New Membership Scaling Fuzzy C-Means Clustering Algorithm

Shuisheng Zhou, Dong Li, Zhuan Zhang, and Rui Ping

***Abstract*—Fuzzy c-means (FCM) is one of the most frequently used methods for clustering. However, with the increasing amount of data, FCM suffers from a slow convergence and a large amount of calculation, since all samples are involved in updating the solutions per iteration without considering the current clustering results. In this paper, based on an observation that the samples, whose nearest cluster center is v, will help the convergence of v and the rest samples will prevent the convergence of v, we propose a new membership scaling FCM (MSFCM). In the new algorithm, many samples which will not change their closeness relationships in next iteration are chosen by the triangle inequality; then a new scheme for scaling the membership degrees of the chosen samples is suggested to boost the effect of the in-cluster samples and weaken the effect of the out-of-cluster samples in clustering process. The new scheme not only improves the convergence of the algorithm but also keeps the high quality of the clustering. Many experimental results on synthetic and real data sets verify the effectiveness of the proposed algorithm for improving the convergence of the fuzzy clustering. Especially, comparing to FCM, MSFCM saves about 1/3 iterations without significant increasing the cost per iteration.**

***Index Terms*—FCM, Triangular inequality, membership degree, membership scaling.**

I. Introduction

DATA clustering is an important topic of machine learning. The clustering algorithm is unsupervised learning method, whose aim is to divide a data set of physical or abstract objects into similar groups by the similarity measures. Cluster analysis is widely used in many fields, including data mining, pattern recognition, image processing [1], [2], [3], [4], [5] etc.

The classic c-means algorithm (Lloyds algorithm) [6], which is a common method of data clustering, consists of two steps: for an input of n data points and c initial cluster centers, the assignment step assigns each point to its closest cluster, and the update step renews each of the c cluster centers with the centroid of the points assigned to that cluster. The algorithm repeats itself until convergence. We do not plan to review more researches on the c-means further. But the triangle inequality for clustering, first in [7] and recently extend in [8], is a very efficient tool to avoid unnecessary distance calculations and leverage the performance of clustering algorithms. In this brief, we firstly equip it with the fuzzy clustering, which will be introduced next.

S. Zhou, D. Li, Z. Zhang, and R. Ping are with School of Mathematics and Statistics, Xidian University, Xi'an, P. R. China. e-mail: sszhou@mail.xidian.edu.cn.

This work was supported by the National Natural Science Foundation of China under Grants No. 61772020.

Manuscript received xxxx, 2019; revised xxxx, 2019.

The fuzzy clustering [9] is a relatively new trend for data clustering, in which a sample item does not exclusively belong to a single cluster, but is a member of every cluster with a membership degree in $[0, 1]$. A well-known example of fuzzy clustering algorithms is the fuzzy c-means (FCM) in [10]. Due to its flexibility and robustness for ambiguity, FCM has been widely used (see[1], [6], [9] and the references therein).

However, FCM suffers from the slow convergence since all samples have effects on all cluster centers all the time.

There are many researches on those issues. For example, fuzzy c-means clustering based on weights and gene expression programming (WGFCM) [11] was proposed by introducing a weight vectors based on entropy and an update of cluster determined by the gene expression programming. Hathaway and Hu [12] designed the density-weighted fuzzy c-means (DWFCM) to improve the convergence by reducing the larger data set to a weighted smaller one. Geometric progressive fuzzy c-means (GOFCM) and minimum sample estimate random fuzzy c-means (MSERFCM) in [13] accelerated FCM by progressive sampling and random sampling ways respectively. An FCM algorithm based on morphological reconstruction and membership filtering (FRFCM) [14] was proposed to improve the speediness and robustness.

Some variant FCM algorithms are aimed to large scale clustering problem. Such as,the gradient based fuzzy c-means (GBFCM) [15] utilized the gradient descent to improve speed and stability of convergence. Recently, the stochastic gradient descent based fuzzy clustering (SGFCM) [16] with the mini-batch scheme was recommended to improve the clustering efficiency for large-scale data sets. Havens, Bezdek *et al.* have presented LFCM and rseFCM algorithms in [17] for very large data, including nonlinear clustering by kernel trick, to decrease the clustering time with many types of relaxing of the convergence conditions. Khoshkbarchi *et al.* [18] also suggested a modified hybrid fuzzy clustering method (MHFCM) for big data by the map-reduce technique and particle swarm optimization. In [19], FCM was improved to deal with both large scale and high dimensionality document categorization.

In this paper, we propose an improved fuzzy c-means called the membership scaling FCM (MSFCM) which is originally based on the triangle inequality and a new perspective of membership to scale the membership degree. Compared to some of FCM algorithms listed above, the new MSFCM has the properties with less iterations, fast convergence, lower time consumption, and the high quality of cluster. It also has the potential possibility for very large scale clustering problem.

The remaining parts of this paper are organized as follows: Section II gives a review and some new findings of FCM algorithm. In Section III, after introducing the triangle inequality in clustering with a new geometric interpretation, we propose a new membership degree scaling scheme, and hence give a new clustering algorithm. The experimental results with discussion are listed in Section IV and Section V concludes the paper.

II. Review and some new findings of FCM

Given the data set be $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$ with $\mathbf{x}_j \in \mathbb{R}^p$ being the j-th sample and the target clusters number c, FCM clustering partitions the $\mathbf{X}$ into c clusters by the cluster centers $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_c] \in \mathbb{R}^{p\times c}$ and the membership degree matrix $\mathbf{U} = [u_{ij}] \in \mathbb{R}^{c\times n}$. For a given fuzziness weighting exponent $m > 1$, $\mathbf{V}$ and $\mathbf{U}$ are solved iteratively according to the following optimization problem

$$\min_{\mathbf{U},\mathbf{V}} J(\mathbf{U},\mathbf{V}) = \sum_{i=1}^{c}\sum_{j=1}^{n} u_{ij}^m \|\mathbf{x}_j - \mathbf{v}_i\|^2, \qquad (1)$$

$$s.t. \sum_{i=1}^{c} u_{ij} = 1, u_{ij} \geq 0.$$

FCM scheme usually initializes $\mathbf{U}^{(0)}$ and updates $\mathbf{V}$ and $\mathbf{U}$ alternatively as

$$\mathbf{v}_i^{(t+1)} = \frac{\sum_{j=1}^{n}\left(u_{ij}^{(t)}\right)^m \mathbf{x}_j}{\sum_{j=1}^{n}\left(u_{ij}^{(t)}\right)^m}, \qquad (2)$$

$$u_{ij}^{(t+1)} = \left[\sum_{k=1}^{c}\left(\frac{\|\mathbf{x}_j - \mathbf{v}_i^{(t+1)}\|}{\|\mathbf{x}_j - \mathbf{v}_k^{(t+1)}\|}\right)^{\frac{2}{m-1}}\right]^{-1}, \qquad (3)$$

until convergence. Next we prove the upper and lower bounds of the membership degree u_{ij} and give it a new interpretation.

A. The bounds of the membership degree

For any sample $\mathbf{x}_j$, the distances between $\mathbf{x}_j$ and the cluster centers $\mathbf{V}$ are $d_{ij}^{(t)} = \|\mathbf{x}_j - \mathbf{v}_i^{(t)}\|$, $i = 1, 2, \cdots, c$. We rearrange them in the ascending order and denote them by $D_j^{(1)}, D_j^{(2)}, \cdots, D_j^{(c)}$, namely $D_j^{(1)} \leq D_j^{(2)} \leq \cdots \leq D_j^{(c)}$. Then we have the following lemma.

