Agglomerative Fuzzy K-Means Clustering Algorithm with Selection of Number of Clusters

Mark Junjie Li, Michael K. Ng, Yiu-ming Cheung, *Senior Member*, *IEEE*, and Joshua Zhexue Huang

Abstract—In this paper, we present an agglomerative fuzzy K-Means clustering algorithm for numerical data, an extension to the standard fuzzy K-Means algorithm by introducing a penalty term to the objective function to make the clustering process not sensitive to the initial cluster centers. The new algorithm can produce more consistent clustering results from different sets of initial clusters centers. Combined with cluster validation techniques, the new algorithm can determine the number of clusters in a data set, which is a well-known problem in K-Means clustering. Experimental results on synthetic data sets (2 to 5 dimensions, 500 to 5,000 objects and 3 to 7 clusters), the BIRCH two-dimensional data set of 20,000 objects and 100 cluster0and the WINE data set of 178 objects, 17 dimensions, and 3 clusters from UCI have demonstrated the effectiveness of the new algorithm in producing consistent clustering results and determining the correct number of clusters in different data sets, some with overlapping inherent clusters.

Index Terms—Fuzzy K-Means clustering, agglomerative, number of clusters, cluster validation.

1 Introduction

Clustering is a process of grouping a set of objects into clusters so that the objects in the same cluster have high similarity but are very dissimilar with objects in other clusters. Various types of clustering methods have been proposed and developed, see, for instance, [1]. The K-Means algorithm [1], [2], [3], [5] is well known for its efficiency in clustering large data sets. Fuzzy versions of the K-Means algorithm have been reported by Ruspini [4] and Bezdek [6], where each pattern is allowed to have memberships in all clusters rather than having a distinct membership to one single cluster. Numerous problems in real-world applications, such as pattern recognition and computer vision, can be tackled effectively by the fuzzy K-Means algorithms, see, for instance, [7], [8], and [9].

There are two major issues in the application of K-Means-type (nonfuzzy or fuzzy) algorithms in cluster analysis. The first issue is that the number of clusters k needs to be determined in advance as an input to these algorithms. In a real data set, k is usually unknown. In practice, different values of k are tried, and cluster validation techniques are used to measure the clustering results and determine the best value of k, see, for instance, [1]. In [10], Hamerly and Elkan studied statistical methods to learn k in K-Means-type algorithms.

- *M.J. Li is with the Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong. E-mail: jjli@math.hkbu.edu.hk.*
- *M.K. Ng is with the Centre for Mathematical Imaging and Vision and the Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong. E-mail: mng@math.hkbu.edu.hk.*
- *Y.-m. Cheung is with the Department of Computer Science, Hong Kong University, Kowloon Tong, Hong Kong. E-mail: ymc@comp.hkbu.edu.hk.*
- *J.Z. Huang is with the E-Business Technology Institute, The University of Hong Kong, Pokfulam Road, Hong Kong. E-mail: jhuang@eti.hku.hk.*

Manuscript received 1 May 2007; revised 1 Jan. 2008; accepted 21 Apr. 2008; published online 1 May 2008.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2007-05-0245.
Digital Object Identifier no. 10.1109/TKDE.2008.88.

The second issue is that the K-Means-type algorithms use alternating minimization methods to solve nonconvex optimization problems in finding cluster solutions [1]. These algorithms require a set of initial cluster centers to start and often end up with different clustering results from different sets of initial cluster centers. Therefore, the K-Means-type algorithms are very sensitive to the initial cluster centers. Usually, these algorithms are run with different initial guesses of cluster centers, and the results are compared in order to determine the best clustering results. One way is to select the clustering results with the least objective function value formulated in the K-Means-type algorithms, see, for instance, [11]. In addition, cluster validation techniques can be employed to select the best clustering result, see, for instance, [1]. Other approaches have been proposed and studied to address this issue by using a better initial seed value selection for K-Means algorithm using genetic algorithm [12], [13], [14], [15]. Recently, Arthur and Vassilvitskii [16] proposed and studied a careful seeding for initial cluster centers to improve clustering results.

In this paper, we propose an agglomerative fuzzy K-Means clustering algorithm for numerical data to tackle the above two issues in application of the K-Means-type clustering algorithms. The new algorithm is an extension to the standard fuzzy K-Means algorithm by introducing a penalty term to the objective function to make the clustering process not sensitive to the initial cluster centers. The new algorithm can produce more consistent clustering results from different sets of initial clusters centers. Combined with cluster validation techniques, the new algorithm can determine the number of clusters in a data set. Experimental results have demonstrated the effectiveness of the new algorithm in producing consistent clustering results and determining the correct number of clusters in different data sets, some with overlapping inherent clusters.

