P}_j^{(t)}$. Based on the Eq. (7) and (8), $1 < \alpha_j^{(t)} = \frac{1}{1 - \sum_{i \in \mathcal{P}_j^{(t)}} u_{\text{FCM}}_{ij}^{(t)}} < 2$. Therefore, we have:

$$\int J_{\text{FCM}}^{(t)} < J_{\text{AMFCM}}^{(t)} < [\max_{1 \le j \le n} \{\alpha_j^{(t)}\}]^m \cdot J_{\text{FCM}}^{(t)}.$$
 (15)

For the upper bound of $J_{AMFCM}^{(t+1)}$, based on the Lagrangian multiplier method, we have:

$$J_{AMFCM}^{(t+1)} = \sum_{j=1}^{n} \sum_{i=1}^{c} (\tilde{u}_{ij}^{(t)})^{m} \|\mathbf{x}_{j} - \mathbf{v}_{AMFCM_{i}}^{(t+1)}\|^{2},$$

$$< \sum_{j=1}^{n} \sum_{i=1}^{c} (\tilde{u}_{ij}^{(t)})^{m} \|\mathbf{x}_{j} - \mathbf{v}_{FCM_{i}}^{(t+1)}\|^{2}.$$
 (16)

Based on a similar derivation from the Eq. (14), we have:

$$J_{\text{AMFCM}}^{(t+1)} < [\max_{1 \le j \le n} \{\alpha_j^{(t)}\}]^m \cdot J_{\text{FCM}}^{(t+1)}.$$
 (17)

Finally, based on the Eq. (15) and (17), we have:

$$\frac{J_{\text{AMFCM}}^{(t)}}{J_{\text{AMFCM}}^{(t+1)}} > \frac{J_{\text{FCM}}^{(t)}}{[\max_{1 \le j \le n} \{\alpha_j^{(t)}\}]^m \cdot J_{\text{FCM}}^{(t+1)}}.$$
 (18)

Here, we set the decline ratio of the objective function of FCM as $\lambda^{(t)} = \frac{J_{\text{FCM}}^{(t)}}{J_{\text{FCM}}^{(t+1)}}$. As analyzed in [22], the convergence rate of the alternating optimization algorithm (AO) drops with the iteration. Therefore, $\lambda^{(t)}$ is monotonically decreasing with the iteration, i.e., $\lambda^{(t)} \ge 1$ and $\lim_{t \to +\infty} \lambda^{(t)} = 1$. Obviously, in the mid stage, when $\lambda^{(t)} > [\max_{1 \le j \le n} {\alpha_j^{(t)}}]^m$, we have:

$$\frac{J_{\text{AMFCM}}^{(t)}}{J_{\text{AMFCM}}^{(t+1)}} > 1.$$
(19)

In conclusion, AMFCM does not converge precociously in the mid stage of the clustering process.

APPENDIX D PROOF OF THEOREM 3

First, let $\hat{\mathbf{X}}^{(t)}$ be the set of samples \mathbf{x}_j with $|\mathcal{P}_j^{(t)}| = 1$ in t iteration, where $j = 1, 2, ..., |\hat{\mathbf{X}}^{(t)}|$. Given the same initialization $\mathbf{V}^{(t)}$ on $\hat{\mathbf{X}}^{(t)}$, obviously, $J_{\text{Hard}}^{\text{AMFCM}(t+1)} = J_{\text{Hard}}^{\text{FCM}(t+1)}$. Next, after one iteration of centers $\mathbf{V}^{(t)}$ through FCM and AMFCM respectively, based on the Lagrangian multiplier method, we have:

$$J_{AMFCM}^{(t+1)} = \sum_{j=1}^{|\hat{\mathbf{X}}^{(t)}|} \sum_{i=1}^{c} (\tilde{u}_{ij}^{(t)})^m \|x_j - \mathbf{v}_{AMFCM_i}^{(t+1)}\|^2 < \sum_{j=1}^{|\hat{\mathbf{X}}^{(t)}|} \sum_{i=1}^{c} (\tilde{u}_{ij}^{(t)})^m \|x_j - \mathbf{v}_{FCM_i}^{(t+1)}\|^2.$$
(20)

Because $|\mathcal{P}_{j}^{(t)}| = 1$ for any $\mathbf{x}_{j} \in \hat{\mathbf{X}}^{(t)}$, we have that $\tilde{u}_{I_{j}^{*}j}^{(t)} = 1$, and $\tilde{u}_{ij}^{(t)} = 0$ for $i \neq I_{j}^{*}$. Therefore, we have:

$$J_{\text{Hard}}^{\text{AMFCM}(t+1)} = J_{\text{AMFCM}}^{(t+1)} < J_{\text{Hard}}^{\text{FCM}(t+1)}.$$
 (21)

In conclusion, for the samples \mathbf{x}_j with $|\mathcal{P}_j^{(t)}| = 1$, the corresponding hard objective value of AMFCM is smaller than that of FCM in *t* iteration.