Lemma 1. *For the sample $\mathbf{x}_j$, its membership degrees u_{ij} satisfy the following inequalities*

$$0 \leq u_{ij} \leq \left[1 + (c-1)\left(\frac{d_{ij}^{(t)}}{D_j^{(c)}}\right)^{\frac{2}{m-1}}\right]^{-1}, 1 \leq i \leq c. \qquad (4)$$

Especially,

$$\frac{1}{c} \leq u_{I_j^* j} \leq \left[1 + (c-1)\left(\frac{D_j^{(1)}}{D_j^{(c)}}\right)^{\frac{2}{m-1}}\right]^{-1}, \qquad (5)$$

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

Proof: In view of (3) and the notation $D_j^{(k)}$ above, we have

$$u_{ij} = \left[\sum_{k=1}^{c}\left(d_{ij}^{(t)}/D_j^{(k)}\right)^{\frac{2}{m-1}}\right]^{-1}.$$

Inequalities (4) and (5) follow by the definition of $D_j^{(k)}$ and the monotonicity of the involved functions. ■

In Section III, we will use those inequalities to scale the membership degrees of some samples directly to improve the convergence of the clustering and save the computing cost.

B. New interpretation about the membership degree

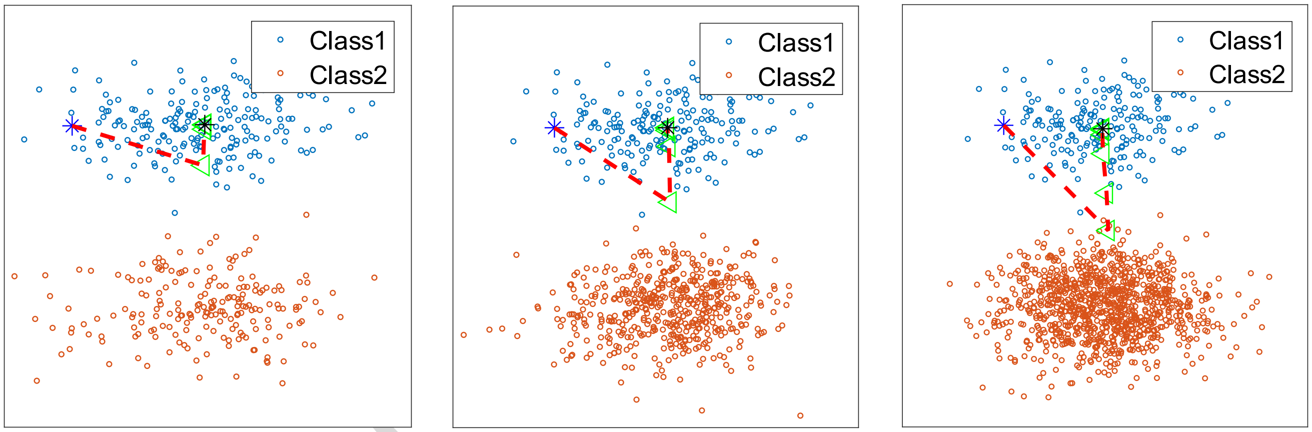
Firstly, we perform a set of experiments to evaluate the effect of FCM on the size of the data set. There is a set of data which contains two simple clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ as in Fig. 1. Both data sets are generalized by two-dimensional Gaussian distribution with a standard deviation of 0.5, where $\mathcal{C}_1$ is fixed with 200 samples, while $\mathcal{C}_2$ is varied with 200, 500, 1000 samples respectively. FCM is used in those three situations with $c = m = 2$ and the same start points. Here we mainly focus on the effect of the size of $\mathcal{C}_2$ to the convergence trajectory of $\mathcal{C}_1$. The experimental results are plotted in Fig. 1.

From Fig. 1, it is clear that the size of $\mathcal{C}_2$ greatly affects the convergence trajectory of $\mathcal{C}_1$. In order to explain this phenomenon, let us have deep explore of the FCM iterations. In view of (2), each sample in the data set has an effect on the updating of any cluster center in ways of the membership degree. In our case, we have $\mathbf{v}_1 = \frac{1}{\sum_j u_{1j}^m}\left(\sum_{j\in\mathcal{C}_1} u_{1j}^m \mathbf{x}_j + \sum_{j\in\mathcal{C}_2} u_{1j}^m \mathbf{x}_j\right)$. It is clear that the samples in $\mathcal{C}_1$ do the "right" thing to attract $\mathbf{v}_1$ approaching to its target, which ensures the convergence of the algorithm. On the contrary, all the samples in $\mathcal{C}_2$ do the "wrong" thing to attract $\mathbf{v}_1$ to departure from its target. Since $u_{1j} \geq u_{1j'}$ for $j \in \mathcal{C}_1, j' \in \mathcal{C}_2$ by (3), if the size of $\mathcal{C}_2$ is small, the "wrong" effect on $\mathbf{v}_1$ by $\mathcal{C}_2$ is less as Fig. 1(a) shows. While the size of $\mathcal{C}_2$ becomes large, the total "wrong" effect is very significant as Fig. 1(c) shows. Namely, the samples in $\mathcal{C}_1$ let $\mathbf{v}_1$ close to its target, while the samples in $\mathcal{C}_2$ prevent $\mathbf{v}_1$ close to its target. This explains the finding of Fig. 1. Hence, we conclude that the samples in $\mathcal{C}_2$ delay the convergence of $\mathbf{v}_1$. And the similar analysis on $\mathbf{v}_2$ will reveal that the samples in $\mathcal{C}_1$ also delay the convergence of $\mathbf{v}_2$.

For $c > 2$, the analysis will be similar. That is to say, the update of $\mathbf{v}$ can be split into two aspects. In the first aspect, the samples, whose nearest center is $\mathbf{v}$, help the convergence of $\mathbf{v}$; and in the second aspect, the rest samples, whose nearest center is not $\mathbf{v}$, prevent the convergence of $\mathbf{v}$. This is one of the reasons for the slow convergence of FCM in the case of the large data with multiple clusters. In order to improve the convergence of FCM, we can manage to enhance the update in the first aspect, and to weaken it in the second aspect.

To accomplish this, a initial thought is to increase the membership degree u_{ij} if $\mathbf{v}_i$ is the nearest center of $\mathbf{x}_j$, and decrease u_{ij} if $\mathbf{v}_i$ is not the nearest center of $\mathbf{x}_j$. However, there are two difficulties need to overcome.

The first one is how to choose the samples to increase or decrease their membership degrees, since we do not know the final clusters. $\mathbf{v}_i$ is the nearest center of $\mathbf{x}_j$ in current iteration, but it is not always to be the nearest center of $\mathbf{x}_j$ in next iteration. The exception is that, if $\mathbf{v}_i$ has been the nearest center of $\mathbf{x}_j$ many times, then it has a high probability



(a) $|\mathcal{C}_2|$=200 (b) $|\mathcal{C}_2|$=500 (c) $|\mathcal{C}_2|$=1000

Fig. 1: The trajectories of cluster centers of $\mathcal{C}_1$ in the FCM iterations. Here, FCM converges within 7, 9 and 11 iterations and costs 0.004, 0.006 and 0.010 seconds corresponding to the case (a), (b), and (c) respectively, where the blue star is the initial point, the green triangle is the cluster center per iteration, and the black star is the target.

keeping the relationship. In Section III, we filter out some samples $\mathbf{x}_j, j \in J$, which will not change their closeness relationships in next iteration by the triangle inequality. For those $\mathbf{x}_j$, we increase its membership degree corresponding to the closest cluster, and simultaneously decrease the other membership degrees. Many experimental results support that this technique can improve the convergence of FCM.

