The *K*-Means-Type Algorithms Versus Imbalanced Data Distributions

Jiye Liang, Liang Bai, Chuangyin Dang, *Senior Member, IEEE*, and Fuyuan Cao

***Abstract*—K-means is a partitional clustering technique that is well-known and widely used for its low computational cost. The representative algorithms include the hard k-means and the fuzzy k-means. However, the performance of these algorithms tends to be affected by skewed data distributions, i.e., imbalanced data. They often produce clusters of relatively uniform sizes, even if input data have varied cluster sizes, which is called the "uniform effect." In this paper, we analyze the causes of this effect and illustrate that it probably occurs more in the fuzzy k-means clustering process than the hard k-means clustering process. As the fuzzy index m increases, the "uniform effect" becomes evident. To prevent the effect of the "uniform effect," we propose a multicenter clustering algorithm in which multicenters are used to represent each cluster, instead of one single center. The proposed algorithm consists of the three subalgorithms: the fast global fuzzy k-means, Best M-Plot, and grouping multicenter algorithms. They will be, respectively, used to address the three important problems: 1) How are the reliable cluster centers from a dataset obtained? 2) How are the number of clusters which these obtained cluster centers represent determined? 3) How is it judged as to which cluster centers represent the same clusters? The experimental studies on both synthetic and real datasets illustrate the effectiveness of the proposed clustering algorithm in clustering balanced and imbalanced data.**

***Index Terms*—Imbalanced data, multirepresentatives, the k-means-type clustering algorithms, the number of clusters, the production of cluster centers.**

I. Introduction

CLUSTER analysis is an important branch in statistical multivariate analysis and unsupervised machine learning which has extensive applications in various domains, including financial fraud, medical diagnosis, image processing, information retrieval, and bioinformatics. The goal of clustering is to group a set of objects into clusters so that objects in the same cluster have high similarity but are very dissimilar with objects in other clusters. Many types of clustering techniques have been developed in the literature (see, e.g., [1] and references therein). Among them, k-means is one of the most computationally efficient clustering techniques, which begins with an initial set of cluster centers and iteratively refines this set so as to decrease the sum of squared errors. The representative algorithms include the hard k-means [2] and the fuzzy k-means [3], [4]. The hard k-means algorithm classifies each object of the dataset just to one cluster. However, in many real situations, issues such as limited spatial resolution, poor contrast, overlapping intensities, noise, and intensity inhomogeneities reduce its effectiveness. In the fuzzy k-means algorithm, each object is allowed to have memberships in all clusters rather than having a distinct membership in one single cluster. Compared with the hard clustering algorithm, the fuzzy clustering algorithm has robust characteristics for ambiguity and can retain much more information, which makes it widely used to tackle effectively many problems in real applications. We know that when the fuzzy index m approaches 1, the fuzzy k-means algorithm becomes the hard k-means algorithm [5]. This conclusion states us that the hard k-means algorithm is a special case of the fuzzy k-means algorithm.

The k-means-type algorithms have attracted great interest in the literature. There are considerable research efforts to characterize the key features of the k-means-type clustering algorithms. People have identified several factors [6] that may strongly affect the k-means clustering analysis including high dimensionality [7]–[9], sparseness of the data [10], noise and outliers in the data [11]–[13], scales of the data [14]–[17], types of attributes [18], [19], the fuzzy index m [20]–[23], initial cluster centers [24]–[29], and the number of clusters [30]–[32]. However, further investigation is needed to understand how data distributions can have the impact on the performance of k-means-type clustering. For skewed-distributed data, such as imbalance data, the k-means-type clustering algorithms tend to have poor performance because they often partition a part of objects belonging to the majority classes into the minority classes, which makes clusters have relatively uniform sizes, although input data have varied cluster sizes. This effect is called the "uniform effect" in [33]. It is of great value to study how imbalanced data distributions affect the performance of the k-means-type algorithms because many real data are imbalanced, such as fraud detection, oil spill detection, risk management, and medical

Manuscript received April 7, 2011; revised September 9, 2011; accepted December 17, 2011. Date of publication January 2, 2012; date of current version August 1, 2012. This work was supported by the National Natural Science Foundation of China under Grant 71031006, Grant 70971080, and Grant 60903110; by the grant GRF: CityU 112809 of Hong Kong Special Administrative Region Government; by the Foundation of Doctoral Program Research of Ministry of Education of China under Grant 20101401110002; and by the National Key Basic Research and Development Program of China (973 Program) under Grant 2011CB311805.

J. Y. Liang and F. Y. Cao are with the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China (e-mail: ljy@sxu.edu.cn; cfy@sxu.edu.cn).

L. Bai is with the Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China, and also with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong (e-mail: sxbailiang@126.com).

C. Dang is with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong (e-mail: mecdang@cityu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TFUZZ.2011.2182354

diagnosis/monitoring. Furthermore, people are also interested in identifying rare classes in the datasets with imbalanced data distributions. For example, in the domain of network intrusion detection, the number of malicious network activities is usually very small compared with the number of normal network connections.

However, most of these studies on imbalanced data focus on supervised learning. In unsupervised learning, Xiong *et al*. [33] provided a formal and organized study of the effect of skewed data distributions on the hard k-means clustering. However, the theoretic analysis is only based on the hard k-means algorithm. Our experimental studies illustrate that the fuzzy k-means clustering algorithm has more evident "uniform effect" than the hard k-means clustering algorithm. Moveover, as the fuzzy index m increases, the "uniform effect" becomes obvious. A theoretic analysis on the effect of imbalanced data distributions on the fuzzy k-means clustering is provided in Section III.

Furthermore, how to use the k-means-type technique to effectively cluster imbalanced data is also an important issue. In the literature, several variations of the fuzzy k-means algorithm, such as the kernel-based fuzzy algorithms [34]–[36], the Gustafson–Kessel algorithm [37], and the iterative compatible cluster merging algorithm [38], have been proposed to discover clusters of varied sizes. Although they have more robustness to cluster datasets with varied cluster sizes than the fuzzy k-means algorithm, we have found that these algorithms cannot effectively avoid the occurrence of the "uniform effect." Because of the fact that a single center cannot sufficiently represent a majority class, a subset of objects in the majority class is often wrongly partitioned into several minority classes. In this paper, we will use multicenters to represent each cluster, instead of one single center, because multicenters can help divide the objects of a majority class into several subclusters with considerable smaller sizes similar to those of the minority classes. This can rebalance the scales of the majority classes and the minority classes to reduce the effect of imbalanced data distributions on the performance of the k-means-type algorithms.

Guha *et al*. [39] early proposed to make use of multiple representative points to get the shape information of the "natural" clusters with nonspherical shapes [1] and achieve an improvement on noise robustness over the single-link algorithm. In [40], a multiprototype clustering algorithm was proposed, which applies the hard k-means algorithm to discover clusters of arbitrary shapes and sizes. However, there are following problems in the real applications of these algorithms to cluster imbalanced data. 1) These algorithms depend on a set of parameters whose tuning is problematic in practical cases. 2) These algorithms make use of the randomly sampling technique to find cluster centers. However, when data are imbalanced, the selected samples more probably come from the majority classes than the minority classes. 3) The number of clusters k needs to be determined in advance as an input to these algorithms. In a real dataset, k is usually unknown. 4) The separation measures between subclusters that are defined by these algorithms cannot effectively identify the complex boundary between two subclusters. Their shortcomings are analyzed in Section IV. 5) The definition of clusters in these algorithms is different from that of k-means. Their definition is similar to that of density-based spatial clustering of applications with noise (DBSCAN) [41]. They assume that if objects are density connected, they should belong to the same cluster. A major advantage of such definition is that clusters of different shapes can be found. However, when data are nonuniform and sparse, objects in the same cluster tend to be density unconnected. This induces the objects in the same cluster to be partitioned into several clusters. In the k-means-type algorithms, a cluster is viewed as a hypersphere. The density nonconnectivity within a cluster has a minor impact on discovering the cluster. However, a disadvantage of this definition is that only the clusters which are linearly separable in the input space can be obtained. To overcome this limitation, the kernel techniques are applied to the k-means-type algorithms [34]–[36], which implicitly map the nonlinearly separable data in the input space into a high-dimensional space where data are linearly separable. In fact, we cannot determine which one of the two definitions is right because different people have different interpretations about what is a cluster. The answer of this question often depends on human experience and intent. In this paper, we only consider that clusters are linearly separated. The nonlinearly separated clusters can be dealt with by the kernel trick, which is outside the scope of this paper.

In this paper, we will first give a theoretic analysis on the effect of imbalanced data distributions on the fuzzy k-means algorithm. Furthermore, in order to avoid the "uniform effect," we will propose a multicenter (MC) clustering algorithm which uses multicenters to represent each cluster, instead of one single center. This algorithm consists of the three subalgorithms: the fast global fuzzy k-means (FGFKM), "Best-M Plot" (BMP), and grouping multicenter (GMC) algorithms. First, the FGFKM algorithm will be presented to obtain several reliable cluster centers. Next, we will propose the BMP algorithm to find the most appropriate value of m and determine the number of clusters k. Finally, a new separation measure between subclusters will be defined. Based on this measure, the GMC algorithm will be proposed to group the cluster centers to represent k clusters.

The rest of this paper is organized as follows. A detailed review of the k-means-type algorithms is presented in Section II. Section III illustrates the effect of imbalanced data distributions on the k-means-type clustering. In Section IV, an MC clustering algorithm is proposed to effectively cluster imbalanced data. Section V illustrates the effectiveness of the proposed algorithm. Finally, a concluding remark is given in Section VI.

II. K-Means-Type Clustering Algorithms

Let $X = \{X_1, X_2, \ldots, X_n\}$ be a set of n objects. Object $X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,d}\}$ is characterized by a set of d attributes (variables). The k-means-type algorithms [2]–[4] search for a partition of X into k clusters that minimize the objective function F with unknown variables U and V as follows:

$$F(U,V) = \sum_{l=1}^{k}\sum_{i=1}^{n}(u_{l,i})^m \|X_i - V_l\|^2 \qquad (1)$$

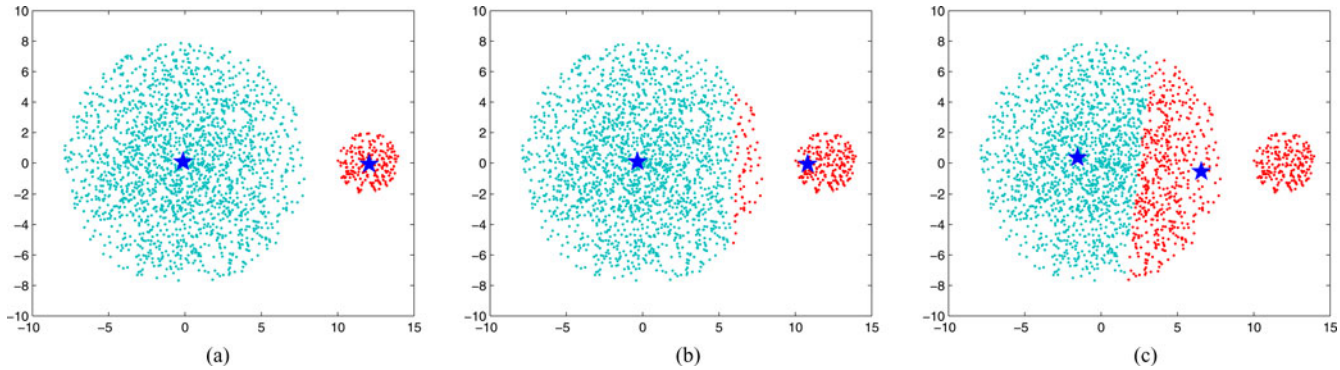


Fig. 1. "Uniform effect" occurs when the hard k-means algorithm is implemented. (a) Imbalanced data distribution before clustering. (b) Clustering result in the first iteration. (c) Clustering result in the last iteration.

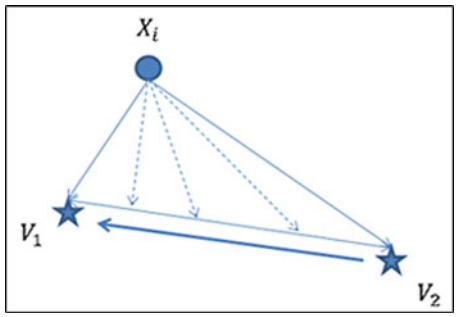


Fig. 2. As $\|V_1 - V_2\|^2$ decreases, the difference between $\|X_i - V_1\|^2$ and $\|X_i - V_2\|^2$ decreases.

subject to

$$u_{l,i} \in [0,1], \quad \sum_{l=1}^{k} u_{l,i} = 1, \quad 0 < \sum_{i=1}^{n} u_{l,i} < n \tag{2}$$

for $1 \le i \le n$ and $1 \le l \le k$, where $U = [u_{l,i}]$ is a k-by-n real matrix, with $u_{l,i}$ being the membership degree of the ith object X_i to the lth cluster; $V = [V_1, V_2, \ldots, V_k]$; $V_l = [v_{l,1}, v_{l,2}, \ldots, v_{l,d}]$ is the lth cluster center with d attributes; $\|X_i - V_l\|^2 = \sum_{j=1}^{d}(x_{i,j} - v_{l,j})^2$ is the Euclidean distance between the object X_i and the center V_l of the lth cluster; and $m \in [1, +\infty)$ is the fuzzy index. When $m = 1$, the fuzzy k-means clustering algorithm becomes the hard k-means clustering algorithm.

The optimal minimum value of F is normally solved by the alternative optimization method. It relates to two updated equations: U and V. The update equations are given as follows.

If $m = 1$, U is updated by

$$\hat{u}_{l,i} = \begin{cases} 1, & \text{if } \|X_i - \hat{V}_l\|^2 \le \|X_i - \hat{V}_h\|^2, 1 \le h \le k \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

for $1 \le i \le n$, $1 \le l \le k$. If $m > 1$, U is updated by

$$\hat{u}_{l,i} = 1 \Big/ \sum_{h=1}^{k} \left[\frac{\|X_i - \hat{V}_l\|^2}{\|X_i - \hat{V}_h\|^2} \right]^{1/(m-1)} \tag{4}$$

for $1 \le i \le n$, $1 \le l \le k$.