The organization of this paper is as follows: In Section 2, we review the related work. In Section 3, we formulate the

agglomerative fuzzy K-Means algorithm and combine it with clustering validation techniques to select the number of clusters. In Section 4, experimental results are given to illustrate the effectiveness of the new algorithm. Finally, concluding remarks are given in Section 5.

2 Related Work

2.1 Cluster Validation

The most important parameter in the K-Means-type algorithm is the number of clusters. The number of clusters in a data set is a user-defined parameter, which is difficult to specify. In practice, different k values are tried, and the results are compared and analyzed with cluster validation techniques to determine the most appropriate number of clusters. For this purpose, different validation indices have been proposed [1], [17], [18], [19], [20], [21]. For instance, Gath et al. [19] proposed a cluster validation index based on performance measure using hypervolume and density criteria. For the evaluation of the fuzzy K-Means clustering, several validation indices are available, including partitional coefficients and classification entropy [18]. Recently, Rezaee et al. [21] proposed a validation index that is derived from a linear combination of the average scattering (compactness) of clusters and the distance (separation) between clusters. Sun et al. [20] proposed a validation index with a suitable balance between the compactness factor and cluster separation.

On the other hand, some validation indices were developed for the probabilistic mixture-model framework. In density estimation, the commonly used criteria of AIC [22] and BIC [23] seem to be adequate in finding the correct number of clusters for a suitable density estimate. However, these conventional criteria can overestimate or underestimate the cluster number due to the difficulty of choosing an appropriate penalty function [24]. Hamerly and Elkan [10] proposed a statistical framework to test a hypothesis on the subset of data following a Gaussian distribution. Other comprehensive criteria include Efron information criterion (EIC) [25], cross-validation-based information criterion (CVIC) [26], minimum information ratio criterion (MIR) [27], and informational complexity criterion (ICOMP) [28], see the summary in [29].

2.2 Optimization Functions

An alternative approach to determining the number of clusters is to define an optimization function that involves both cluster solutions and the number of clusters. Recently, Cheung [30] studied a rival penalized competitive learning algorithm [31] that has demonstrated a very good result in finding the cluster number. His algorithm is formulated by learning the parameters of a mixture model through the maximization of a weighted likelihood function. In the learning process, some initial seed centers move to the genuine positions of the cluster centers in a data set, and other redundant seed points will stay at the boundaries or outside of the clusters.

The Bayesian-Kullback Ying-Yang learning theory has been proposed in [32]. It is a unified algorithm for both unsupervised and supervised learning, which provides us a reference for solving the problem of selection of the cluster number. The experimental results worked very well for many samples. However, for a relatively small number of samples, the maximum likelihood method with the expectation-maximization algorithm for estimating the mixture model parameters do not adequately reflect the characteristics of the cluster structure [33].

2.3 Competitive Agglomeration

An agglomerative clustering procedure starts with each object as one cluster and forms the nested sequence by successively merging clusters. The main advantage of the agglomerative procedure is that clustering is not influenced by initialization and local minima. In addition, the number of clusters need not be specified a priori. Practitioners can analyze the dendrogram produced by the clustering process, cut the dendrogram at a suitable level, and then identify the clusters. Based on the agglomerative procedure, Frigui and Krishnapuram [34] proposed a new fuzzy clustering algorithm that minimizes an objective function that produces a sequence of partitions with a decreasing number of clusters. The initial partition has an over specified number of clusters, and the final one has the optimal number of clusters. In the clustering process, adjacent clusters compete for objects in a data set, and the clusters that lose the competition gradually become depleted and vanish. Experimental results have shown that the performance of the competitive agglomeration algorithm is quite good. We remark that their proposed algorithm assumes the objects-clusters membership value do not change significantly from one iteration to the next one to simplify the computational procedure. Moreover, in the clustering process, the objects-clusters membership value may not be confined between 0 and 1. An additional procedure may be applied to the algorithm to set the suitable values.