The other difficult is how much the increment/decrement of the chosen membership degree is. Since $\sum_{i=1}^{c} u_{ij} = 1$, we should only focus on the increment of the membership degree corresponding to the nearest cluster $\mathbf{v}_{I_j^*}$. If the increment is too small, the improvement will be insignificant; while the increment is too large, such as set $u_{I_j^* j}$ be 1 or very near 1, the algorithm is degenerated into hard c-means. In this work, in view of the new results in Lemma 1, we increase $u_{I_j^* j}$ to its upper bounds as it is the true FCM membership degree.

Those two issues are the motivation for us to propose a new FCM algorithm in Section III.

III. New membership scaling FCM clustering

Some state-of-the-art researches [7], [8] on c-means (or called k-means) show that the triangle inequality can speedup the clustering algorithms by avoiding unnecessary distance calculations. Motivated by the analysis above, we introduce the triangle inequality in FCM clustering and propose a new membership scaling fuzzy c-means algorithm, called MSFCM. As far as we know, the idea and the method are all novel.

A. Triangle inequality in clustering

As they have done in works [7], [8], we can identify whether a sample changes its cluster after an iteration or not with a single comparison by the triangle inequality. For each sample $\mathbf{x}_j$, if the algorithm maintains the distance to the closest cluster center, $D_j^{(1)}$, and the distance to the second-closest cluster center, $D_j^{(2)}$, then we can infer that which kinds of samples will not change their clusters in next iteration by a statement in [8]. We relist it here in our version for convenience.

Lemma 2. *Let $\delta_i = d(\mathbf{v}_i^{(t+1)}, \mathbf{v}_i^{(t)})$ be displacement of the cluster center $\mathbf{v}_i^{(t)}$ $(1 \le i \le c)$. A sample $\mathbf{x}_j$, whose nearest cluster center is $\mathbf{v}_{I_j^*}^{(t)}$ with $I_j^* = \arg\min_{1\le i\le c}\{d_{i,j}^{(t)}\}$, will not change its nearest cluster after another FCM update, if*

$$D_j^{(2)} - \max_{1\le i\le c} \delta_i \ge D_j^{(1)} + \delta_{I_j^*}. \tag{6}$$

Namely, $\arg\min_{1\le i\le c}\{d_{i,j}^{(t+1)}\} = I_j^*$ keeps in this case.

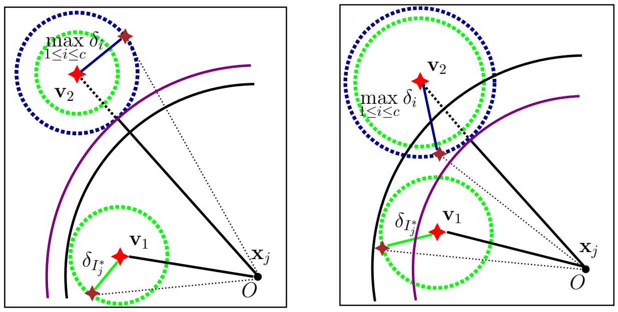
The proof of Lemma 2 is similar in [8]. Here we just give a new geometric explanation as Fig. 2 shows (See the caption of Fig. 2 for details).

In hard k-means algorithm, after the new cluster centers are obtained as $\mathbf{v}_i^{(t)}, i = 1, \cdots, c$, there needs a lot of cost to determine the nearest cluster center of all samples $\mathbf{x}_j, j = 1, \cdots n$. However, by Eq. (6), they [7], [8] filter out many samples, whose nearest cluster will not change according to current $\mathbf{V}^{(t)}$. Hence, much computational cost is saved and the speed of algorithm is improved.

In FCM algorithm, we can also filter out some samples whose nearest cluster will not change in next iteration by Eq. (6). Then with those priori information, we design a scheme to improve the clustering by the analysis in Section II-B. The details will be posted in next subsection.

B. The membership degrees scaling scheme

For FCM, if the current cluster $\mathbf{V}^{(t)}$ is obtained by (2), then we need to compute $\mathbf{U}^{(t+1)}$ by (3). By the analysis in Section II-B, if we know which kinds of samples will not change their cluster, we can use them to improve the convergence of the algorithm. Specifically, let $\mathbf{X}_J = \{\mathbf{x}_j | j \in J\}$ be a subset of the filtered samples by Eq. (6), where $J \subset \{1, \cdots, n\}$ is the subscript set of the filtered samples. Namely, the index of the nearest cluster center of $\mathbf{x}_j$, $j \in J$, will not change when the cluster center matrix is updated from $\mathbf{V}^{(t)}$ to $\tilde{\mathbf{V}}^{(t+1)}$.



(a) (b)

Fig. 2: Geometric explanation of Lemma 2. For $\mathbf{x}_j$, let $\mathbf{v}_1$ be its nearest center and $\mathbf{v}_2$ be its second-nearest center. The radius of the green dotted circle is $\delta_i = d(\mathbf{v}_i^{(t+1)}, \mathbf{v}_i^{(t)})$, and the radius of the darkblue dotted circle is $\max\limits_{1\le i\le c} \delta_i$. Hence the new $\mathbf{v}_i (i \ge 2)$ must be at the out of the purple solid arc, and the new $\mathbf{v}_1$ must be inner the dark solid arc. In case (a), since $|O\mathbf{v}_2| - \max\limits_{1\le i\le c} \delta_i \ge |O\mathbf{v}_1| + \delta_{I_j^*}$, which satisfies Lemma 2, $\mathbf{x}_j$ must not change its nearest cluster in next iteration. In case (b), since $|O\mathbf{v}_2| - \max\limits_{1\le i\le c} \delta_i < |O\mathbf{v}_1| + \delta_{I_j^*}$, we do not know whether the nearest cluster of $\mathbf{x}_j$ changes or not in next iteration.

In this situation, we propose a new algorithm with two different kinds of schemes to update $\mathbf{U}^{(t+1)}$, but not only dependent on (3). One is for the samples in $\mathbf{X}_J$, whose membership degrees $u_{i,j}^{(t+1)}$ are adjusted by a new technique described in next paraph, where much computational cost is saved. The other is for the sample $\mathbf{x}_j, j \notin J$, whose membership degrees are still computed by (3), which is very important for the algorithm to maintain the advantages of FCM. Especially at the very beginning of the algorithm without the well-chosen initial centers, most of the membership degrees are updated by this way.

For the sample $\mathbf{x}_j \in \mathbf{X}_J$, its membership degrees to the cluster centers $\mathbf{V}^{(t)}$ are the vector $\mathbf{u}_j^{(t)} = (u_{1,j}, \cdots, u_{c,j})^\top$. We will obtain $\mathbf{u}_j^{(t+1)}$ by a simple scheme to improve the convergence of the algorithm. Let I_j^* be the index of the nearest cluster center of $\mathbf{x}_j$. Based on the new interpretation on the membership degree in Section II-B, we simply increase $u_{I_j^*,j}^{(t)}$ to $u_{I_j^*,j}^{(t+1)}$ by multiplying a factor α_j larger than 1 (Of course, $u_{I_j^*,j}^{(t+1)} \le 1$ must be kept), and decrease $u_{i,j}^{(t)}$ to $u_{i,j}^{(t+1)}$ for $i \ne I_j^*$ by multiplying a factor β_j less than 1. In view of Lemma 1, we can increase $u_{I_j^*,j}^{(t)}$ to its upper bounds

$$M_j = \left[1 + (c-1)\left(\frac{D_j^{(1)}}{D_j^{(c)}}\right)^{\frac{2}{m-1}}\right]^{-1}. \tag{7}$$

Namely, $u_{I_j^*,j}^{(t+1)} = M_j$ and $\alpha_j = \frac{M_j}{u_{I_j^*,j}^{(t)}}$. Thus, we have

$$\beta_j = \frac{1 - M_j}{1 - u_{I_j^*,j}^{(t)}}. \tag{8}$$

As the result, our new update scheme for $\mathbf{U}^{(t+1)}$ is

$$u_{i,j}^{(t+1)} = \begin{cases} M_j, & j \in J, i = I_j^*, \\ \beta_j u_{i,j}^{(t)}, & j \in J, i \ne I_j^*, \\ u_{i,j}^{(t)}, & j \notin J, \\ & 1 \le i \le c, \end{cases} \tag{9}$$

where $d_{ij}^{(t)} = \|\mathbf{x}_j - \mathbf{v}_i^{(t)}\|$. Notice that for the samples $\mathbf{x}_j$ ($j \notin J$), the distances between the samples and the cluster centers, $d_{ij}^{(t)}$, had already been calculated, and there no needs any extra cost.