V is updated by

$$\hat{v}_{l,j} = \frac{\sum_{i=1}^{n} (\hat{u}_{l,i})^m x_{i,j}}{\sum_{i=1}^{n} (\hat{u}_{l,i})^m} \tag{5}$$

for $1 \le l \le k$, $1 \le j \le d$.

Unlike the hard k-means algorithm, the fuzzy k-means algorithm produces a fuzzy partition matrix U. We obtain the cluster memberships from U as follows. The object X_i is assigned to the lth cluster if $u_{l,i} = \max_{1 \le h \le k} u_{h,i}$. When the maximum is not unique, if X_i is assigned to the cluster of first achieving the maximum, a partition of X is formed. If we assign X_i to all the clusters of achieving the maximum, a covering of X is formed.

III. Effect of Imbalanced Data Distributions on K-Means-Type Clustering

In [33], Xiong *et al*. provided a formal and organized study of the effect of skewed data distributions on the hard k-means clustering. Furthermore, they formally illustrated that the hard k-means clustering algorithm tends to produce clusters of relatively uniform sizes, even if input data have varied "true" cluster sizes. This effect is called the "uniform effect."

They discussed the effect of the hard k-means clustering on the distribution of the cluster sizes when the number of clusters k is 2. They rewrote the objective function (1) as

$$F(U,V) = \frac{-2n_1 n_2 \|V_1 - V_2\|^2}{2n} + \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} \|X_i - X_j\|^2}{2n} \tag{6}$$

where n_l is the number of objects in the lth cluster, for $l = 1, 2$. We know that $\sum_{i=1}^{n}\sum_{j=1}^{n} \|X_i - X_j\|^2$ is a constant for a given dataset, regardless of U and V. In addition, $n = \sum_{l=1}^{2} n_l$ is the total number of objects in the data. In other words, the minimization of the objective function F is equivalent to the maximization of $n_1 n_2 \|V_1 - V_2\|^2$. If the effect of $\|V_1 - V_2\|^2$ is isolated, the maximization of $n_1 n_2 \|V_1 - V_2\|^2$ implies the maximization of $n_1 n_2$, which leads to $n_1 = n_2 = n/2$. This means that for the minimization of the objective function F, the hard k-means algorithm tends to produce clusters of relatively uniform sizes. For instance, Fig. 1 shows a scenario where the "uniform effect" occurs when the hard k-means clustering algorithm is used to cluster an imbalanced dataset which contains a majority cluster (2000 objects) and a minority cluster (200 objects).

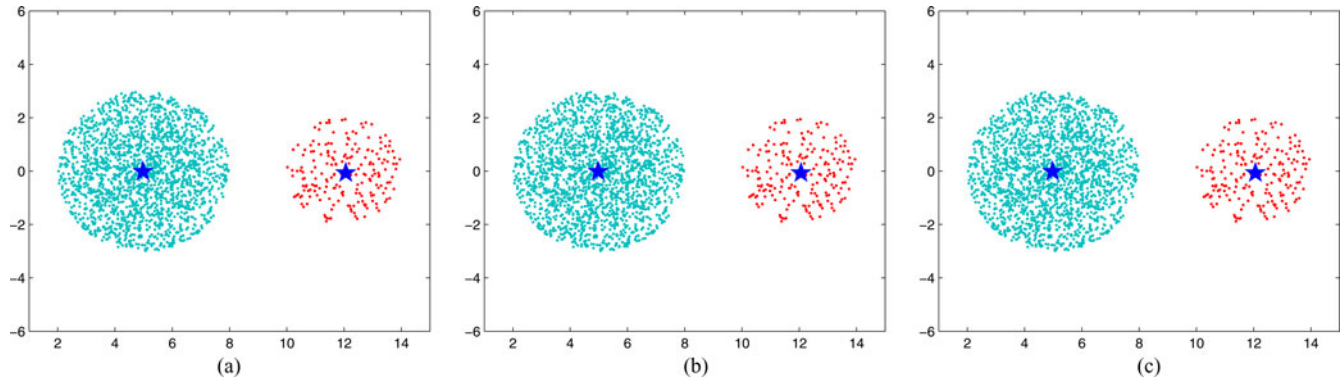


Fig. 3. "Uniform effect" does not occur when the hard k-means algorithm is implemented. (a) Imbalanced data distribution before clustering. (b) Clustering result in the first iteration. (c) Clustering result in the last iteration.

In the aforementioned analysis, the effect of the component $\|V_1 - V_2\|^2$ is isolated. However, for real-world datasets, the values of the two components are related to each other. According to (6), we also see that the smaller the distance between the cluster centers, the larger the effect of the "uniform effect', whereas when the distance between the cluster centers does not decrease in iterations, the probability of the occurrence of the "uniform effect" decreases. In the following, we will analyze the contribution of the distance between cluster centers, i.e., $\|V_1 - V_2\|^2$, to the occurrence of the "uniform effect" when the k-means-type algorithms are used to cluster imbalanced data. To simplify the analysis, we only consider that the number of clusters is 2. Let D_1 be the majority class and D_2 be the minority class. To minimize the effect of initial cluster centers on the performance of the k-means-type algorithms, we set $V_l^{(1)} = \sum_{X_i \in D_l} / X_i|D_l|$ for $l = 1, 2$, which are the "true" cluster centers.

We rewrite

$$V_2 = \frac{\sum_{i=1, X_i \in D_1}^{n} (u_{2,i})^m X_i + \sum_{i=1, X_i \in D_2}^{n} (u_{2,i})^m X_i}{\sum_{i=1, X_i \in D_1}^{n} (u_{2,i})^m + \sum_{i=1, X_i \in D_2}^{n} (u_{2,i})^m}. \quad (7)$$

According to (7), we know that the larger the $\sum_{i=1, X_i \in D_2}^{n} (u_{2,i})^m$ is compared with $\sum_{i=1, X_i \in D_1}^{n} (u_{2,i})^m$, the more important role the objects in D_2 play in computing V_2. If $\sum_{i=1, X_i \in D_2}^{n} (u_{2,i})^m \gg \sum_{i=1, X_i \in D_1}^{n} (u_{2,i})^m$, V_2 has good representability in D_2, whereas if $\sum_{i=1, X_i \in D_2}^{n} (u_{2,i})^m \leq \sum_{i=1, X_i \in D_1}^{n} (u_{2,i})^m$, the objects in D_1 play a more important role in computing V_2 than the objects in D_2. In this case, V_2 has more representability in D_1 than D_2. When the value of m is larger than 1, $\sum_{i=1, X_i \in D_1}^{n} (u_{2,i})^m$ is not equal to zero. This means that the objects in D_1 always take a certain relevance in computing V_2. To simplify the analysis, we set U as

$$\hat{u}_{l,i} = \begin{cases} 1/q, & \text{if } X_i \notin D_l, \\ 1 - 1/q, & \text{if } X_i \in D_l, \end{cases} \quad (8)$$

for $1 \leq i \leq n$, $1 \leq l \leq 2$, where $q > 1$. We write V as

$$V_1 = \frac{(1 - 1/q)^m \sum_{i=1, X_i \in D_1}^{n} X_i + (1/q)^m \sum_{i=1, X_i \in D_2}^{n} X_i}{|D_1|(1 - 1/q)^m + |D_2|(1/q)^m} \quad (9)$$

and

$$V_2 = \frac{(1/q)^m \sum_{i=1, X_i \in D_1}^{n} X_i + (1 - 1/q)^m \sum_{i=1, X_i \in D_2}^{n} X_i}{|D_1|(1/q)^m + |D_2|(1 - 1/q)^m}. \quad (10)$$

Given D_2, when the number of objects in D_1 approaches a very large value, we can obtain

$$V_l \approx \frac{\sum_{i=1, X_i \in D_1}^{n} X_i}{|D_1|} \quad (11)$$

for $1 \leq l \leq 2$.

This states us that $\|V_1 - V_2\|^2$ decreases as the number of objects in D_1 increases. Because of $\|V_1 - V_2\|^2 \geq |\|X_i - V_1\|^2 - \|X_i - V_2\|^2|$, the closer the V_2 to V_1, the smaller the difference between $\|X_i - V_1\|^2$ and $\|X_i - V_2\|^2$, which is shown in Fig. 2. This means that $u_{2,i}$ will be close to $u_{1,i}$, for $1 \leq i \leq n$. The closer the V_2 to V_1, the more the objects in D_1 have no less memberships to D_2 than D_1. This makes $|C_2|$ close to $|C_1|$, where $C_1 = \{X_i | u_{2,i} \leq u_{1,i}, X_i \in X\}$, and $C_2 = \{X_i | u_{1,i} \leq u_{2,i}, X_i \in X\}$. In this case, the "uniform effect" will occur.

When taking the limits of (4) and (5) as m approaches 1 [5], we obtain

$$\lim_{m \to 1} \left\{ u_{l,i} = 1 \Big/ \sum_{h=1}^{k} \left[\frac{\|X_i - \hat{V}_l\|^2}{\|X_i - \hat{V}_h\|^2} \right]^{1/(m-1)} \right\} = \begin{cases} 1, & \|X_i - \hat{V}_l\|^2 \leq \|X_i - \hat{V}_h\|^2, 1 \leq h \leq k \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

for $1 \leq i \leq n$, $1 \leq l \leq k$ and

$$\lim_{m \to 1} \left\{ v_{l,j} = \frac{\sum_{i=1}^{n} (u_{l,i})^m x_{i,j}}{\sum_{i=1}^{n} (u_{l,i})^m} \right\} = \frac{\sum_{X_i \in C_l} x_{i,j}}{n_l} \quad (13)$$

for $1 \leq j \leq d$, $1 \leq l \leq k$. According to (12) and (13), we see that when m approaches 1, the fuzzy k-means algorithm becomes the hard k-means algorithm. In this case, $\sum_{i=1, X_i \in D_1}^{n} (u_{2,i})^m$ is equal to zero. This means that the objects in D_1 do not take a part in computing V_2 when the hard k-means algorithm is implemented. Then, $\|V_1 - V_2\|^2$ and $|\|X_i - V_1\|^2 - \|X_i - V_2\|^2|$ do not necessarily become small as the number of the objects in D_1 increases. If $\|X_i - V_1\|^2 < \|X_i - V_2\|^2$ for each $X_i \in D_1$ and $\|X_i - V_1\|^2 > \|X_i - V_2\|^2$

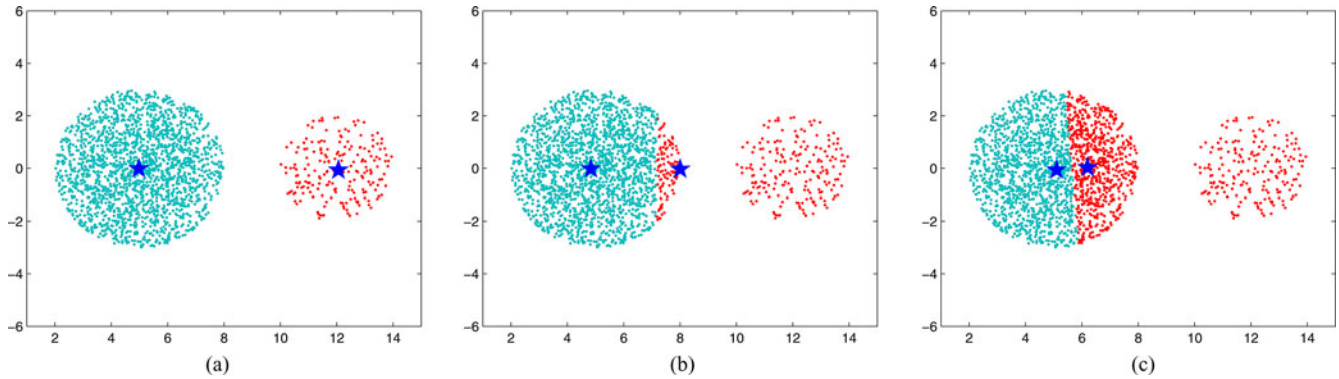


Fig. 4. "Uniform effect" occurs when the fuzzy k-means algorithm is implemented. (a) Imbalanced data distribution before clustering. (b) Clustering result in the first iteration. (c) Clustering result in the last iteration.

for each $X_i \in D_2$, the "uniform effect" does not occur in the hard k-means clustering process. This indicates that the characteristic of the data imbalanced distribution does not necessarily result in the occurrence of the "uniform effect," when the hard k-means clustering algorithm is implemented. There are other factors that contribute to the occurrence of the "uniform effect." Fig. 3 shows an example where the "uniform effect" does not occur when the hard k-means algorithm with the "true" cluster centers was used to cluster an imbalanced dataset. In this example, there are two clusters: a majority cluster (2000 objects) and a minority cluster (200 objects). Fig. 4 shows that when the fuzzy k-means algorithm with the same cluster centers is used to cluster the imbalanced dataset in Fig. 3, the "uniform effect" occurs.

When taking the limits of (4) and (5) as m approaches infinity [5], we obtain

$$\lim_{m\to+\infty}\left\{u_{l,i}=1\Big/\sum_{h=1}^{k}\left[\frac{\|X_i-\hat{V}_l\|^2}{\|X_i-\hat{V}_h\|^2}\right]^{1/(m-1)}\right\}=\frac{1}{k} \quad (14)$$

for $1 \le i \le n$, $1 \le l \le k$ and

$$\lim_{m\to+\infty}\left\{v_{l,j}=\frac{\sum_{i=1}^{n}(u_{l,i})^m x_{i,j}}{\sum_{i=1}^{n}(u_{l,i})^m}\right\}=\frac{\sum_{i=1}^{n}x_{i,j}}{n} \quad (15)$$

for $1 \le j \le d$, $1 \le l \le k$. According to (14) and (15), we see that when m approaches infinity, the membership degrees of each object to all the clusters approach the same value, i.e., $1/k$ and all the cluster centers become the same as each other. This results in $|C_2| = |C_1|$, where $C_1 = \{X_i | u_{2,i} \le u_{1,i}, X_i \in X\}$, and $C_2 = \{X_i | u_{1,i} \le u_{2,i}, X_i \in X\}$. This indicates that the "uniform effect" becomes evident as m increases.