C. The proposed algorithm

Now we propose a new membership degree scaling FCM algorithm integrating with traditional one. In new algorithm, after a traditional FCM iteration, we adjust the current $\mathbf{U}$ and $\mathbf{V}$ using the scaling scheme in Subsection III-B. The proposed algorithm, called the **m**embership **s**caling **FCM** (**MSFCM**), is listed as Algorithm 1.

Algorithm 1 MSFCM

Input: Dataset $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, cluster number c;
Output: Membership degree matrix $\mathbf{U}$ and cluster center matrix $\mathbf{V}$.

1: Initialize fuzzy exponent m, convergence threshold ε, compute the cluster center $\mathbf{V}^{(1)}$ by the initial membership degree matrix $\mathbf{U}^{(0)} \in \mathbb{R}^{c\times n}$ according to (2); Set $t := 1$.
2: Compute $d_{i,j}^{(t)} = \|\mathbf{x}_j - \mathbf{v}_i^{(t)}\|$ for $1 \le i \le c, 1 \le j \le n$;
3: Compute $\mathbf{U}^{(t)}$ with $u_{ij}^{(t)} = \left[\sum_{k=1}^{c}\left(\frac{d_{ij}^{(t)}}{d_{kj}^{(t)}}\right)^{\frac{2}{m-1}}\right]^{-1}$;
4: Compute $\tilde{\mathbf{V}}^{(t+1)}$ with $\tilde{\mathbf{v}}_i^{(t+1)} = \frac{\sum_{j=1}^{n}\left(u_{ij}^{(t)}\right)^m \mathbf{x}_j}{\sum_{j=1}^{n}\left(u_{ij}^{(t)}\right)^m}$;
5: Compute $\delta_i (i = 1, \cdots, c)$ using $\mathbf{V}^{(t)}$ and $\tilde{\mathbf{V}}^{(t+1)}$;
6: Filter out the subset $\mathbf{X}_J$ according to (6);
7: Update $\mathbf{U}^{(t+1)}$ with new scheme according to (9);
8: Compute $\mathbf{V}^{(t+1)}$ with $\mathbf{v}_i^{(t+1)} = \frac{\sum\limits_{j=1}^{n}\left(u_{ij}^{(t+1)}\right)^m \mathbf{x}_j}{\sum\limits_{j=1}^{n}\left(u_{ij}^{(t+1)}\right)^m}$;
9: **if** $|\mathbf{X}_J| < n$ and $\|\mathbf{V}^{(t+1)} - \mathbf{V}^{(t)}\| \ge \varepsilon$ **then**
10: Set $t := t + 1$, Goto Step 2;
11: **else**
12: **return** $\mathbf{U} = \mathbf{U}^{(t+1)}$, $\mathbf{V} = \mathbf{V}^{(t+1)}$;
13: **end if**

Now we do some explanations of the new MSFCM:

- The difference between the proposed MSFCM and FCM can be illustrated as the next follow chart:

$$\to \mathbf{V}^{(t)} \xrightarrow{(3)} \mathbf{U}^{(t)} \boxed{\xrightarrow[(6),\ (9)]{(2)\ \tilde{\mathbf{V}}^{(t+1)}} \mathbf{U}^{(t+1)}} \xrightarrow{(2)} \mathbf{V}^{(t+1)} \to .$$

  Compared with the traditional FCM, MSFCM only inserts the boxed part, corresponding to Step 5-7 in Algorithm 1, to update a new $\mathbf{U}^{(t+1)}$ by a novel scheme.
- The complexity of the new inserted Step 5-7 is very low, since $d_{ij}^{(t)}$ (or $D_j^{(1)}$ and $D_j^{(2)}$) in (6) and (9) has been calculated in Step 2 and the cost of computing

$\delta_i (i = 1, \cdots, c)$ is only $O(cp)$. The extra cost of MSFCM over FCM per iteration is on Step 4, whose cost is $O(ncp)$. Compared to FCM, whose cost is $O(2ncp)$ per iteration, the cost of MSFCM is $O(3ncp)$ per iteration. However, the inserted part improves the convergence of the algorithm greatly, and the iteration number of the new algorithm is no more than one third of FCM, which will be illustrated by many experiments in Section IV. Hence, the total cost is saved.

- There exists another possible scheme to implement our scaling idea for FCM. It is neglecting standard FCM update in Step 2 and Step 4, and only uses the process from Step 5-8. This version works very quickly, but it losses some advantages of FCM. Hence, it often has a little lower accuracy. Therefore, the proposed algorithm is good tradeoff between the speed and the accuracy.
- One shortcoming of MSFCM is clear. It cannot maintain the monotone of $J(\mathbf{U}, \mathbf{V})$ in the iteration, since we modified some values derived by the ordinary optimization theory. However, we can prove that the corresponding objective of the hard-clustering, defined as

$$\hat{J}(\mathbf{U}, \mathbf{V}) = \sum_{j=1}^{n} \|\mathbf{x}_j - \mathbf{v}_{I_j^*}\|^2, \ (I_j^* = \max_{1 \le i \le c} u_{ij}), \quad (10)$$

is the monotonic decreasing as FCM in the iteration. This is owing to Lemma 2, which guarantees our modification without changing the hard-clustering. All those will be illustrated by experiments in Section IV.

Next section, we perform many experiments to illustrate the efficiency of the proposed algorithm.

IV. Experimental Results

To estimate the effectiveness and efficiency of the proposed MSFCM, experimental studies are conducted on synthetic data sets and real word data sets respectively. Four state-of-the-art clustering algorithms, FCM [10], Mini-batch SGFCM [16], LFCM and resFCM [17], are employed in these experiments to compare with the proposed MSFCM. The reason of the selections is that these algorithms have their own advantages. For example, Mini-batch SGFCM shows comparable or better accuracy with significant less time consumption [16]. LFCM and resFCM have a fast convergence for very large data sets [17]. All the experiments are run on a Personal Computer with a Intel Core i7-6700 CPU and a maximum of 8Gbytes of memory available for all processes. The computer runs Windows 7 with Matlab R2017a. In all experiments, we set the termination parameter $\varepsilon = 10^{-6}$ and the fuzziness weighting exponent $m = 2$ in all the experiments for simplicity.

A. Experiments on Synthetic data sets

In the first set of experiments, we have redone the experiments according to Fig. 1 with our proposed MSFCM for comparison. All the settings are same as the experiments in Fig. 1. The results are plotted in Fig. 3.

Comparing Fig. 3 with Fig. 1, we can observe that the convergence trajectories of $\mathcal{C}_1$ with MSFCM are reduced significantly, and the effect of the $\mathcal{C}_2$ is weakened. All the observations are in accordance with the previous analysis, which shows that MSFCM can achieve a fast convergence by boosting the effect of the in-cluster samples and weakening the effect of the out-of-cluster samples.