IV. Multicenters Clustering Algorithm

In this section, we will propose a clustering algorithm to effectively cluster imbalanced data and avoid the "uniform effect." In the proposed algorithm, we use multicenters to represent each cluster, instead of one single center (shown in Fig. 5), because multicenters can help divide the objects of a majority class into several subgroups with small sizes similar to those of the minority classes. This can rebalance the scales of the majority classes and the minority classes to reduce the effect of imbalanced data distributions on the performance of the k-means-type algorithms.

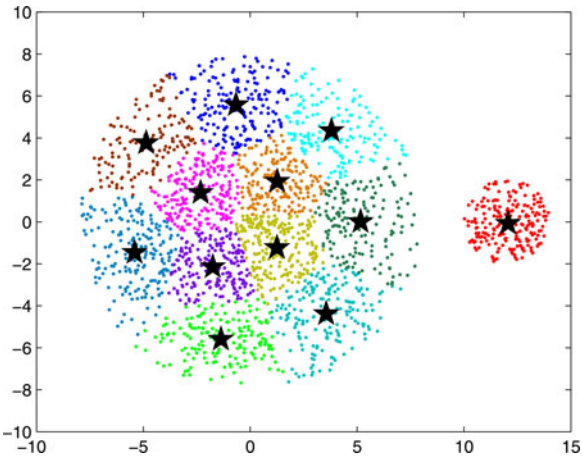


Fig. 5. Multicenters are used to represent each cluster.

The MC clustering algorithm needs to address the following three problems.

1) How do we obtain a set of cluster centers from the dataset?
2) How do we determine the number of clusters?
3) How do we judge which cluster centers represent the same clusters?

In the following, we separately investigate each of the aforementioned problems that are involved in the multicenter clustering procedure.

A. Production of Cluster Centers

The k-means-type algorithms perform iteratively the partition step and new cluster center generation step until convergence [1]. It is noted that they face the local minimum problem. That is, the clustering result guarantees a local minimum solution only. These algorithms are very sensitive to the initial cluster centers. For simplicity, users often use the random initialization method to obtain an initial set of cluster centers. However, these clustering algorithms need to rerun many times with different initializations in an attempt to find an optimal solution. In addition, the random initialization method works well only

when data are balanced and chances are good that at least one random initialization is close to a good solution. Therefore, how to select initial cluster centers is extremely important since they have a direct impact on the formation of the final clusters.

Recently, several attempts have been reported to solve the cluster center initialization problem [25]–[29]. Among them, the fast global k-means clustering algorithm (FGKM), which was proposed by Likas [27], is an effective method to solve the local minimum problem. The algorithm proceeds in an incremental way: To solve a clustering problem with h clusters, all intermediate problems with 1,2, $\ldots, h-1$ clusters are sequentially solved. The basic idea underlying the proposed method is that an optimal solution for a clustering problem with h clusters can be obtained using a series of local searches (using the hard k-means algorithm). At each local search, the $h-1$ cluster centers are always initially placed at their optimal positions corresponding to the clustering problem with $h-1$ clusters. The remaining hth cluster center is initially placed at several positions within the data space. Since, for $h=1$, the optimal solution is known, we can iteratively apply the aforementioned procedure to find optimal solutions for all k-clustering problems: $k=1,2,\ldots,h$. Such an approach leads at least to a near global minimizer. To reduce the computational complexity of FGKM, some modified FGKM algorithms have been proposed in [28] and [29]. A detailed description of the FGKM algorithms can be found in [27]. The results of numerical experiments have shown that the FGKM algorithm is able to locate either a global minimizer or a local minimizer close to global one. However, the FGKM algorithm cannot be used for fuzzy clustering. Therefore, we will propose the FGFKM algorithm to solve the local minimum problem of the fuzzy k-means algorithm. When $m>1$, the objective function (1) can be reduced by (4) as follows:

$$\begin{aligned}\mathbb{f}(V) &= \sum_{i=1,\forall V_l\in V,\|X_i-V_l\|^2\neq 0}^{n} 1/B(X_i,V)^{m-1} \\ &\quad + \sum_{i=1,\exists V_l\in V,\|X_i-V_l\|^2=0}^{n} B(X_i,V) \\ &= \sum_{i=1,\forall V_l\in V,\|X_i-V_l\|^2\neq 0}^{n} 1/B(X_i,V)^{m-1} \qquad (16)\end{aligned}$$

where

$$B(X_i,V)=\begin{cases}\sum_{l=1}^{|V|}\left[1/\|X_i-V_l\|^2\right]^{1/(m-1)} \\ \qquad\qquad \forall V_l\in V,\|X_i-V_l\|^2\neq 0 \\ 0, \qquad \exists V_l\in V,\|X_i-V_l\|^2=0.\end{cases} \qquad (17)$$

Property 1: If $\exists X_i\in X$ and $B(X_i,V)\neq 0$, then $\mathbb{f}(V)>\mathbb{f}(V\cup\{V_q\})$ for any $V_q\in R^d$.

Proof: Given V, we add a new cluster center V_q to V and obtain

$$B(X_i,V\cup\{V_q\})=\begin{cases}B(X_i,V)+\left[1/\|X_i-V_q\|^2\right]^{1/(m-1)} \\ \text{if } B(X_i,V)\neq 0 \text{ and } \|X_i-V_q\|^2\neq 0 \\ 0, \text{if } B(X_i,V)=0 \text{ or } \|X_i-V_q\|^2=0.\end{cases} \qquad (18)$$

By (17) and (18), we find the following properties.

1) If $B(X_i,V)\neq 0$ and $\|X_i-V_q\|^2\neq 0$, then $B(X_i,V\cup\{V_q\})>B(X_i,V)$, i.e., $1/B(X_i,V)^{m-1}>1/B(X_i,V\cup\{V_q\})^{m-1}$. In this case, the value of the function $\mathbb{f}$ decreases.
2) If $B(X_i,V)\neq 0$ and $\|X_i-V_q\|^2=0$, then $B(X_i,V)>B(X_i,V\cup V_q)=0$, i.e., $1/B(X_i,V)^{m-1}>B(X_i,V\cup\{V_q\})$. In this case, the value of the function $\mathbb{f}$ decreases.
3) If $B(X_i,V)=0$, then $B(X_i,V)=B(X_i,V\cup\{V_q\})=0$. In this case, the value of the function $\mathbb{f}$ does not change.

According to the aforementioned analysis, we know that if $\exists X_i\in X$ and $B(X_i,V)\neq 0$, the value of the function $\mathbb{f}$ decreases. Hence, the result follows. ■

If $B(X_i,V)=0$ for each object $X_i\in X$, clustering such a dataset is meaningless. Therefore, without loss of generality, we think that after a new cluster center V_q is added, the value of the function $\mathbb{f}$ will decrease. Moreover, the membership degrees of each object to the first h fuzzy clusters have been changed.

Property 2: $u_{l,i}(V)\geq u_{l,i}(V\cup\{V_q\})$, for $1\leq i\leq n$, where $u_{l,i}(V)$ is a membership degree of X_i to the lth fuzzy cluster when V is as the set of cluster centers.

Proof: According to (4), we can obtain

$$u_{l,i}(V)=\frac{[1/\|X_i-V_l\|^2]^{1/m-1}}{B(X_i,V)}, \text{ if } \forall V_l\in V,\|X_i-V_l\|^2\neq 0. \qquad (19)$$

By (18), we find that if $\|X_i-V_l\|^2\neq 0$, then $B(X_i,V\cup\{V_q\})>B(X_i,V)$, i.e., $u_{l,i}(V)\geq u_{l,i}(V\cup\{V_q\})$. If $\|X_i-V_q\|^2=0$, then $u_{|V|+1,i}(V\cup\{V_q\})=1$ and $u_{l,i}(V\cup\{V_q\})=0\leq u_{l,i}(V)$ for $1\leq l\leq|V|$. Hence, the result follows. ■

Property 3: $C_l(V)\supseteq C_l(V\cup\{V_q\})$, for $1\leq i\leq n, 1\leq l\leq|V|$, where $C_l(V)$ is the lth fuzzy cluster that is obtained by using V as the set of cluster centers to fuzzily partition X.

Proof: By Property 2, the result can be easily obtained. ■

Properties 1, 2, and 3 state that after a new cluster center is added, it can obtain some objects from the existed clusters to form a new cluster, which makes the value of the function $\mathbb{f}$ decrease. This means that while we add an object to V, the more the value of the function $\mathbb{f}$ decreases, the more reliable the object is as the $(h+1)$th cluster center. Therefore, an incremental algorithm is used to obtain the k cluster centers. This algorithm starts with one cluster center $V_1=\sum_{i=1}^{n}X_i/n$ and attempts to optimally add one object as a new cluster center at each stage. The selected object $X_{j'}$ as the $(h+1)$th cluster center can minimize the function $\mathbb{f}$. After the object $X_{j'}$ is found, the FGFKM algorithm with the initial cluster centers $V=V\cup\{X_{j'}\}$ is used to fuzzily partition the dataset into $h+1$ fuzzy clusters. The result set of cluster centers will be used as the initial cluster centers in the next stage. The algorithm is presented as follows.

Fast Global Fuzzy K-Means Algorithm

Step 1. Input the parameter m and the matrix $D = [D_{i,j}]$, where $D_{i,j} = \|X_i - X_j\|^2$ for $1 \le i \le j \le n$. Compute $V = \{V_1\}$ from the dataset X where $V_1 = \sum_{i=1}^{n} X_i/n$, and compute $B(X_i, V)$ for $1 \le i \le n$, according to (17). Set $h = 1$.

Step 2. Find the object $X_{j'} \in X$ which satisfies

$$\begin{aligned} X_{j'} &= \text{argmin}_{j=1}^{n} \mathbb{f}(V \cup \{X_j\}) \\ &= \text{argmin}_{j=1}^{n} \sum_{i=1, D_{ij} \neq 0, B(X_i,V) \neq 0}^{n} \\ &\quad \times \frac{1}{\left[B(X_i, V) + D_{ij}^{-1/m-1}\right]^{m-1}}. \end{aligned} \quad (20)$$

Set $V = V \cup \{X_{j'}\}$.

Step 3. Use V as the set of initial cluster centers and apply the fuzzy k-means algorithm to partition X into $h+1$ clusters and obtain the result set of cluster centers $V^* = \{V_1^*, V_2^*, \ldots, V_{h+1}^*\}$. According to (17), update $B(X_i, V^*)$ for $1 \le i \le n$. Set $V = V^*$.

Step 4. Set $h = h + 1$. If $h < k$, go to Step 2; otherwise, stop.

Before the implementation of the FGFKM algorithm, we need to compute D, which is $O(n^2)$. Although this operation is expensive, D will only need to be precalculated for once in the proposed multicenters algorithm. In the FGFKM algorithm, adding an object as the $(h+1)$th cluster center needs $O(n)$ calculations. After a new cluster center is added, using the fuzzy k-means algorithm to calculate the new cluster centers for the FGFKM algorithm requires $O(nkt)$ calculations, where t is the number of iterations that are performed by the fuzzy k-means algorithm. Therefore, the computational cost of the FGFKM algorithm to generate a set of $k_{\max}$ cluster centers is $O((n + nk_{\max}t)k_{\max})$. Here, $k_{\max} \ll n$ is the number of cluster centers that we hope to obtain.

B. Determination of the Number of Clusters

After obtaining the $k_{\max}$ cluster centers by using the FGFKM algorithm, we need to determine the number of clusters k. Traditionally, cluster analysis uses statistical validity indices that are based on the within and between-cluster information to validate the clustering results. A typical index curve consists of the index values for different k number of clusters. Those ks at the peaks, valleys, or distinguishing "knees" on the index curve are regarded as candidates of the optimal number of clusters (the best k); see, for instance, [42]. However, for imbalanced data, the traditional validity indices cannot effectively evaluate the clustering results because the distances between some points in the majority class may be larger than the distances between some points in different classes. Therefore, we will propose a new method to evaluate the number of clusters for imbalanced data.

According to the limiting properties of (4) and (5), we know that the closer the value of m is to 1, it is more likely that each object is assigned to a single cluster, whereas the larger the value of m, the more clusters to which each object is assigned. This makes the cluster centers which represent the same clusters move to the same locations and form several rallying points. As m increases to a certain level, the number of rallying points of the cluster centers is the same as the number of the "true" clusters in the dataset. This indicates that the value of m can help us to find the "true" number of clusters. However, as m increases even further, the number of rallying points becomes smaller than the number of the true clusters in the dataset. Finally, the number of rallying points becomes 1.

Let us consider the following example to demonstrate the aforementioned properties. Fig. 6 shows the movements of the cluster centers that are obtained by the FGFKM algorithm with different values of the fuzzy index m. In this example, we stimulate three clusters which are marked by different colors: green (2000 objects), purple (400 objects), and red (200 objects), respectively. When m increases to 2, we see that the 13 cluster centers move to three locations. The number of the rallying points is just equal to the number of the "true" clusters. However, as m continues to grow, some cluster centers get away from their rallying points to the global center of the dataset. When m is equal to 2.5, all the cluster centers are moved to the global center.

The previous analysis states us that the determination of the number of clusters needs to solve the following subproblems.

1) How do we obtain an appropriate value of m?
2) Given m, how do we find the number of rallying points of cluster centers?

To choose an appropriate fuzzy index m is very important when implementing the fuzzy k-means algorithm. Up to date, theoretical and empirical results on the study of setting the fuzzy index m have been obtained [20]–[23]. In 1976, a physical interpretation of the fuzzy k-means algorithm when $m = 2$ was given in [20]. Based on the performance of some cluster validity indices, Pal and Bezdek [4] have given heuristic guidelines regarding the best choice for m, suggesting that it is probably in the $[1.5, 2.5]$ interval. Similar recommendations have appeared in [21] and [22]. Yu gave some theoretical rules to select the fuzzy index m in [23]. Most researchers have also proposed $m = 2$. In our proposed algorithm, we will select an appropriate value of m from the $[1.1, 2.5]$ interval to determine the number of clusters.

The example in Fig. 6 indicates that the changes of m will cause the changes of the distances between the cluster centers which represent the same clusters and the changes of the numbers of neighbors of the cluster centers. In the following, we will analyze these changes using three cases. 1) As m increases from 1.1, the cluster centers which represent the same clusters move to the same stations, which makes the distances between them decrease and the numbers of their neighbors increase. 2) As m continues to grow, some of them get away from others to the global center, which makes some of their distances increase and the numbers of neighbors of some cluster centers decrease; 3) As m increases to a certain value, all of them move to the global center, which makes their distances decrease and the numbers of their neighbors increase. From these cases, we see that the optimum value of m should be a boundary value between the first case and the second case. Next, we will find the boundary value m by examining the changes of their neighbors

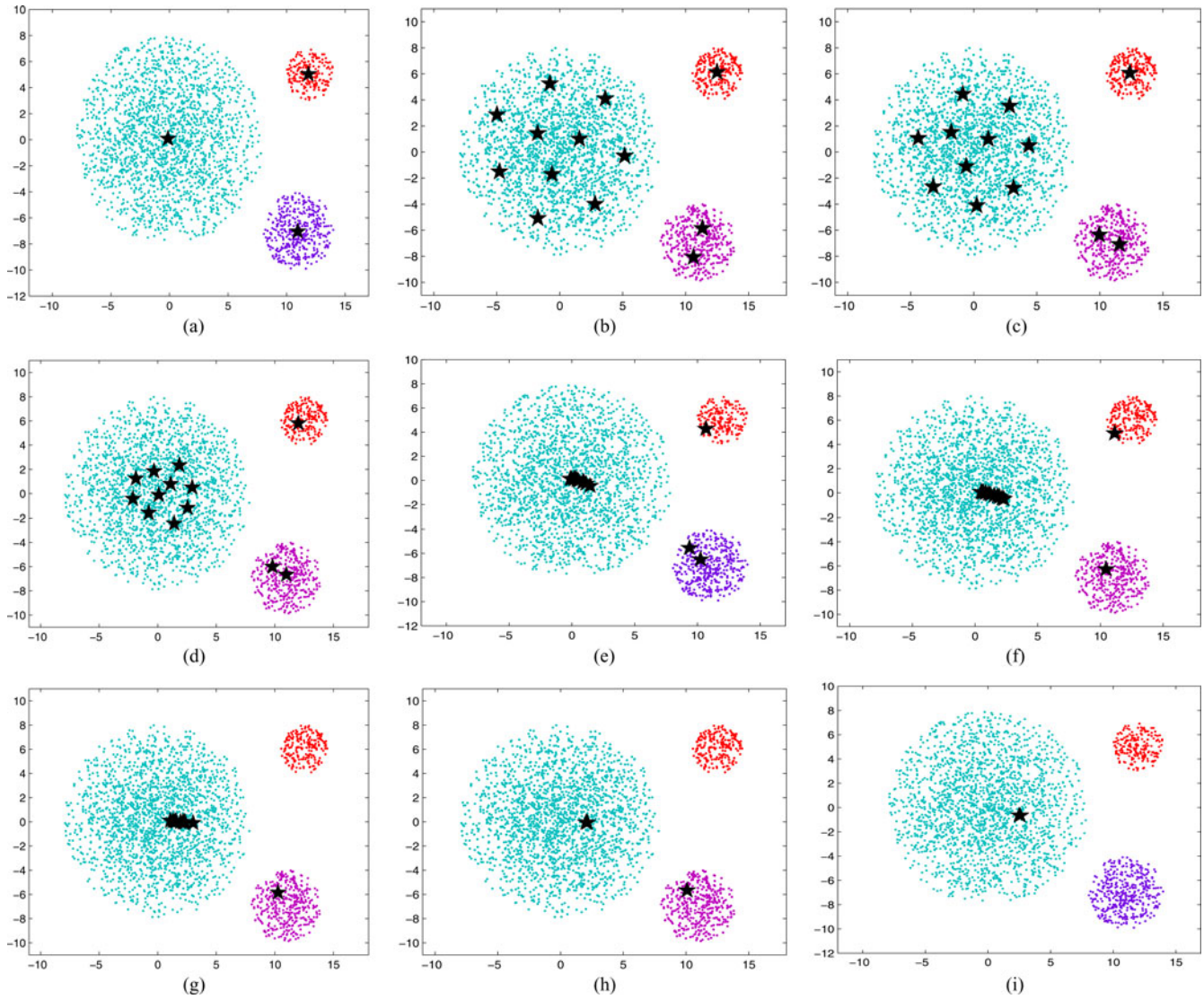


Fig. 6. (a) Data distribution on the dataset. (b) Cluster centers obtained by the FGFKM algorithm with $m = 1.1$. (c) Cluster centers obtained by the FGFKM algorithm with $m = 1.5$. (d) Cluster centers obtained by FGFKM with $m = 1.8$. (e) Cluster centers obtained by FGFKM with $m = 2.0$. (f) Cluster centers obtained by FGFKM with $m = 2.1$. (g) Cluster centers obtained by FGFKM with $m = 2.2$. (h) Cluster centers obtained by FGFKM with $m = 2.3$. (i) Cluster centers obtained by FGFKM with $m = 2.6$.

from the first case to the second case. The neighborhood of a cluster center is defined as follows.

Definition 1 (Neighborhood of a Cluster Center): Let $V = \{V_1, V_2, \ldots, V_{k_{\max}}\}$ be a result set of cluster centers obtained by the FGFKM algorithm. For arbitrary $V_i \in V$, the neighborhood $\delta(V_i)$ of V_i is defined as

$$\delta(V_i) = \{V_j | V_j \in V, \|V_j - V_i\|^2 \leq \theta\} \tag{21}$$

where θ is a threshold.

The size of the neighborhood depends on the threshold θ. More cluster centers fall into the neighborhood of V_i if θ takes a great value. Therefore, the threshold θ is a key factor for the neighborhood. We set θ as the minimal distance between the cluster centers that is obtained by the FGFKM algorithm with $m = 1.1$, i.e.,

$$\theta = \min_{1 \leq i < j \leq k_{\max}} \|V_i - V_j\|^2. \tag{22}$$

The reason to set such threshold θ is given next. We select m in the interval $[1.1, 2.5]$, which means that 1.1 is the minimal value proposed by us. When $m = 1.1$, the distances between the cluster centers that are obtained by the FGFKM algorithm should be large, and the number of their neighbors should be small, compared with other values of m in the given interval.

Next, we will propose the BMP algorithm to conveniently capture the dramatic difference between the numbers of neighbors of the cluster centers.

Best M-Plot Algorithm

Step 1. Input m_{start}, m_{end}, λ, and $k_{\max}$. Set $m_{\text{tmp}} = m_{\text{start}}$.

Step 2. Obtain a resulting set V of the cluster centers by the FGFKM algorithm with $m = m_{\text{start}}$. Then, compute the threshold θ and $|\delta(V_i)|$ for each cluster center $V_i \in V$ to store by ascending order into a queue $Q_{m_{\text{tmp}}} = [Q_{m_{\text{tmp}}}(1), Q_{m_{\text{tmp}}}(2), \ldots, Q_{m_{\text{tmp}}}(k_{\max})]$.

TABLE I
QUEUES Q_m WITH DIFFERENT m

m	$Q_m(1)$	$Q_m(2)$	$Q_m(3)$	$Q_m(4)$	$Q_m(5)$	$Q_m(6)$	$Q_m(7)$	$Q_m(8)$	$Q_m(9)$	$Q_m(10)$	$Q_m(11)$	$Q_m(12)$	$Q_m(13)$
1.1	1	1	1	1	1	1	1	1	1	1	1	2	2
1.2	1	1	1	1	1	1	1	1	1	1	1	2	2
...							...						
1.5	1	1	1	1	1	1	1	1	1	1	1	2	2
1.6	1	1	1	1	1	1	1	1	1	2	2	2	2
1.7	1	1	1	2	2	2	2	2	2	3	3	4	4
1.8	1	2	2	2	3	3	3	4	4	4	5	5	5
1.9	1	2	2	4	5	5	5	5	6	7	7	8	8
2.0	1	2	2	10	10	10	10	10	10	10	10	10	10
2.1	1	1	11	11	11	11	11	11	11	11	11	11	11
2.2	1	12	12	12	12	12	12	12	12	12	12	12	12
...							...						
2.5	1	12	12	12	12	12	12	12	12	12	12	12	12
2.6	13	13	13	13	13	13	13	13	13	13	13	13	13

Step 3. If $m_{\text{tmp}} + \lambda > m_{\text{end}}$, then put out m_{tmp}, and stop; otherwise, apply the FGFKM algorithm with $m = m_{\text{tmp}} + \lambda$ to obtain a resulting set V of the cluster centers. Compute $|\delta(V_i)|$ for each cluster center $V_i \in V$ to store by ascending order into a queue $Q_{m_{\text{tmp}}+\lambda} = [Q_{m_{\text{tmp}}+\lambda}(1), Q_{m_{\text{tmp}}+\lambda}(2), \ldots, Q_{m_{\text{tmp}}+\lambda}(k_{\max})]$.

Step 4. For $1 \le i \le k_{\max}$, if $Q_{m_{\text{tmp}}}(i) > Q_{m_{\text{tmp}}+\lambda}(i)$, then put out m_{tmp}, and stop.

Step 5. Set $m_{\text{tmp}} = m_{\text{tmp}} + \lambda$, and go to Step 3.

In this algorithm, the value of m increases from m_{start} to m_{end} with the step length of λ. In this paper, we propose to set $m_{\text{start}} = 1.1$, $m_{\text{end}} = 2.5$, and $\lambda = 0.1$. The computational cost of the BMP algorithm is $O([(n + nk_{\max}t)k_{\max} + k_{\max}^2]\beta)$, where $\beta = (m_{\text{end}} - m_{\text{start}})/\lambda$. Here, the $O(k_{\max}^2)$ operations are used to calculate $Q_{m_{\text{tmp}}}$, while the value of m_{tmp} is given.

Let us reconsider the example in Fig. 6. Table I shows the changes of the queues Q_m with different $m \in [1.1, 2.6]$. We see that from $m = 1.1$ to 2.0, the values of Q_m increase gradually. However, from $m = 2.0$ to 2.1, $Q_m(2)$ decreases. This means that the boundary value of m is 2. We will use the cluster centers that are obtained by the FGFKM algorithm with $m = 2$ to determine the number of clusters in the given dataset.

The second subproblem can be solved by using the Max–Min distances between the cluster centers. We first construct a descending sequence of cluster centers and define a possibility function of an existing rallying point. Furthermore, we will find the number of rallying points of the cluster centers, which is used to determine the number of clusters.

Definition 2 (Descending Sequence of Cluster Centers): Let $V = \{V_1, V_2, \ldots, V_{k_{\max}}\}$ be a result set of cluster centers obtained by the FGFKM algorithm. We construct a sequence $S = \{V_1', V_2', \ldots, V_{k_{\max}}'\}$ of the cluster centers in V, such that

$$\begin{cases} \|V_l' - V_{l+1}'\|^2 = \max\limits_{1 \le i < j \le k_{\max}} \|V_i - V_j\|^2, & \text{if } l = 1 \\ \min\limits_{j=1}^{l-1} \|V_l' - V_j'\|^2 = \max\limits_{h=1}^{k_{\max}} \min\limits_{j=1}^{l-1} \|V_h - V_j'\|^2, & \text{if } l > 1 \end{cases} \tag{23}$$

for $1 \le l \le k_{\max}$.

Definition 3 (Possibility Function of an Existing Rallying Point): The possibility of the existing lth rallying point is defined as

$$P(l) = \begin{cases} \min\limits_{j=1}^{l-1} \|V_l' - V_j'\|^2, & \text{if } l > 1 \\ P(2), & \text{if } l = 1 \end{cases} \tag{24}$$

for $1 \le l \le k_{\max}$.

Intuitively, if there are k rallying points in the obtained cluster centers, the distances between these rallying points should be large. Hence, we could select the first k cluster centers which are the farthest from each other to be as rallying points. Since the $k+1$ rallying point does not exist, the $(k+1)$th cluster center which is selected as the $(k+1)$th rallying point will be very close to one of the first k cluster centers. This means that $P(k+1)$ is far smaller than $P(l), 1 \le l \le k$. The values of the function P from k to $k+1$ show a dramatic change. At the same time, the values of the function P from $k+1$ to $k_{\max}$ should be much less distinctive, because these chosen $k+2, k+3, \ldots, k_{\max}$ cluster centers will also be very close to one of the first k initial cluster centers. This heuristic states us that the value of $P(l)$ could reflect the possibility of the existing lth rallying point. The higher the value, the more possibly the lth rallying point exists. While we select the $(k+1)$th cluster center as a rallying point, the function P from $k+1$ goes into a plateau. This means that $k+1$ should be a knee point on the function P. However, since the number of clusters is not usually less than 2, the values of the function P from $k = 1$ to 2 should not show a dramatic change; otherwise, it is meaningless. Therefore, we set $P(1) = P(2)$ in the definition of the function P. We will determine the number of clusters by analyzing the function P to find a knee point. Fig. 7 shows a curve of the function P of the example in Fig. 6. According to Fig. 7, we can see that the determined number of clusters is 3 since $k = 4$ is a knee point. After the cluster centers are obtained, the computation complexity to find the number of clusters is $O(k_{\max}^2)$.

C. Grouping Multicenters to Represent Each Cluster

In this section, we will first propose a separation measure to evaluate how well two subclusters are separated. Next, we will present the GMC algorithm in which the separation measure is used to group multicenters to represent each cluster.