The other synthetic data set is D31, in which 31 clusters are generalized by the normal distribution with a same standard deviation 0.5 and their cluster centers are randomly placed in $\mathbb{R}^2$ as Fig. 4 shows. Each cluster has 200 samples. This similar dataset also appears in [20], [21]. We perform FCM and MSFCM on this data set and the results are illustrated in Fig. 4(a) and Fig. 4(b).

From the plotting of Fig. 4, we firstly see that, with the similar initial points, the clustering results of two algorithms are identical. Both algorithms can find all 31 cluster centers. However, the costs of two algorithm are greatly different. The iterations and the time of the proposed MSFCM are less than one third of those of FCM. The new scaling scheme can great improve the convergence speed of FCM clustering. Furthermore, comparing with Fig. 4(a) and 4(b), it is clear that the convergence trajectories of MSFCM are simpler than those of FCM. This is again because the new scaling scheme is boosting the effect of the in-cluster samples and weakening the effect of the out-of-cluster samples.

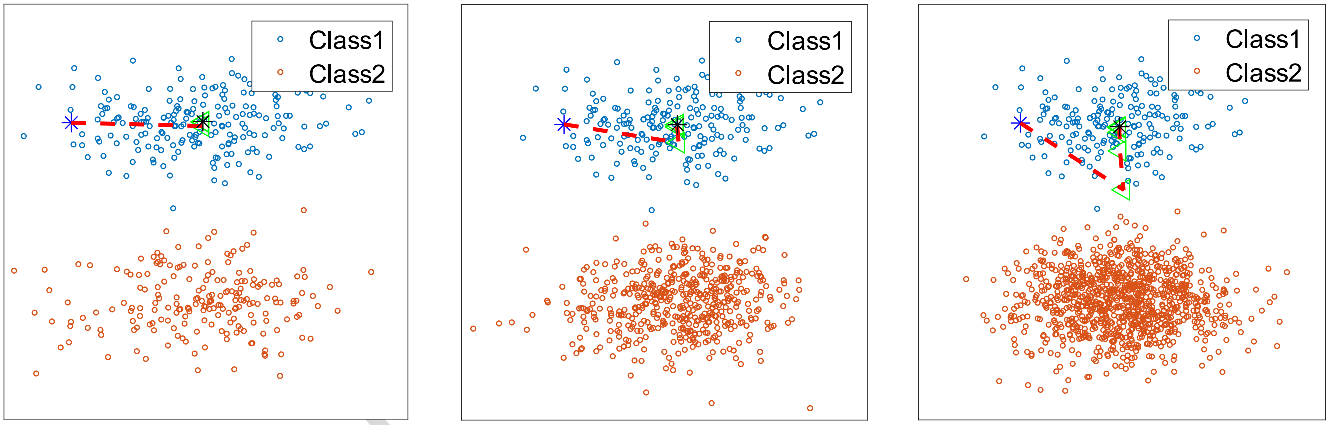
B. Experiments on Real world data sets

In order to evaluate the performance of different cluster algorithms, three external metrics, including the overall F-measure for the entire data set (F^*), Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) [16], [13], [19], [22], are used in this subsection. All three criteria, the larger the better, are used to measure the agreement of the ground truth and the clustering results produced by an algorithm. Specifically, let n be the total number of samples, $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_c\}$ be the partition of the ground truth, and $\{\hat{\mathcal{C}}_1, \hat{\mathcal{C}}_2, \cdots, \hat{\mathcal{C}}_{\hat{c}}\}$ be the partition by an algorithm. Denote that $\hat{n}_i = |\hat{\mathcal{C}}_i|$ is the number of samples in $\hat{\mathcal{C}}_i$, $n_l = |\mathcal{C}_l|$ is the number of samples in $\mathcal{C}_l$, and $n_i^l = |\mathcal{C}_l \cap \hat{\mathcal{C}}_i|$ is the number of the common objects in $\mathcal{C}_l$ and $\hat{\mathcal{C}}_i$, where $i = 1, 2, \cdots, \hat{c}$ and $l = 1, 2, \cdots, c$. Then the measure $F(l, i) = \frac{2n_i^l}{n_l + \hat{n}_i}$ is the harmonic mean of Precision and Recall of $\mathcal{C}_l$ and its potential prediction $\hat{\mathcal{C}}_i$. Therefore, the overall F-measure $\mathbf{F}^*$, **NMI** and **ARI** are defined as the following equations.

$$\mathbf{F}^* = \sum_{l=1}^{c} \frac{n_l}{n} \max\{F(l, i) | i = 1, \cdots, \hat{c}\}, \quad (11)$$

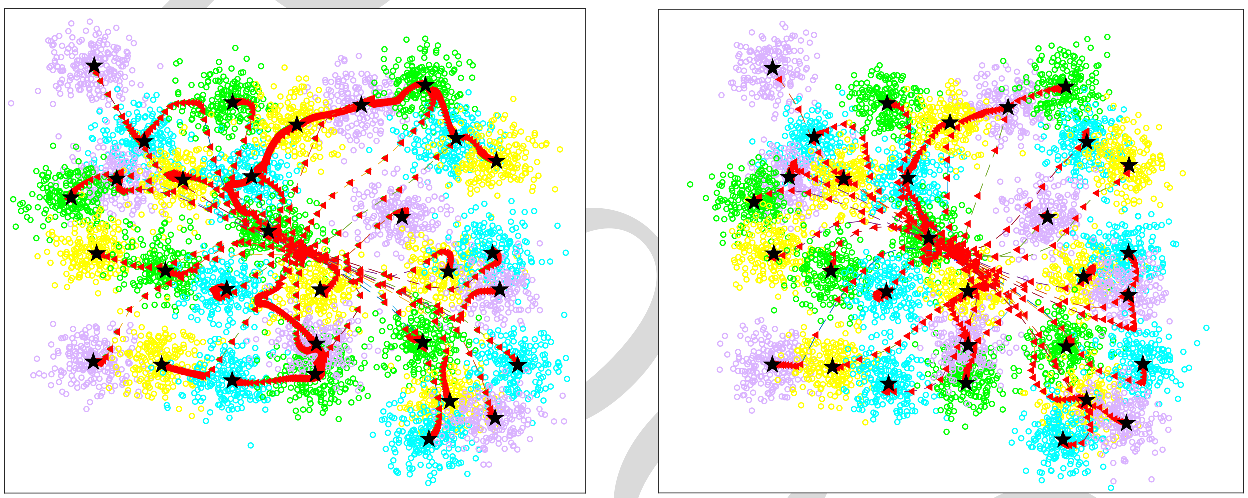
$$\mathbf{NMI} = \frac{\sum_{i=1}^{\hat{c}} \sum_{l=1}^{c} n_i^l \log(\frac{n \cdot n_i^l}{\hat{n}_i \cdot n_l})}{\sqrt{\left(\sum_{i=1}^{\hat{c}} \hat{n}_i \log(\frac{\hat{n}_i}{n})\right) \left(\sum_{l=1}^{c} n_l \log(\frac{n_l}{n})\right)}}, \quad (12)$$

$$\mathbf{ARI} = \frac{\sum_{i=1}^{\hat{c}} \sum_{l=1}^{c} \binom{n_i^l}{2} - \sum_{i=1}^{\hat{c}} \binom{t_i}{2} \sum_{l=1}^{c} \binom{s_l}{2} / \binom{n}{2}}{\frac{1}{2}\left(\sum_{i=1}^{\hat{c}} \binom{t_i}{2} + \sum_{l=1}^{c} \binom{s_l}{2}\right) - \sum_{i=1}^{\hat{c}} \binom{t_i}{2} \sum_{l=1}^{c} \binom{s_l}{2} / \binom{n}{2}}, \quad (13)$$



(a) $|\mathcal{C}_2| = 200$ (b) $|\mathcal{C}_2| = 500$ (c) $|\mathcal{C}_2| = 1000$

Fig. 3: The trajectories of the cluster centers of $\mathcal{C}_1$ in the MSFCM iterations. In this case, MSFCM algorithm is converged with 2, 3 and 4 iterations and costs 0.002, 0.003 and 0.006 seconds corresponding to (a), (b) and (c) respectively. In figures, the blue star is the initial point, the green triangle is the cluster center per iteration, and the black star is the target.