A separation measure is used to reflect how well two clusters are separated. Conceptually, a large separation of two clusters indicates less of an inclination to integrate these clusters

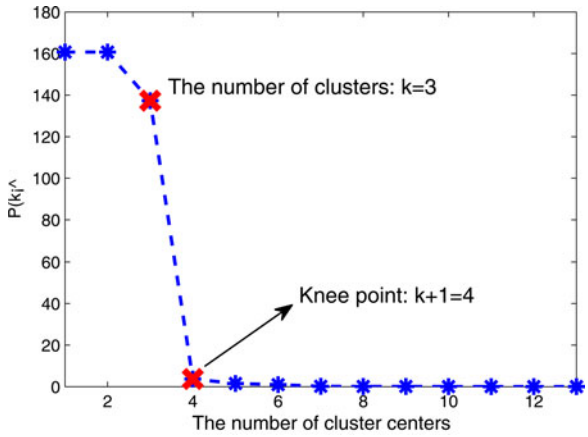


Fig. 7. Sketch of the function $P(k)$.

into a larger one. It is also called cluster distance or similarity in the literature [39], [43]–[45]. The distances between the closest or farthest data points of two clusters are used to measure the cluster separation in the agglomerative clustering algorithms [43], [44]. They are not only computationally expensive but sensitive to noise as well because of the dependence on a few points. In the center-based clustering algorithms, the separation of two clusters is often measured by using the distance between their centers. Although this measure is computationally efficient and robust to noise, it cannot reflect the degree of overlap between two clusters. For example, Fig. 8 shows that for the same distance between two clusters, the separation of two clusters can be different. Intuitively, we see that Clusters A and B are less separated than Clusters B and C. This is because there is a clear boundary between Clusters B and C, compared with that between A and B. However, with the use of the separation measure, the separation of Clusters A and B is equal to that of Clusters B and C. In [40], Liu *et al*. proposed to find a sparse region between two clusters and count data points falling into the region. The number of data points in the region is used to reflect the separation of the two clusters. The question is how to find the sparse region between the two clusters. Liu *et al*. first project data points from two clusters onto the line that connects the two centers. Next, they partition the line into $2B$ bins of equal size and count the data points that fall into each bin. The bin that has the smallest number of data points is selected as the sparse region. However, there are following problems. 1) The value of B has an impact on the number of data points that fall into a bin. It is difficult to select an appropriate B value when the domain knowledge of the datasets is not available. 2) In addition to B, a bin size depends on the distance between two centers. Given B, if the distances between two pairs of centers are different, the two obtained sparse regions have different sizes.

To overcome these shortcomings, we use the degree of overlap between two clusters to reflect their separation. Given two clusters, the more the data objects have similar memberships to them, the larger their overlapping degree. When the overlapping degree is large, the boundary between the two clusters is not clear. This means that their separation is poor. When the overlapping degree is very small so that each data object is clearly assigned to only one cluster, the separation of the two clusters is very large. We will define the separation measure between two clusters as follows.

Definition 4 (Separation Measure): Let $C = \{C_1, C_2, \ldots, C_{k_{\max}}\}$ be a partition of X and $V = \{V_1, V_2, \ldots, V_{k_{\max}}\}$, where V_l is the center of C_l for $1 \le l \le k_{\max}$. The separation measure between C_l and C_h is defined as

$$S(C_l, C_h) = \begin{cases} \dfrac{1}{2|C_l'|}\displaystyle\sum_{X_{j'} \in C_l'} \dfrac{\big|\|X_{j'} - V_l\|^2 - \|X_{j'} - V_h\|^2\big|}{\|X_{j'} - V_l\|^2 + \|X_{j'} - V_h\|^2} \\ \qquad + \dfrac{1}{2|C_h'|}\displaystyle\sum_{X_{j'} \in C_h'} \dfrac{\big|\|X_{j'} - V_l\|^2 - \|X_{j'} - V_h\|^2\big|}{\|X_{j'} - V_l\|^2 + \|X_{j'} - V_h\|^2} \\ \qquad\qquad \text{if } |C_l'| \neq 0 \text{ and } |C_h'| \neq 0 \\ \dfrac{1}{2} + \dfrac{1}{2|C_l'|}\displaystyle\sum_{X_{j'} \in C_l'} \dfrac{\big|\|X_{j'} - V_l\|^2 - \|X_{j'} - V_h\|^2\big|}{\|X_{j'} - V_l\|^2 + \|X_{j'} - V_h\|^2} \\ \qquad\qquad \text{if } |C_l'| \neq 0 \text{ and } |C_h'| = 0 \\ \dfrac{1}{2} + \dfrac{1}{2|C_h'|}\displaystyle\sum_{X_{j'} \in C_h'} \dfrac{\big|\|X_{j'} - V_l\|^2 - \|X_{j'} - V_h\|^2\big|}{\|X_{j'} - V_l\|^2 + \|X_{j'} - V_h\|^2} \\ \qquad\qquad \text{if } |C_l'| = 0 \text{ and } |C_h'| \neq 0 \\ 1, \qquad\qquad \text{if } |C_l'| = 0 \text{ and } |C_h'| = 0 \end{cases} \tag{25}$$

for $1 \le l < h \le k$, where

$$C_l' = \left\{ X_{j'} \Big| \frac{\|V_l - V_h\|^2 + \|V_l - X_{j'}\|^2 - \|V_h - X_{j'}\|^2}{2\sqrt{\|V_l - X_{j'}\|^2}\sqrt{\|V_l - V_h\|^2}} \ge 0 \right. \\ \left. X_{j'} \in C_l \right\},$$

and

$$C_h' = \left\{ X_{j'} \Big| \frac{\|V_l - V_h\|^2 + \|V_h - X_{j'}\|^2 - \|V_l - X_{j'}\|^2}{2\sqrt{\|V_h - X_{j'}\|^2}\sqrt{\|V_l - V_h\|^2}} \ge 0 \right. \\ \left. X_{j'} \in C_h \right\}.$$

In the definition, we use the distances between objects and centers to measure the degree of overlap between clusters, instead of memberships of objects to clusters. According to (4), we obtain that the difference between the memberships of $X_{j'}$ to clusters C_l and C_h is

$$\|u_{l,j'} - u_{h,j'}\| = \frac{\left| \left(\frac{1}{\|X_{j'} - V_l\|^2}\right)^{1/m-1} - \left(\frac{1}{\|X_{j'} - V_h\|^2}\right)^{1/m-1} \right|}{\left(\frac{1}{\|X_{j'} - V_l\|^2}\right)^{1/m-1} + \left(\frac{1}{\|X_{j'} - V_h\|^2}\right)^{1/m-1} + P} \tag{26}$$

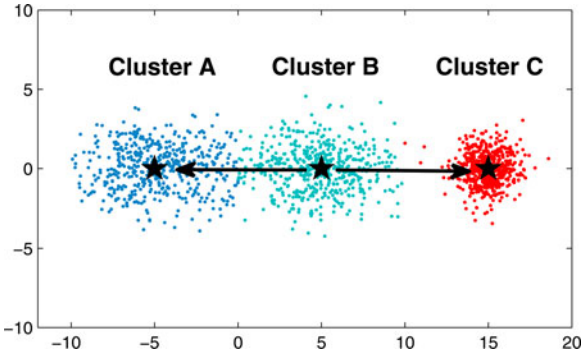


Fig. 8. Distances between the centers of three clusters.

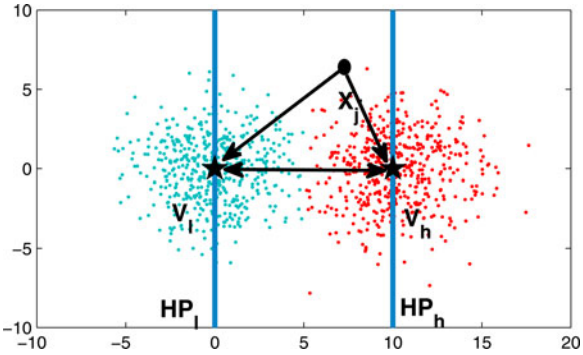


Fig. 9. Overlapping region between two clusters.

where

$$P = \sum_{1\le t\le k_{\max}, t\neq l, t\neq h} \left(\frac{1}{\|X_{j'} - V_t\|^2}\right)^{1/m-1}.$$

We see that $\|u_{l,j'} - u_{h,j'}\|$ is affected by two factors: m and P. For each object in $C_l \cup C_h$, the closer the value of m is to 1, the larger the difference between its memberships to clusters C_l and C_h, which makes the degree of overlap between clusters computed by using the memberships approach 0. This overlapping measure does not reflect the separation between the two clusters. In terms of the effect of P, the larger the P, the smaller the difference between its memberships to clusters C_l and C_h, which weakens the degree of overlap between the two clusters. In addition, computing the separation measure between the two clusters should be independent of other clusters. Therefore, we remove m and P from (26) and obtain

$$\frac{\left|1/\|X_{j'} - V_l\|^2 - 1/\|X_{j'} - V_h\|^2\right|}{1/\|X_{j'} - V_l\|^2 + 1/\|X_{j'} - V_h\|^2} = \frac{\left|\|X_{j'} - V_l\|^2 - \|X_{j'} - V_h\|^2\right|}{\|X_{j'} - V_l\|^2 + \|X_{j'} - V_h\|^2} \quad (27)$$

which is used to measure the separation between the two clusters.

In this separation measure, we only take into account a subset of objects in C_l and C_h. Fig. 9 shows that the overlap between the clusters C_l and C_h should only emerge between the hyperplanes HP_l which is with the point V_l lying on and a normal vector $\overrightarrow{V_l V_h}$ and HP_h which is with the point V_h lying on and a normal vector $\overrightarrow{V_l V_h}$. Therefore, we select the objects between HP_l and HP_h from C_l and C_h to measure the separation between C_l and C_h. Next, we propose the GMC algorithm that is based on the separation measure, which is described as follows.

Grouping Multicenter Algorithm

Step 1. Input k, which is the number of clusters, $C = \{C_1, C_2, \ldots, C_{k_{\max}}\}$, which is a partition of X, and $V = \{V_1, V_2, \ldots, V_{k_{\max}}\}$, which is the set of cluster centers. Let $CP = \{CP_1, CP_2, \ldots, CP_{k_{\max}}\}$ be an initial partition of C, where $CP_i = \{C_i\}$ for $1 \le i \le k_{\max}$.

Step 2. If $|CP| \le k$, go to Step 3; otherwise, find CP_i and CP_j from CP, which are satisfied as

$$\min_{1\le i<j\le k_{\max}} \min_{C_{i'}\in CP_i, C_{j'}\in CP_j} S(C_{i'}, C_{j'}), \quad (28)$$

add $CP_i \cup CP_j$ to CP, and remove CP_i and CP_j. Go to Step 2.

Step 3. Construct a partition $VP = \{VP_1, VP_2, \ldots, VP_k\}$ of V, where $VP_i = \{V_{i'} | V_{i'} \in V$ is the center of $C_{i'} \in CP_i\}$, for $1 \le i \le k$. Output VP.

We apply the GMC algorithm to obtain a partition VP of V. For $1 \le i \le k$, all the cluster centers in VP_i are used to collectively represent the ith cluster. The computation complexity of the GMC algorithm is $O(nk_{\max} + k_{\max}^2 \log k_{\max})$.

D. Overall Implementation

The MC clustering algorithm is implemented under the framework that is shown in Fig. 10. This algorithm consists of the three subalgorithms: the FGFKM, BMP, and CMG algorithms. In the first phase, we apply the FGFKM algorithm to obtain $k_{\max}$ reliable cluster centers from a dataset. Before the implementation of the FGFKM algorithm, we need to input two parameters: the number of cluster centers $k_{\max} (\ge k)$ and the fuzzy index m. In clustering imbalanced data, if the fuzzy index m is an oversized value, the obtained cluster centers may have similar features, which makes the "uniform effect" to occur. However, if the m value is equal to 1, the algorithm cannot properly deal with noise and overlapping properties of clusters. Therefore, we suggest $m \in [1.1, 1.5]$, which could make each of these obtained cluster centers to have a great difference with each other and represent a different subset of objects. In the second phase, the BMP algorithm is used to find the most appropriate value m^* from the $[m_{\text{start}}, m_{\text{end}}]$ interval with the step length of λ and determine the number of clusters k. We suggest setting $m_{\text{start}} = m$, which is used in the first phase, $m_{\text{end}} = 2.5$, and $\lambda = 0.1$. In the last phase, we use the GMC algorithm to group the cluster centers that are obtained in the first phase to represent k clusters.

V. Experimental Results

In this section, we present three experiments to evaluate the effectiveness of the proposed algorithm. The first two experiments were conducted on synthetic datasets. The last

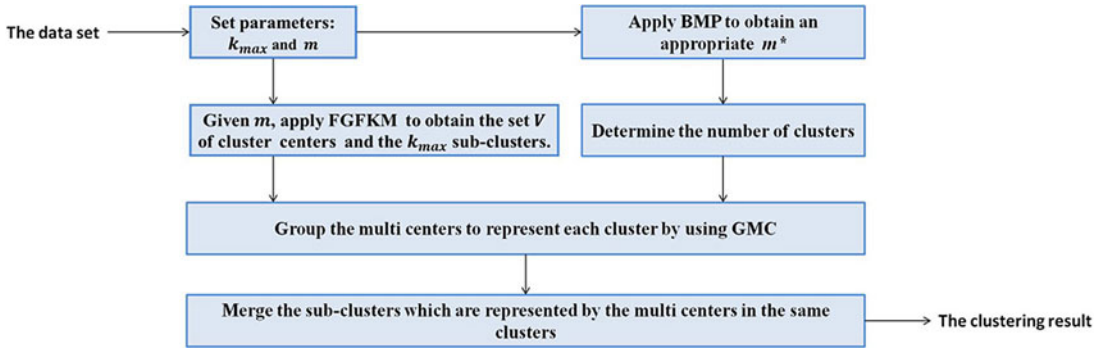


Fig. 10. Flowchart of the overall implementation of the MC clustering algorithm.