(a) D31 by FCM (b) D31 by MSFCM

Fig. 4: Clustering the synthetic data sets D31 by FCM and MSFCM. The initial points are randomly generated by initializing $\mathbf{U}$. The red solid triangles are the cluster centers per iteration and the black solid stars are the resulted cluster centers. (a) Clustering result of D31 by FCM, in which FCM converges within 286 iterations and costs 59.33 seconds; (b) Clustering result of D31 by MSFCM, where MSFCM converges only within 72 iterations and costs 5.58 seconds.

where $\binom{n}{i} = \frac{n!}{i!(n-i)!}$, $s_l = \sum_{i=1}^{\hat{c}} n_i^l$, and $t_i = \sum_{l=1}^{c} n_i^l$.

The chosen clustering methods are applied to seven real word data sets, which are obtained on the UCI machine learning repository[1]. The detailed information of the data sets is listed in the first column of TABLE I, where n is the number of training size, p is the dimension of the sample, and c is the given cluster number.

We perform the traditional FCM, LFCM and resFCM in [17], SGFCM [16] with batch size as $1\%n$, $2.5\%n$ and $5\%n$, and our proposed MSFCM. The experimental results are listed in TABLE I. All the results in TABLE I are averaged on ten trials with random initializations, and the standard deviations are also presented after the means. The best results are in bold.

[1] https://archive.ics.uci.edu/ml/index.php

From the experimental results in TABLE I, we have the following findings.

- Firstly from the results in last column, it observes that MSFCM almost wins on all data sets in term of F^*, ARI, NMI, training time and iteration in our setting. Hence the new proposed algorithm not only accelerates the clustering process of FCM, but also keeps a higher clustering quality.
- Comparing with FCM, the iterations of MSFCM are always less than one forth of those of FCM with the comparable clustering performance. Hence by the computational complexity analyzed in Section III, approximately at least one third of the training time is saved. At the same time, the cluster qualities of MSFCM are always better than or

TABLE I: Experimental results on seven real world data sets with different algorithms. All the values are averaged on ten trials with random initializations, where the standard deviations are presented after the means linked with ±. Best results are in bold.

Data sets	Evaluate criteria	FCM	LFCM	rseFCM	Mini-batch SGFCM			MSFCM
					1%	2.5%	5%	
Wine	F^*	0.699±0.001	0.699±0.001	0.701±0.005	0.667±0.032	0.666±0.061	0.682±0.050	**0.720±0.001**
(n=178	ARI	0.354±0.001	0.354±0.001	0.352±0.006	0.332±0.029	0.345±0.066	0.355±0.030	**0.375±0.001**
p=13	NMI	0.417±0.001	0.417±0.001	0.419±0.006	0.392±0.002	0.401±0.050	0.411±0.030	**0.432±0.001**
c=3)	Time/s	0.073±0.001	0.043±0.009	0.036±0.004	0.062±0.005	0.058±0.003	0.061±0.006	**0.035±0.001**
	Iteration	62.5±4.3	53.9±2.9	50.1±17.2	53.5±5.1	90.3±19.9	82.9±31.6	**17.1±2.3**
Vehicle	F^*	0.451±0.001	0.451±0.001	0.451±0.001	0.439±0.032	0.446±0.019	0.450±0.006	**0.453±0.001**
(n=846	ARI	**0.118±0.001**	**0.118±0.001**	**0.118±0.001**	0.110±0.020	0.103±0.021	0.113±0.002	**0.118±0.001**
p=18	NMI	0.180±0.001	0.180±0.001	0.181±0.001	0.166±0.009	0.172±0.013	0.180±0.002	**0.184±0.001**
c=4)	Time/s	0.187±0.030	0.187±0.053	0.181±0.024	0.321±0.053	0.589±0.082	0.209±0.136	**0.130±0.022**
	Iteration	95.5±5.4	63.9±7.6	56.7±10.3	135.6±50.5	399.1±79.6	301.9±191.8	**30.6±4.9**
Segment	F^*	0.590±0.032	0.539±0.036	0.589±0.007	0.519±0.059	0.559±0.050	0.572±0.034	**0.612**±0.037
(n=2310	ARI	0.352±0.001	0.335±0.060	0.366±0.010	0.312±0.051	0.356±0.026	0.355±0.053	**0.391**±0.008
p=19	NMI	0.473±0.036	0.455±0.035	0.472±0.051	0.450±0.061	0.479±0.021	0.466±0.034	**0.530±0.002**
c=7)	Time/s	3.610±1.587	3.152±0.899	2.028±0.569	4.961±0.178	4.653±0.792	5.970±3.125	**0.764±0.012**
	Iteration	371.6±80.8	152.5±36.8	157.7±51.2	1583.1±535.7	2536.7±109.3	4631.9±537.1	**48.2±10.1**
Satimage	F^*	0.553±0.001	0.553±0.001	0.553±0.001	0.572±0.035	0.561±0.021	0.567±0.013	**0.608**±0.004
(n=6435	ARI	0.292±0.001	0.292±0.001	0.292±0.001	0.290±0.003	0.299±0.013	0.321±0.015	**0.350±0.001**
p=36	NMI	0.450±0.001	0.458±0.001	0.458±0.001	0.455±0.032	0.454±0.002	0.463±0.005	**0.486**±0.011
c=6)	Time/s	2.627±0.326	2.283±0.539	3.728±0.581	4.826±3.762	5.662±2.113	6.991±3.692	**0.884±0.093**
	Iteration	157.9±12.1	55.5±7.6	45.6±8.2	968.3±277.9	1992.5±368.8	1556.3±522.1	**39.2**±31.3
Avila	F^*	0.277±0.001	0.275±0.001	0.276±0.002	0.269±0.002	0.263±0.027	0.279±0.050	**0.306**±0.023
(n=20867	ARI	0.010±0.001	0.009±0.001	0.009±0.002	0.011±0.020	0.015±0.036	0.020±0.002	**0.022±0.001**
p=10	NMI	0.039±0.001	0.038±0.001	0.038±0.001	0.040±0.012	0.047±0.011	0.050±0.001	**0.065±0.001**
c=12)	Time/s	657.4±35.2	303.3±9.6	179.6±15.6	89.6±20.6	86.9±9.3	99.3±29.6	**20.5±6.6**
	Iteration	3957.9±199.6	1053.9±25.5	1096.7±50.9	6639.1±521.5	6766.9±810.7	6593.4±598.3	**153.5**±36.6
Shuttle	F^*	0.504±0.001	0.504±0.001	0.503±0.002	0.499±0.007	0.493±0.013	0.490±0.030	**0.512**±0.012
(n=58000	ARI	0.114±0.001	0.114±0.001	0.114±0.001	0.102±0.013	0.109±0.031	0.115±0.002	**0.153**±0.027
p=9	NMI	0.244±0.001	0.244±0.001	0.246±0.005	0.248±0.009	0.263±0.013	0.269±0.002	**0.271**±0.009
c=7)	Time/s	38.9±7.2	70.9±19.1	53.5±17.5	59.3±29.2	59.3±29.1	48.3±29.1	**17.6±5.5**
	Iteration	206±20.5	221.6±19.9	199.0±25.5	3329.2±396.3	4561.2±179.6	4695.7±299.6	**70.7±16.5**
Seismic	F^*	0.448±0.001	0.448±0.001	0.448±0.001	0.436±0.020	0.439±0.006	0.443±0.030	**0.451±0.001**
(n=78823	ARI	**0.038±0.001**	**0.038±0.001**	**0.038±0.001**	0.036±0.002	0.036±0.009	0.035±0.006	**0.038±0.001**
p=50	NMI	**0.043±0.001**	**0.043±0.001**	**0.043±0.001**	**0.043**±0.002	0.037±0.011	0.038±0.009	**0.043±0.001**
c=3)	Time/s	52.8±3.9	81.9±5.3	53.1±7.6	39.3±9.1	42.1±10.3	31.6±8.1	**13.3±1.2**
	Iteration	259.6±2.9	209.6±31.9	200.9±15.6	2690.2±296.1	3165.5±513.9	2937.7±399.6	**50.5**±3.5