TABLE II
SUMMARY OF DATASETS

Data set	objects	attributes	clusters	Original Classes				
				P_1	P_2	P_3	P_4	P_5
BDS1	3000	2	3	1000	1000	1000	0	0
IDS1	2600	2	3	2000	400	200	0	0
IDS2	3200	2	5	2000	200	200	400	400
Wine	178	13	3	59	71	48	0	0
Breast Cancer	683	9	2	444	239	0	0	0
Ecoli	327	7	5	143	77	35	20	52

TABLE III
CLUSTER RECOVERY FOR THE IDS1 DATA BY USING SIX ALGORITHMS

Clusters found	HKM(B) Objects in cluster	HKM(B) Original classes P_1	P_2	P_3	FKM(B) Objects in cluster	FKM(B) Original classes P_1	P_2	P_3
C_1	1722	1722	0	0	1195	1195	0	0
C_2	438	38	400	0	696	696	0	0
C_3	440	240	0	200	709	109	400	200
Clusters found	GK(B) Objects in cluster	GK(B) Original classes P_1	P_2	P_3	KFKM(B) Objects in cluster	KFKM(B) Original classes P_1	P_2	P_3
C_1	1182	1182	0	0	1476	1476	0	0
C_2	475	75	400	0	489	89	400	0
C_3	943	743	0	200	635	435	0	200
Clusters found	ICCM Objects in cluster	ICCM Original classes P_1	P_2	P_3	MC Objects in cluster	MC Original classes P_1	P_2	P_3
C_1	1028	1028	0	0	1981	1981	0	0
C_2	954	954	0	0	412	12	400	0
C_3	618	18	400	200	207	7	0	200

experiment was conducted on three real datasets from the University of California at Irvine. [48].Detailed information including data size, the number of attributes, and class distribution can be found in Table II. In Experiments II and III, the proposed algorithm was compared with the hard k-means (KM) ($m = 1$), fuzzy k-means (FKM) ($m = 2$), kernel fuzzy k-means (KFKM) (the kernel function is the Gaussian kernel; $m = 2; \sigma^2 = 100$), Gustafson–Kessel (GK) ($m = 2$), and iterative compatible cluster merging algorithm (ICCM) ($m = 2, c_1 = 0.95, c_2 = 0.05, c_3 = 3$). In the comparisons, we performed the KM, FKM, KFKM, and GK algorithms with the 20 randomly selected cluster centers and the "true" cluster centers as the initial sets, respectively. In Tables III–IX, R denotes the average results of these algorithms with these randomly selected cluster centers. B denotes the results of these algorithms with the "true" cluster centers. For the ICCM algorithm, we presented its best clustering result on each dataset over 20 random runs in Tables III–IX.

To evaluate the performance of clustering algorithms in the experiments, we consider the five validity measures [33], [46], [47]: 1) accuracy (AC); 2) precision (PE); 3) recall (RE); 4) adjusted rand index (ARI); and 5) coefficient of variation (CV). Let X be a dataset, $C = \{C_1, C_2, \ldots, C_k\}$ be a clustering result of X, $P = \{P_1, P_2, \ldots, P_{k'}\}$ be a partition of the original classes in X, n_{ij} be the number of common objects of groups C_i and P_j: $n_{ij} = |C_i \cap P_j|$, b_i be the number of objects in C_i, and d_j be the number of objects in P_j. These validity measures are defined as

$$\mathrm{AC} = \frac{1}{n}\sum_{i=1}^{k} \max_{j=1}^{k'} n_{ij}, \mathrm{PE} = \frac{1}{k}\sum_{i=1}^{k} \frac{\max_{j=1}^{k'} n_{ij}}{b_i}$$

$$\mathrm{RE} = \frac{1}{k}\sum_{i=1}^{k} \frac{\max_{j=1}^{k'} n_{ij}}{d_{\mathrm{argmax}_{j=1}^{k'} n_{ij}}}$$

$$\mathrm{ARI} = \frac{\sum_{ij} C_{n_{ij}}^2 - [\sum_i C_{b_i}^2 \sum_j C{d_j}^2]/C_n^2}{\frac{1}{2}[\sum_i C_{b_i}^2 + \sum_j C_{d_j}^2] - [\sum_i C_{b_i}^2 \sum_j C_{d_j}^2]/C_n^2}$$

$$\mathrm{CV} = \left| \frac{\sqrt{\sum_{i=1}^{k}(b_i - \sum_{i=1}^{k} b_i/k)/(k-1)}}{\sum_{i=1}^{k} b_i/k} - \frac{\sqrt{\sum_{j=1}^{k'}(d_j - \sum_{j=1}^{k'} d_j/k')/(k'-1)}}{\sum_{j=1}^{k'} d_j/k'} \right|.$$

If the clustering result is close to the true class distribution, then the values of AC, PE, RE, and ARI are high. However, for imbalanced data, some of AC, PE, RE, and ARI tend to not capture the "uniform effect" and provided misleading information about the clustering performance. For example, if the "uniform effect" occurs, the clustering result tends to have a high value of AC but low values of the other indices. In this case, it is not enough to only consider AC to evaluate the clustering result. We do not deny AC. Conversely, we believe that the larger the value of AC, the better the clustering solution. We mean that other measures should be simultaneously considered. In addition, we employ the CV measure [33] which is a necessary criterion to validate the clustering results. Although this criterion does not necessarily indicate a good clustering performance if the CV value of the clustering result is low, it indicates that if the CV value is high, the clustering performance is poor.

TABLE IV
SUMMARY CLUSTERING RESULTS OF DIFFERENT ALGORITHMS ON THE IDS1 DATA

	HKM(R)	HKM(B)	FKM(R)	FKM(B)	GK(R)	GK(B)	KFKM(R)	KFKM(B)	ICCM	MC
AC	0.9028	0.9085	0.8371	0.8812	0.8581	0.8942	0.8454	0.8888	0.9162	0.9927
PE	0.8624	0.8196	0.8273	0.8547	0.8415	0.8767	0.8746	0.8343	0.8824	0.9790
RE	0.6579	0.6603	0.6292	0.6485	0.6283	0.6542	0.5580	0.6518	0.6637	0.9968
ARI	0.4764	0.5697	0.2364	0.3450	0.3104	0.3307	0.2984	0.4681	0.3898	0.9725
CV	0.5214	0.1665	1.0221	0.8102	0.7037	0.7234	0.3528	0.5237	0.4987	0.0186

A. Experiment I

In this experiment, we investigated the performance of the proposed algorithm in clustering balanced data. We generated 3000 synthetic data points from a mixture of three bivariate Gaussian densities given by

$$\frac{1}{3}\text{Gaussian}\begin{pmatrix}-5\\10\end{pmatrix}\begin{pmatrix}5&0\\0&5\end{pmatrix}+\frac{1}{3}\text{Gaussian}\begin{pmatrix}5\\0\end{pmatrix}\begin{pmatrix}5&0\\0&5\end{pmatrix}+\frac{1}{3}\text{Gaussian}\begin{pmatrix}0\\20\end{pmatrix}\begin{pmatrix}5&0\\0&5\end{pmatrix}$$

where Gaussian $[X, Y]$ is a Gaussian normal distribution with the mean X and the covariance matrix Y. The generated dataset by this density function, which is called BDS1, is shown in Fig. 11(a). In the experiment, we first applied the FGFKM algorithm to obtain the three cluster centers, as shown in Fig. 11(b). The three locations were very close to the "true" centers of the three clusters in the dataset. We also obtained different sets of nine cluster centers by the FGFKM algorithm with different m values and found that as m increases from 1.1 to 2, the distances between cluster centers belonging to the same clusters decrease. Then, we used the cluster centers that are obtained by the FGFKM algorithm with $m^* = 2$ to determine the number of clusters [shown in Fig. 14(a)].

B. Experiment II

In this experiment, we investigated the performance of the proposed algorithm in clustering imbalanced data. We generated two synthetic datasets, which we called IDS1 and IDS2, respectively.

1) IDS1: IDS1 has 2600 synthetic data points which aise from a mixture of three bivariate Gaussian densities given by

$$\frac{10}{13}\text{Gaussian}\begin{pmatrix}0\\0\end{pmatrix}\begin{pmatrix}10&0\\0&10\end{pmatrix}+\frac{2}{13}\text{Gaussian}\begin{pmatrix}12\\-5\end{pmatrix}\begin{pmatrix}2&0\\0&2\end{pmatrix}+\frac{1}{13}\text{Gaussian}\begin{pmatrix}10\\5\end{pmatrix}\begin{pmatrix}1&0\\0&1\end{pmatrix}.$$

The generated dataset by this density function is shown in Fig. 12(a). In the experiment, we applied the FGFKM algorithm to obtain different sets of 13 cluster centers by the FGFKM algorithm with different m values and found that as m increases from 1.1 to 2, the distances between the cluster centers belonging to the same clusters decrease. Then, we used the cluster centers that are obtained by the FGFKM algorithm with $m^* = 2$ to determine the number of clusters [shown in Fig. 14(b)].

TABLE V
CLUSTER RECOVERY FOR THE IDS2 DATA BY USING SIX ALGORITHMS

	HKM(B)						FKM(B)					
Clusters found	Objects in cluster	Original classes					Objects in cluster	Original classes				
		P_1	P_2	P_3	P_4	P_5		P_1	P_2	P_3	P_4	P_5
C_1	1647	1647	0	0	0	0	271	271	0	0	0	0
C_2	308	108	200	0	0	0	151	119	32	0	0	0
C_3	312	112	0	200	0	0	1448	1248	0	200	0	0
C_4	485	86	0	0	399	0	150	150	0	0	0	0
C_5	448	47	0	0	1	400	1180	212	168	0	400	400
	GK(B)						KFKM(B)					
Clusters found	Objects in cluster	Original classes					Objects in cluster	Original classes				
		P_1	P_2	P_3	P_4	P_5		P_1	P_2	P_3	P_4	P_5
C_1	1023	1023	0	0	0	0	1079	1097	0	0	0	0
C_2	864	664	200	0	0	0	31	31	0	0	0	0
C_3	440	240	0	200	0	0	627	384	43	200	0	0
C_4	443	45	0	0	398	0	812	322	0	0	400	90
C_5	430	28	0	0	2	400	651	184	157	0	0	310
	ICCM						MC					
Clusters found	Objects in cluster	Original classes					Objects in cluster	Original classes				
		P_1	P_2	P_3	P_4	P_5		P_1	P_2	P_3	P_4	P_5
C_1	1139	1139	0	0	0	0	1978	1978	0	0	0	0
C_2	1005	811	0	194	0	0	213	13	200	0	0	0
C_3	587	21	200	6	0	360	203	3	0	200	0	0
C_4	469	29	0	0	400	40	404	4	0	0	400	0
C_5	0	0	0	0	0	0	402	2	0	0	0	400

2) IDS2: IDS2 has 3200 synthetic data points which arise from a mixture of five bivariate Gaussian densities given by

$$\frac{10}{16}\text{Gaussian}\begin{pmatrix}0\\0\end{pmatrix}\begin{pmatrix}10&0\\0&10\end{pmatrix}+\frac{1}{16}\text{Gaussian}\begin{pmatrix}12\\4\end{pmatrix}\begin{pmatrix}1&0\\0&1\end{pmatrix}+\frac{1}{16}\text{Gaussian}\begin{pmatrix}12\\-6\end{pmatrix}\begin{pmatrix}1&0\\0&1\end{pmatrix}+\frac{2}{16}\text{Gaussian}\begin{pmatrix}-6\\12\end{pmatrix}\begin{pmatrix}4&0\\0&0.25\end{pmatrix}+\frac{2}{16}\text{Gaussian}\begin{pmatrix}6\\12\end{pmatrix}\begin{pmatrix}4&0\\0&0.25\end{pmatrix}.$$

The generated dataset by this density function is shown in Fig. 13(a). In the experiment, we applied the FGFKM algorithm to obtain different sets of 13 cluster centers by the FGFKM algorithm with different m values and found that as m increases from 1.1 to 1.8, the distances between the cluster centers in the same clusters decrease. Then, we used the cluster centers that are obtained by the FGFKM algorithm with $m^* = 1.8$ to determine the number of clusters [shown in Fig. 14(c)].

We compared the proposed algorithm with other five algorithms on the IDS1 and IDS2 data. According to the clustering results of these existing algorithms (shown in Tables III and V), we find that the "best" initial cluster centers did not bring the best results, because of the "uniform effect." We also see that the results that are obtained by the proposed algorithm are very close to the original classifications of the IDS1 and IDS2 datasets. Tables IV and VI show the comparisons of these existing algorithms for AC, PE, RE, ARI, and CV with the proposed algorithm.

TABLE VI
SUMMARY CLUSTERING RESULTS OF DIFFERENT ALGORITHMS ON THE IDS2 DATA

	HKM(R)	HKM(B)	FKM(R)	FKM(B)	GK(R)	GK(B)	KFKM(R)	KFKM(B)	ICCM	MC
AC	0.8745	0.8894	0.6502	0.6837	0.7892	0.8516	0.6653	0.6887	0.8469	0.9931
PE	0.7984	0.8012	0.7606	0.7978	0.7557	0.8285	0.7147	0.7162	0.8183	0.9819
RE	0.8325	0.9642	0.3605	0.3788	0.6714	0.5917	0.4598	0.5044	0.7188	0.9978
ARI	0.4835	0.7022	0.2442	0.2244	0.3785	0.3921	0.2022	0.2703	0.4540	0.9803
CV	0.7363	0.3099	0.1507	0.2224	0.7666	0.7564	0.4136	0.5963	0.6953	0.0195

TABLE VII
SUMMARY CLUSTERING RESULTS OF DIFFERENT ALGORITHMS ON THE WINE DATA

	HKM(R)	HKM(B)	FKM(R)	FKM(B)	GK(R)	GK(B)	KFKM(R)	KFKM(B)	ICCM	MC
AC	0.9146	0.9719	0.6104	0.6966	0.6269	0.8427	0.7165	0.8427	0.6011	0.9663
PE	0.9269	0.9765	0.7222	0.7753	0.6488	0.8606	0.7838	0.8603	0.6106	0.9718
RE	0.9285	0.9695	0.7072	0.7465	0.6405	0.8477	0.7722	0.8685	0.5791	0.9655
ARI	0.8004	0.9149	0.3632	0.3943	0.3419	0.5778	0.4372	0.5941	0.3779	0.9061
CV	0.1239	0.0652	0.4083	0.4268	0.3135	0.0842	0.3712	0.0517	0.6242	0.0566

TABLE VIII
SUMMARY CLUSTERING RESULTS OF DIFFERENT ALGORITHMS ON THE BREAST CANCER DATA

	HKM(R)	HKM(B)	FKM(R)	FKM(B)	GK(R)	GK(B)	KFKM(R)	KFKM(B)	ICCM	MC
AC	0.9609	0.9619	0.9189	0.9649	0.8882	0.9327	0.7803	0.8067	0.9431	0.9722
PE	0.9609	0.9617	0.9181	0.9640	0.8947	0.9395	0.8483	0.8854	0.9450	0.9654
RE	0.9528	0.9543	0.9128	0.9585	0.8690	0.9125	0.7081	0.7238	0.9293	0.9747
ARI	0.8489	0.8520	0.8219	0.8630	0.7096	0.7451	0.4372	0.3689	0.7826	0.9361
CV	0.0354	0.0331	0.0237	0.0248	0.1104	0.1160	0.3712	0.4863	0.0699	0.0456

TABLE IX
SUMMARY CLUSTERING RESULTS OF DIFFERENT ALGORITHMS ON THE ECOLI DATA

	HKM(R)	HKM(B)	FKM(R)	FKM(B)	GK(R)	GK(B)	KFKM(R)	KFKM(B)	ICCM	MC
AC	0.7846	0.8226	0.7188	0.7554	0.7602	0.7951	0.6916	0.7982	0.7401	0.8746
PE	0.7687	0.8045	0.6759	0.7055	0.7599	0.7936	0.6632	0.7590	0.6806	0.8279
RE	0.6359	0.8358	0.5692	0.6730	0.6369	0.6659	0.5911	0.7078	0.6793	0.8206
ARI	0.5175	0.6969	0.6183	0.6522	0.4605	0.4783	0.5335	0.6720	0.5978	0.7917
CV	0.3162	0.0408	0.1336	0.0900	0.4987	0.5223	0.2069	0.0658	0.1659	0.0324

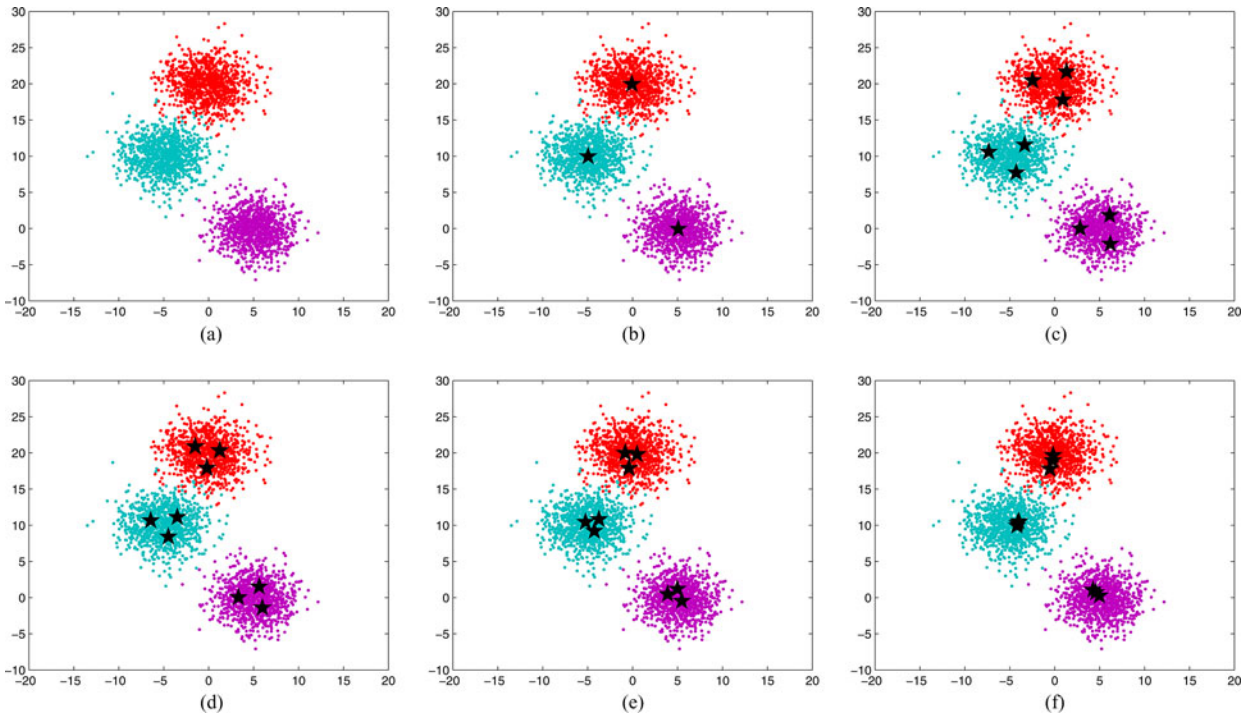


Fig. 11. (a) Data distribution on the BDS1 data. (b) Three cluster centers obtained by the FGFKM algorithm when $m = 1.1$. (c) Nine cluster centers obtained by the FGFKM algorithm when $m = 1.1$. (d) Nine cluster centers obtained by the FGFKM algorithm when $m = 1.5$. (e) Nine cluster centers obtained by the FGFKM algorithm when $m = 1.8$. (f) Nine cluster centers obtained by the FGFKM algorithm when $m = 2$.

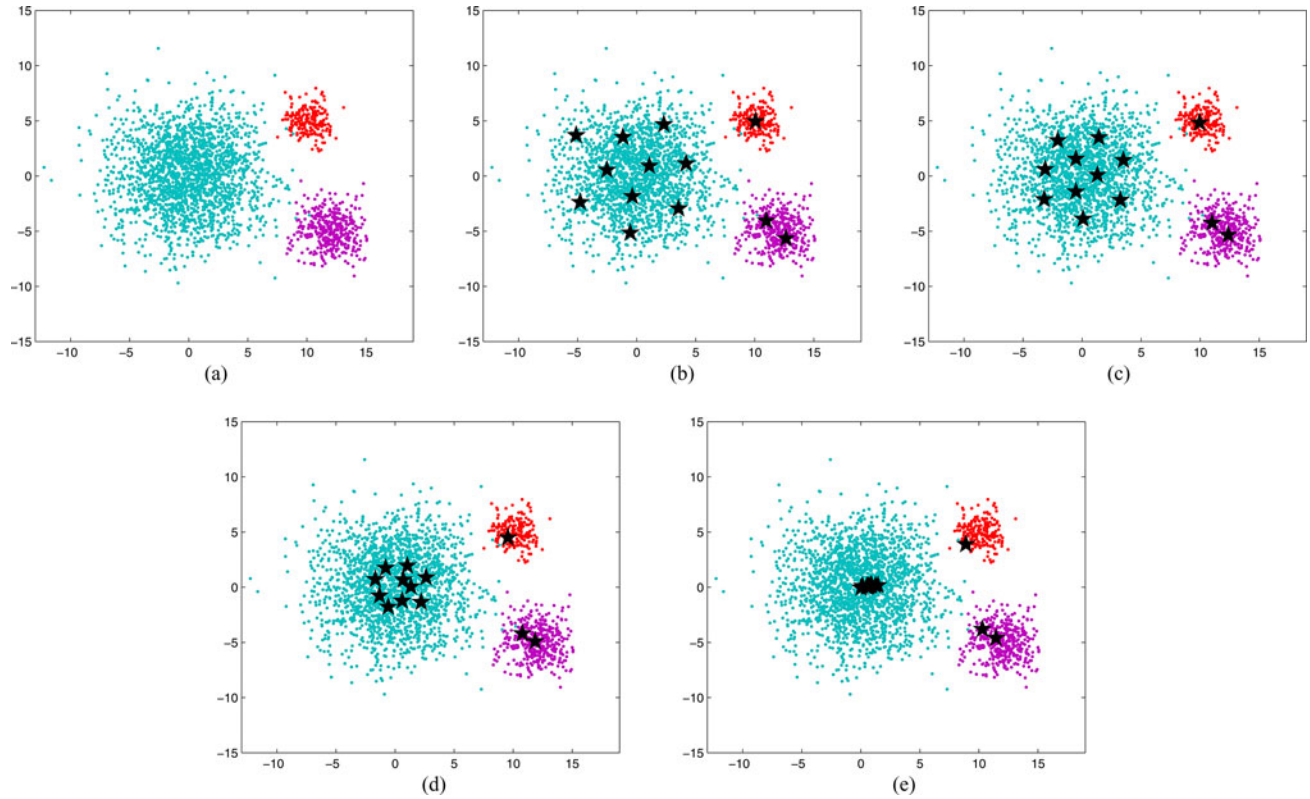


Fig. 12. (a) Data distribution on the IDS1 data. (b) Thirteen cluster centers obtained by the FGFKM algorithm when $m = 1.1$. (c) Thirteen cluster centers obtained by the FGFKM algorithm when $m = 1.5$. (d) Thirteen cluster centers obtained by the FGFKM algorithm when $m = 1.8$. (e) Thirteen cluster centers obtained by the FGFKM algorithm when $m = 2$.

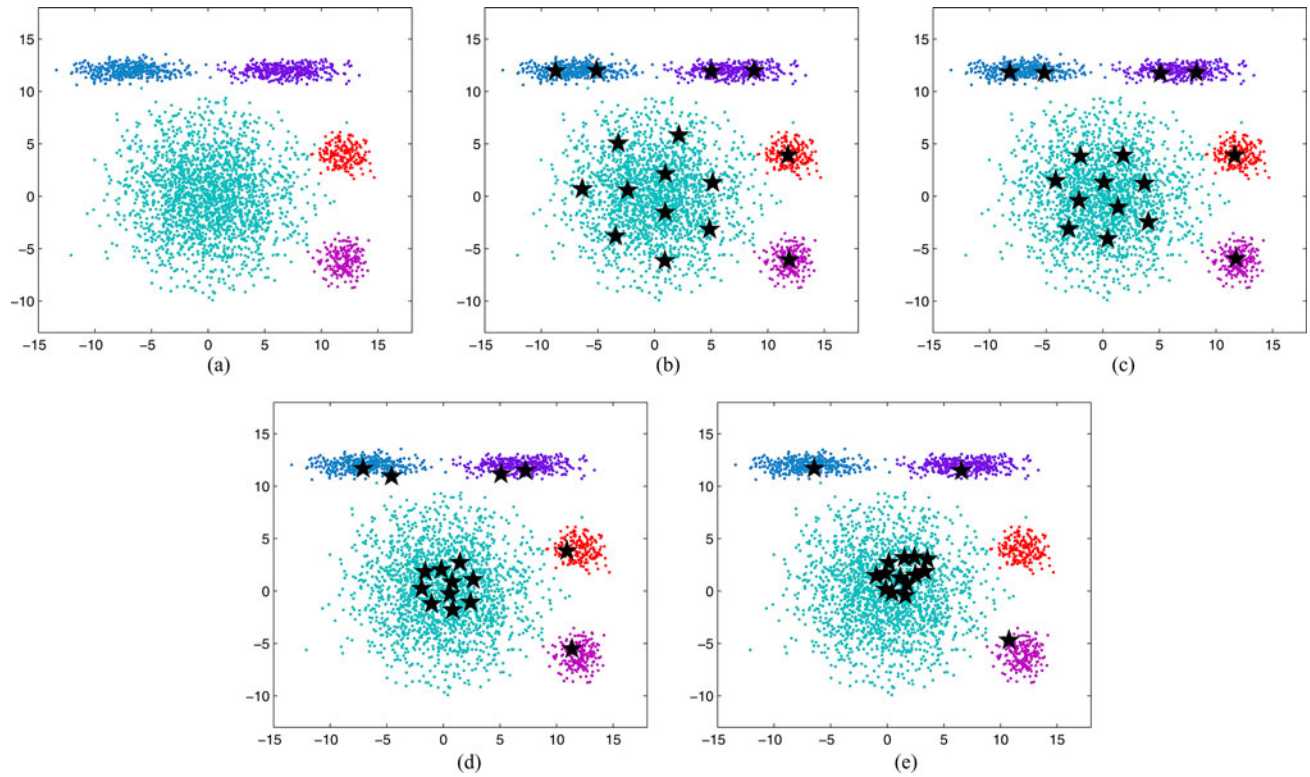


Fig. 13. (a) Imbalanced data distribution on the IDS2 data. (b) Thirteen cluster centers obtained by the FGFKM algorithm when $m = 1.1$. (c) Thirteen cluster centers obtained by the FGFKM algorithm when $m = 1.5$. (d) Thirteen cluster centers obtained by the FGFKM algorithm when $m = 1.8$. (e) Thirteen cluster centers obtained by the FGFKM algorithm when $m = 1.9$.

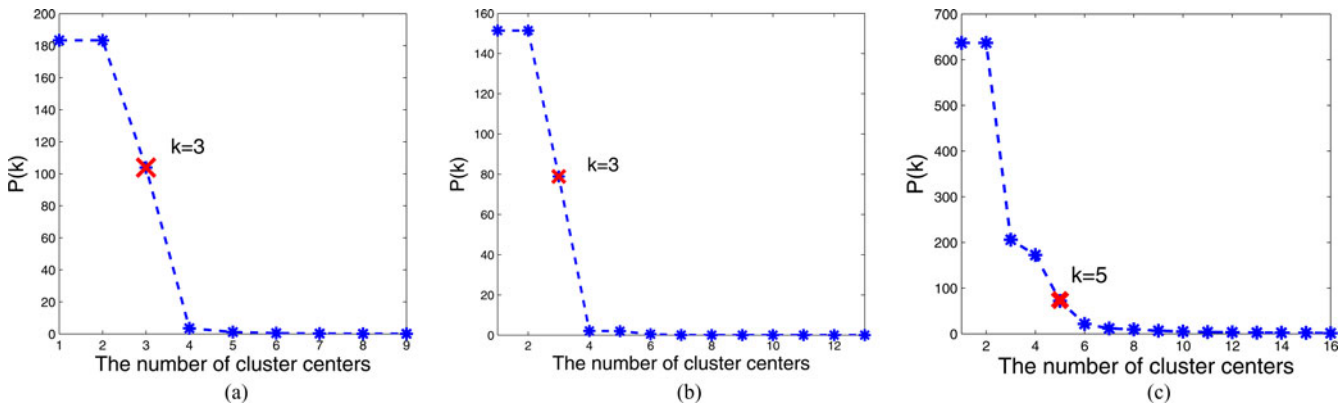


Fig. 14. (a) Determination of the number of clusters on the BDS1 data. (b) Determination of the number of clusters on the IDS1 data. (c) Determination of the number of clusters on the IDS2 data.