similar with those of FCM.

- Sometimes Mini-batch SGFCM is faster than MSFCM, but its clustering qualities, measured by F^*, ARI and NMI, are always lower than those of MSFCM. And it is clear that the variances of SGFCM are always large, hence the variance-reduced version SGD [23] may be helpful to improve this type of algorithms on this aspect.
- Compared with the LFCM and resFCM on larger data sets, as Avila, Shuttle, and Seismic, it observes that their clustering qualities are comparable. At the same time, the proposed MSFCM always has fewer iterations and less training time. Since LFCM and resFCM [17] are designed for very large data sets, our MSFCM should also have the potential possibility to modify to train the very large data sets. We have not discussed this because our current computer has not meet the needs of the very large data sets.

All of those findings again illustrate that the new scaling scheme can great improve not only the convergence speed of FCM clustering but also the clustering quality. Thus, the scheme of boosting the effect of the in-cluster samples and weakening the effect of the out-of-cluster samples works also well on real world data sets.

C. The detailed analysis of the proposed algorithm

In this section, we perform experiments to show some detailed characters of the new proposed MSFCM.

1) The monotonicity of MSFCM: In this part, we do experiment to reveal the the monotonic characteristic of MSFCM, where the fuzzy objective $J(\mathbf{U},\mathbf{V})$ in (1) and the hard c-means objective $\hat{J}(\mathbf{U},\mathbf{V})$ in (10) are discussed. A typical trial of FCM and MSFCM on seven data sets as listed in Table I is given in Fig.5(a) and (b) respectively. And the corresponding hard c-means results are plotted in Fig. 5(c) and (d). In experiments, two algorithms have the same initialization for every data set. In order to plot all results of different data set in one figure, the y-axis is the ratio of objectives to its initial value.

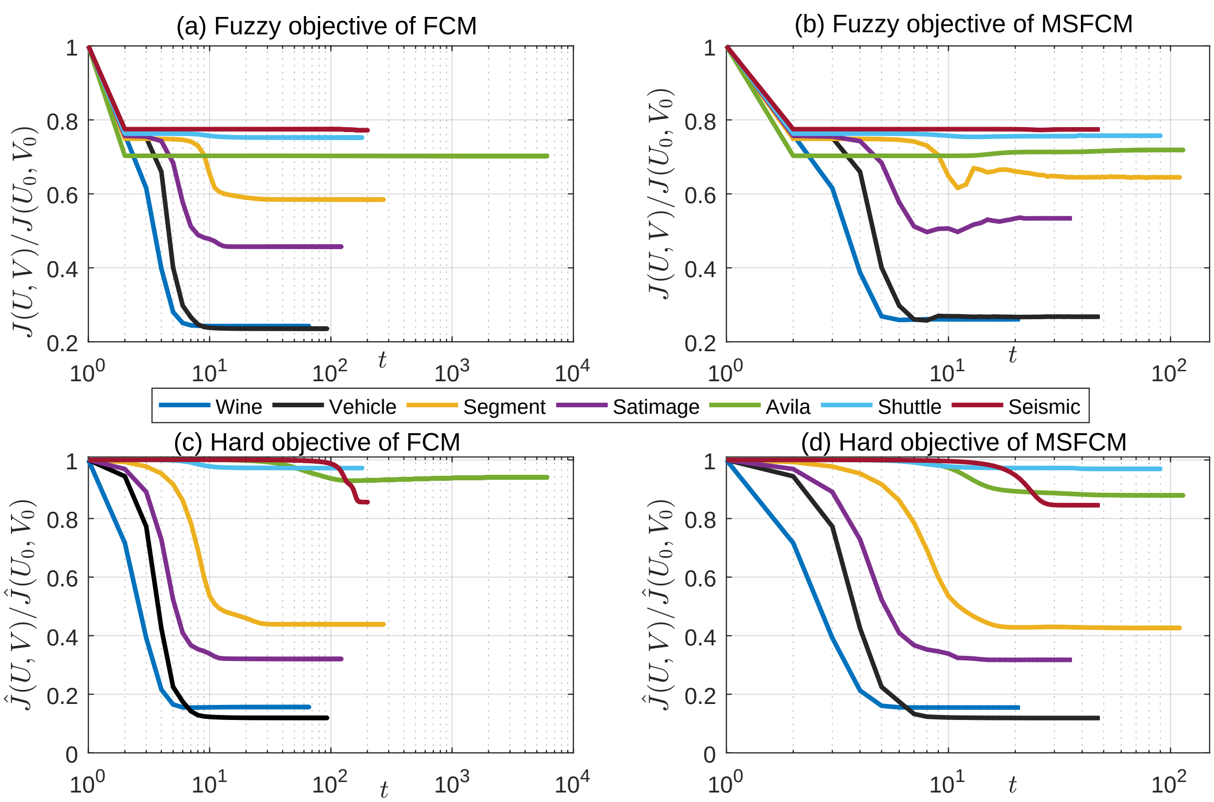


Fig. 5: Plots of $\frac{J(\mathbf{U},\mathbf{V})}{J(\mathbf{U}_0,\mathbf{V}_0)}$ and $\frac{\hat{J}(\mathbf{U},\mathbf{V})}{\hat{J}(\mathbf{U}_0,\mathbf{V}_0)}$ with respect to iterations on seven data sets, where FCM and MSFCM have the same initialization for every data set. The plots reveal that the fuzzy objectives of MSFCM are not as monotone decreasing as those of FCM, while the corresponding hard c-means objectives are all monotone decreasing.

By the plots in Fig. 5(a) and (b), we clearly observe that the fuzzy objectives of MSFCM are not as monotonically decreasing as those of FCM, since we modified the values derived by optimization method. However,they keep decreasing or convergence in most cases. On the other hand, the corresponding hard c-means objectives maintain the monotone decreasing characteristic as those of FCM (see Fig. 5(c) and (d)). At the same time, we observe that MSFCM obtains the better or similar hard c-means objective values with less iterations comparing to FCM.

This set of experiments is again on the selected data sets in Table I. It is designed to reveal how much the new scaling scheme can improve the clustering process by the triangle inequality (6). Let $\hat{n}_t = |J_t|$ be the number of samples filtered by the triangle inequality (6) in iteration t. For all seven data sets, we plot the curves of $\frac{\hat{n}_t}{n}$ with respect to iteration t in Fig. 6, where the log-scale of x-axis is to clearly show the differences of iterations at the very beginning on different data sets .