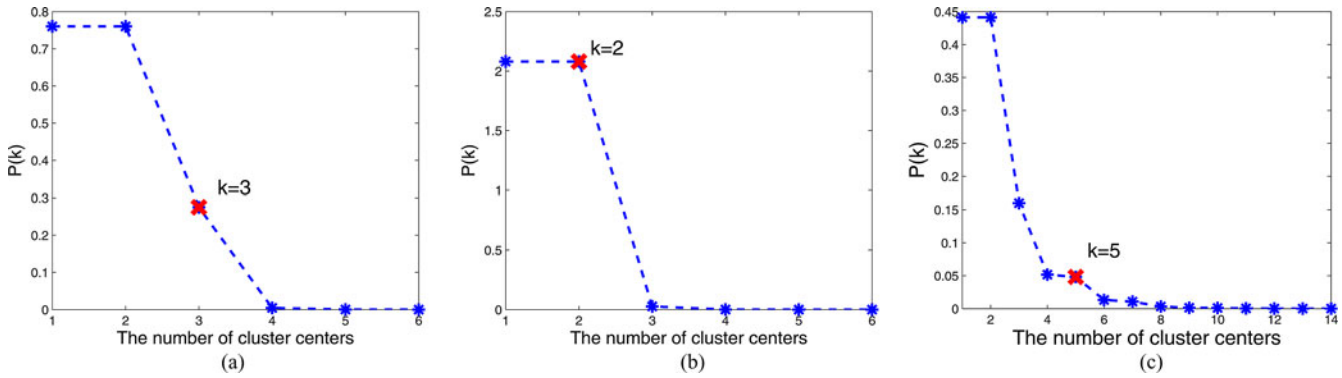


Fig. 15. (a) Determination of the number of clusters on the wine data. (b) Determination of the number of clusters on the breast data. (c) Determination of the number of clusters on the ecoli data.

C. Experiment III

To show the practical applicability of the proposed algorithm, we apply it to three real datasets, wine, breast cancer, and ecoli data, which are available at the UCI Machine Learning Repository.

Similar to Experiment II, the clustering results of the proposed algorithm on the selected real datasets will be compared with other five algorithms. Moreover, the number of clusters for each given dataset determined by the proposed algorithm is shown in Fig. 15.

1) Wine Data: These data were the result of a chemical analysis of wines grown in the same region in Italy. It consists of 178 data objects and 13 continuous attributes. It has three clusters. Fig. 15(a) shows that the number of clusters on the wine data is 3. The clustering results on the wine data are summarized in Table VII.

2) Breast Cancer Data: This breast cancer domain was obtained from the University Medical Center, Institute of Oncology, Ljubljana, Yugoslavia. It consists of 699 data objects with nine continuous attributes. However, there are 16 objects in the dataset that contain a single missing (i.e., unavailable) attribute value. We remove these objects from the dataset. It has two clusters: benign (444 data objects) and malignant (239 data objects). Fig. 15(b) shows that the number of clusters on the breast cancer data is 2 indeed. The clustering results on the breast cancer data are summarized in Table VIII.

3) Ecoli Data: These data were about protein localization sites in eukaryotic cells. It consists of 336 data objects and seven continuous attributes. We select 327 objects from the dataset. It has five clusters. The other nine objects as the outliers are removed. Fig. 15(c) shows that the number of clusters on the ecoli data is 5. The clustering results on the ecoli data are summarized in Table IX.

According to Fig. 15, we see that our proposed algorithm can exactly find the "true" numbers of clusters on these real datasets. Table VII shows that the proposed algorithm can obtain a very good clustering result on the wine dataset which is very close to the result of the hard k-means algorithm with "true" cluster centers. Tables VIII and IX show that on the breast cancer and ecoli datasets, the performance of the proposed algorithm is better than that of other clustering algorithms. In summary, the aforementioned experimental studies illustrate that the proposed algorithm cannot only effectively cluster balanced data but can have a good potential in handling imbalanced data as well.

VI. Conclusion

In this paper, we have presented an organized study of the effect of imbalanced data distributions on the performance of the k-means-type algorithms. We found that the fuzzy k-means algorithm more possibly produce clusters with relatively uniform sizes than the hard k-means algorithm, even if the input data have a range of varied "true" cluster sizes. As the fuzzy

index m increases, the effect becomes evident. We proposed a multicenters algorithm to avoid the occurrence of the effect. In the proposed algorithm, we first use the FGFKM algorithm to obtain several reliable cluster centers and partition the dataset into several subclusters. Furthermore, based on the fuzzy index m and the Max–Min distances between the selected cluster centers, the number of clusters is determined. Finally, a separation measure was proposed to evaluate how well two subclusters are separated. Multicenters with small separations were organized to model each cluster in the agglomerative method, instead of one single center. The proposed algorithm only needs two parameters which are easily set up. Our experimental results have shown the effectiveness of the proposed algorithm to cluster balanced and imbalanced data.

ACKNOWLEDGMENT

The authors are very grateful to the editors and reviewers for their valuable comments and suggestions.

REFERENCES

[1] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice–Hall, 1988.

[2] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probabil.*, 1967, vol. 1, pp. 281–297.

[3] E. R. Ruspini, "A new Approach to clustering," *Inf. Control*, vol. 15, no. 1, pp. 22–32, 1969.

[4] J. C. Bezdek, "A convergence theorem for the fuzzy ISODATA clustering algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-2, no. 1, pp. 1–8, Jan. 1980.

[5] N. R. Pal and J. C. Bezdek, "On cluster validity for the fuzzy c-means model," *IEEE Trans. Fuzzy Syst.*, vol. 3, no. 3, pp. 370–379, Aug. 1995.

[6] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Reading, MA: Addison-Wesley, 2005.

[7] Z. X. Huang, M. K. Ng, H. Rong, and Z. Li, "Automated variable weighting in k-means type clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 657–668, May 2005.

[8] E. Y. Chan, W. K. Ching, M. K. Ng, and Z. X. Huang, "An optimization algorithm for clustering using weighted dissimilarity measures," *Pattern Recognit.*, vol. 37, no. 5, pp. 943–952, 2004.

[9] Y. H. Qian, J. Y. Liang, W. Pedrycz, and C. Y. Dang, "Positive approximation: An accelerator for attribute reduction in rough set theory," *Artif. Intell.*, vol. 174, no. 5–6, pp. 597–618, 2010.

[10] L. P. Jing, M. K. Ng, and Z. X. Huang, "An entropy weighting k-means algorithm for subspace clustering of high-dimensional sparse data," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 8, pp. 1026–1041, Aug. 2007.

[11] J. S. Zhang and Y. W. Leung, "Robust clustering by pruning outliers," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 33, no. 6, pp. 983–998, Dec. 2003.

[12] A. Zhou, F. Cao, Y. Yan, C. Sha, and X. He, "Distributed data stream clustering: A fast EM-based approach," in *Proc. 23rd Int. Conf. Data Eng.*, 2007, pp. 736–745.

[13] M. Breunig, H. P. Kriegel, R. Ng, and J. Sander, "LOF: Identifying density based local outliers," in *Proc. Int. Conf. ACM Special Interest Group Manag. Data*, 2000, pp. 427–438.

[14] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. ACM Special Interest Group Knowl. Discovery Data Mining*, 1996, pp. 226–231.

[15] P. Bradley, U. Fayyad, and C. Reina, "Scaling clustering algorithms to large databases," in *Proc. 4th Int. Conf. ACM Special Interest Group Knowl. Discovery Data Mining*, 1998, pp. 9–15.

[16] F. Murtagh, "Clustering massive data sets," in *Handbook of Massive Data Sets*. Norwell, MA: Kluwer, 2000.

[17] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in *Proc. ACM Special Interest Group Manag. Data*, 1996, pp. 103–114.

[18] Z. X. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data Mining Knowl. Discov.*, vol. 2, no. 3, pp. 283–304, 1998.

[19] F. Y. Cao, J. Y. Liang, L. Bai, X. Zhao, and C. Dang, "A framework for clustering categorical time-evolving data," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 5, pp. 872–882, Oct. 2010.

[20] J. C. Bezdek, "A physical interpretation of Fuzzy ISODATA," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 5, pp. 387–390, May 1976.

[21] L. O. Hall, A. M. Bensaid, and L. P. Clarke, "A comparison of neural network and fuzzy clustering techniques in segmenting magnetic resonance images of the brain," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 672–682, Sep. 1992.

[22] R. L. Cannon, J. V. Dave, and J. C. Bezdek, "Efficient implementation of the fuzzy c-means clustering algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 2, pp. 248–255, Mar. 1986.

[23] J. Yu and M. S. Yang, "Optimality test for generalized FCM and its application to parameter selection," *IEEE Trans. Fuzzy Systems*, vol. 13, no. 1, pp. 164–176, Feb. 2005.

[24] F. Y. Cao, J. Y. Liang, and G. Jiang, "An initialization method for the k-means algorithm using neighborhood model," *Comput. Math. Appl.*, vol. 58, no. 3, pp. 474–483, 2009.

[25] M. Laszlo and S. Mukherjee, "A genetic algorithm using hyper-quadtrees for low-dimensional k-means clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 533–543, Apr. 2006.

[26] D. Arthur and S. Vassilvitskii, "K-means++: the advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algo.*, 2007, pp. 1027–1035.

[27] A. Likas, M. Vlassis, and J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognit.*, vol. 35, no. 2, pp. 451–461, 2003.

[28] A. M. Bagirov, "Modified global k-means algorithm for minimum sum-of-squares clustering problems," *Pattern Recognit.*, vol. 41, no. 10, pp. 3192–3199, 2008.

[29] Z. C. Lai and T. J. Huang, "Fast global k-means clustering using cluster membership and inequality," *Pattern Recognit.*, vol. 43, no. 5, pp. 1954–1963, 2010.

[30] G. Hamerly and C. Elkan, "Learning the k in k-means," in *Proc. 17th Ann. Conf. Neural Inf. Process. Syst.*, Dec. 2003, pp. 1–8.

[31] J. J. Li, M. K. Ng, Y. M. Cheng, and Z. H. Huang, "Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 11, pp. 1519–1534, Nov. 2008.

[32] M. Halkidi and M. Vazirgiannis, "A density-based cluster validity approach using multi-representatives," *Pattern Recognit. Lett.*, vol. 29, pp. 773–786, 2008.

[33] H. Xiong, J. J. Wu, and J. Chen, "K-means clustering versus validation measures: A data-distribution perspective," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 318–331, Apr. 2009.

[34] B. Scholkopf, A. J. Smola, and K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.

[35] F. Camastra and A. Verri, "A novel kernel method for clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 801–805, May 2005.

[36] D. Graves and W. Pedrycz, "Kernel-based fuzzy clustering and fuzzy clustering: A comparative experimental study," *Fuzzy Sets Syst.*, vol. 161, no. 4, pp. 522–543, 2010.

[37] R. Babuska, P. J. vanderVeen, and U. Kaymak, "Improved covariance estimation for Gustafson–Kessel clustering," in *Proc. IEEE Int. Conf. Fuzzy Syst.*, 2002, pp. 1081–1085.

[38] R. Krishnnapuram and C. P. Freg, "Fitting an unknown number of lines and planes to image data through compatible cluster merging," *Pattern Recognit.*, vol. 25, no. 4, pp. 385–400, 1992.

[39] S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases," in *Proc. Int. Conf. ACM Special Interest Group Manag. Data*, 1998, pp. 73–84.

[40] M. H. Liu, X. D. Jiang, and A. C. Kot, "A multi-prototype clustering algorithm," *Pattern Recognit.*, vol. 42, pp. 689–698, 2009.

[41] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowledge Discovery Data Mining*, 1996, pp. 226–231.

[42] D. Pelleg and A. W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 727–734.

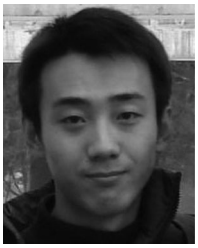
[43] P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy*. San Francisco, CA: Freeman, 1973.

[44] B. King, "Step-wise clustering procedures," *J. Amer. State Assoc.*, vol. 69, pp. 86–101, 1967.

[45] G. Karypis, E.-H. S. Han, and V. Kumar, "Chameleon: A hierarchical clustering algorithm using dynamic modeling," *IEEE Comput.*, vol. 32, no. 8, pp. 68–75, Aug. 1999.
[46] Y. M. Yang, "An evaluation of statistical approaches to text categorization," *J. Inf. Retrieval*, vol. 1, no. 1–2, pp. 67–88, 1999.
[47] L. Hubert and P. Arabie, "Comparing partitions," *J. Classificat.*, vol. 2, no. 1, pp. 193–218, 1985.
[48] (2011). *UCI Machine Learning Repository* [Online]. Available:http://www.ics.uci.edu/mlearn/MLRepository.html.

Jiye Liang received the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1990 and 2001, respectively.

He is currently a Professor with the School of Computer and Information Technology and the Key Laboratory of Computational Intelligence and Chinese Information Processing of the Ministry of Education, Shanxi University, Taiyuan, China. He has authored or coauthored more than 100 journal papers in his research fields. His current research interests include computational intelligence, granular computing, data mining, and knowledge discovery.



Liang Bai received the M.S. degree in computer science from Shanxi University, Taiyuan, China, in 2009. He is currently working toward the Ph.D. degree with the School of Computer and Information Technology, Shanxi University.

His research interests include data mining and machine learning.

Chuangyin Dang received the M.S. degree in applied mathematics from Xidian University, Xidian, China, in 1986 and the Ph.D degree in operations research/economics from the University of Tilburg, Tilburg, The Netherlands, in 1991 .

He is currently an Associate Professor with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong. His research interests include computational intelligence and optimization theory and technology.

Fuyuan Cao received the M.S. and Ph.D. degrees in computer science, both from Shanxi University, Taiyuan, China, in 2004 and 2009, respectively.

He is currently an Associate Professor with the School of Computer and Information Technology, Shanxi University. His research interests include data mining and machine learning.