From the plots in Fig. 6, we can clearly observe that the triangle inequality can filter out a large amount of samples that will not change their nearest clusters in next iteration. Furthermore, since at least 40% of samples are filtered out after about 10 iterations, the new update scheme (9) is always accomplished very efficiently.

From those experimental results, we can conclude that the success of MSFCM is because we have iteratively obtained

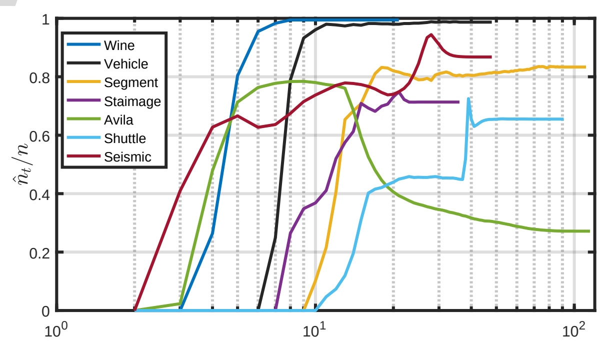


Fig. 6: Plots of $\frac{\hat{n}}{n}$ with respect to iterations.

plenty of priori information by Lemma 2 and used them to scale the fuzzy membership coefficients reasonably according to Lemma 1. Hence, the convergence of the clustering process is definitely improved.

3) Varying fuzziness parameter m: The fuzziness parameter m is a key parameter that can affect the result of FCM clustering [24], [25]. This set of experiments is designed to illustrate that the performance of MSFCM is also fluctuating according to the fuzziness parameter m. For the parameter m ranged from 1.2 to 3.6 by step of 0.2, F^* scores of seven data sets are plotted in Fig. 7. In Fig. 7, the lines of the same color are obtained on the identical data set with the same random initialization by FCM and MSFCM, where the dotted lines are obtained by FCM and the solid lines are obtained by MSFCM. All results are averaged on ten trials.

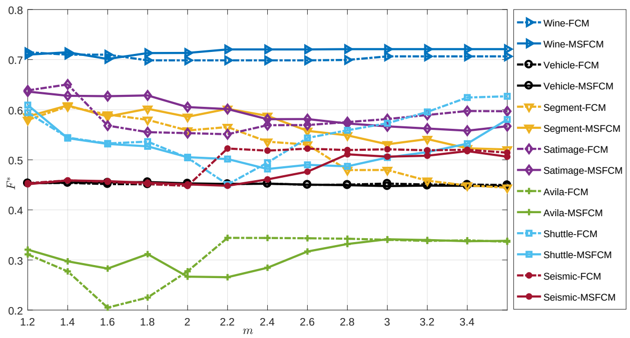


Fig. 7: Plots of F^* with respect to fuzziness parameter m. The lines of the same color are obtained on one data set by FCM and MSFCM with the same random initialization, where the dotted lines are obtained by FCM and the solid lines are obtained by MSFCM.

From the results in Fig. 7, we have the following findings. Firstly, we observes that MSFCM is also fluctuating with respect to m as FCM does. Hence, we also need to tune the parameter m for different data sets. At the same time, it shows that, if m small enough, the performance of those two algorithms is very similar, since all of them will incline to hard c-means algorithm.

From the statistical point of view, the F^* scores of MSFCM are better than those of FCM, since the solid line is plotted above the dotted line with the same color in most case. However, there are two data sets, Shuttle and Seismic, on which FCM is better than MSFCM for different m. So it recommends that the new MSFCM is not a substitute of FCM, but a good supplement.

Of course, the advantage of MSFCM is again on its training speed as Table I shows. Such as for plotting Fig. 7 (totally 7*13*10 trials), FCM needs 397,108 iterations and 455.2 minutes in clustering totally. However, MSFCM only costs 143,239 iterations and 293.9 minutes in all.

V. Conclusion

In this paper, we propose a new membership scaling FCM and verify its effectiveness in the synthetic data sets and real world data sets. We firstly use the triangle inequality to filter out many samples that will not change their nearest clusters in next iteration, and scale the fuzzy membership coefficients according to a new scheme, in which the effect of the in-cluster samples is boosted and the effect of the out-of-cluster samples is weakened. Many experimental results show that the new algorithm is more efficient than or comparable with the state-of-the-art fuzzy clustering methods. Hence, it is a good supplement to fuzzy clustering.

In the experiments, we also find that new algorithm is comparable with some algorithms designed for clustering the very large data sets. Hence we will amend the new algorithm further also for very large data sets in the future. Another interesting aspect is to generalize our idea to nonlinear fuzzy clustering with Kernel FCM as [17] did.

References

[1] J. C. Bezdek, *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media, 2013.
[2] 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.
[3] L. Zhang, W. Zhong, C. Zhong, L. Wei, X. Liu, and W. Pedrycz, "Fuzzy c-means clustering based on dual expression between cluster prototypes and reconstructed data," *International Journal of Approximate Reasoning*, vol. 90, 2017.
[4] H. H. Nguyen and P. Cohen, "Gibbs random fields, fuzzy clustering, and the unsupervised segmentation of textured images," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 1, pp. 1–19, 1993.
[5] I. Gath and A. B. Geva, "Unsupervised optimal fuzzy clustering," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 7, pp. 773–780, 1989.
[6] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.
[7] 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.
[8] 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.
[9] 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.
[10] 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.
[11] Z. Jiang, T. Li, W. Min, Z. Qi, and Y. Rao, "Fuzzy c-means clustering based on weights and gene expression programming," *Pattern Recognition Letters*, vol. 90, pp. 1–7, 2017.
[12] R. J. Hathaway and Y. Hu, "Density-weighted fuzzy c-means clustering," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 1, pp. 243–252, 2008.
[13] 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.
[14] T. Lei, X. Jia, Y. Zhang, L. He, H. Meng, and A. K. Nandi, "Significantly fast and robust fuzzy c-means clustering algorithm based on morphological reconstruction and membership filtering," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 3027–3041, 2018.
[15] D. C. Park and I. Dagher, "Gradient based fuzzy c-means (GBFCM) algorithm," in *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, vol. 3. IEEE, 1994, pp. 1626–1631.
[16] Y. Wang, L. Chen, and J.-P. Mei, "Stochastic gradient descent based fuzzy clustering for large data," in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE, 2014, pp. 2511–2518.
[17] T. C. Havens, J. C. Bezdek, C. Leckie, L. O. Hall, and M. Palaniswami, "Fuzzy c-means algorithms for very large data," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 6, pp. 1130–1146, 2012.
[18] A. Khoshkbarchi, A. Kamali, M. Amjadi, and M. A. Haeri, "A modified hybrid fuzzy clustering method for big data," in *2016 8th International Symposium on Telecommunications (IST)*. IEEE, 2016, pp. 196–201.
[19] 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.
[20] C. J. Veenman, M. J. T. Reinders, and E. Backer, "A maximum variance cluster algorithm," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 9, pp. 1273–1280, 2002.
[21] S. N. Parizi, K. He, R. Aghajani, S. Sclaroff, and P. Felzenszwalb, "Generalized majorization-minimization," *Computer Science*, vol. 121, no. 2, pp. 95–108, 2015.
[22] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 2, no. 1, pp. 193–218, 1985.
[23] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Advances in neural information processing systems*, 2013, pp. 315–323.
[24] J. Yu and M.-S. Yang, "Optimality test for generalized fcm and its application to parameter selection," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 1, pp. 164–176, 2015.
[25] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Transactions on Fuzzy systems*, vol. 3, no. 3, pp. 370–379, 1995.