3 The Agglomerative Fuzzy K-MEANS Algorithm

Let $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$ be a set of n objects in which each object X_i is represented as $[x_{i,1}, x_{i,2}, \ldots, x_{i,m}]$, where m is the number of numerical attributes. To cluster $\boldsymbol{X}$ into k clusters by the agglomerative fuzzy K-Means algorithm [35] is to minimize the following objective function:

$$P(U, Z) = \sum_{j=1}^{k}\sum_{i=1}^{n} u_{i,j} D_{i,j} + \lambda \sum_{j=1}^{k}\sum_{i=1}^{n} u_{i,j} \log u_{i,j} \qquad (1)$$

subject to

$$\sum_{j=1}^{k} u_{i,j} = 1, \quad u_{i,j} \in (0, 1], \quad 1 \le i \le n, \qquad (2)$$

where $U = [u_{i,j}]$ is an n-by-k partition matrix, $u_{i,j}$ represents the association degree of membership of the ith object x_i to the jth cluster z_j, $Z = [z_1, z_2, \ldots, z_k]^T$ is an k-by-m matrix containing the cluster centers, and $D_{i,j}$ is a dissimilarity measure between the jth cluster center and the ith object. Here, the square of the euclidean norm is used as the dissimilarity measure, i.e.,

$$D_{i,j} = \sum_{l=1}^{m} (z_{j,l} - x_{i,l})^2.$$

Such dissimilarity measure is commonly used in clustering, see, for instance, [1] and [2]. The first term in (1) is the cost function of the standard K-Means algorithm. The second term is added to maximize the negative objects-to-clusters membership entropy in the clustering process. Because of the second term, $u_{i,j}$ can choose between 0 and 1, which represents a fuzzy clustering:

- When $u_{i,j}$ is close to zero for all $j \neq j^*$ and u_{i,j^*} is close to one, the negative objects-to-clusters entropy value $-\sum_{j=1}^{k} u_{i,j} \log u_{i,j}$ is close to zero. In this case, the ith object is firmly assigned to the j^*th cluster, and the corresponding entropy value is small.
- However, when $u_{i,j}$ are about the same for some clusters, and the $u_{i,j}$ values for other clusters are close to zero, the negative objects-to-clusters membership entropy becomes more positive, i.e., much larger than zero. In this situation, the ith object belongs to several clusters.

Therefore, with the weight entropy term, the clustering process can simultaneously minimize the within cluster dispersion and maximize the negative weight entropy to determine clusters to contribute to the association of objects.

3.1 The Optimization Procedure

Minimization of P in (1) with the constraints forms a class of constrained nonlinear optimization problems whose solutions are unknown. We can extend the standard K-Means clustering process to minimize P. The usual method toward optimization of P is to use the partial optimization for U and Z. In this method, we first fix U and minimize the reduced P with respect to Z. Then, we fix Z and minimize the reduced P with respect to U.

Given U fixed, Z is updated as

$$z_{jl} = \frac{\sum_{i=1}^{n} u_{i,j}\, x_{i,l}}{\sum_{i=1}^{n} u_{i,j}} \quad \text{for } 1 \leq j \leq k \text{ and } 1 \leq l \leq m. \tag{3}$$

We note that (3) is independent of the parameter λ.

Given that Z fixed, U is updated as follows: We use the Lagrangian multiplier technique to obtain the following unconstrained minimization problem:

$$\tilde{P}(U, \alpha) = \sum_{j=1}^{k} \sum_{i=1}^{n} \left(u_{i,j} D_{i,j} + \lambda u_{i,j} \log u_{i,j}\right) + \alpha_i \sum_{i=1}^{n} \left(\sum_{j=1}^{k} u_{i,j} - 1\right),$$

where $\alpha = [\alpha_1, \ldots, \alpha_n]$ is the vector containing the Lagrangian multipliers. If $(\hat{U}, \hat{\alpha})$ is a minimizer of $\tilde{P}(U, \alpha)$, the gradients in both sets of variables must vanish. Thus,

$$\frac{\partial \tilde{P}(\hat{U}, \hat{\alpha})}{\partial \hat{u}_{i,j}} = D_{i,j} + \lambda(1 + \log u_{i,j}) + \hat{\alpha}_i = 0, \quad 1 \leq j \leq k,\ 1 \leq i \leq n, \tag{4}$$

and

$$\frac{\partial \tilde{P}(\hat{U}, \hat{\alpha})}{\partial \hat{\alpha}_i} = \sum_{i=1}^{k} \hat{u}_{i,j} - 1 = 0. \tag{5}$$

From (4), we obtain

$$\hat{u}_{i,j} = \exp\left(\frac{-D_{i,j}}{\lambda}\right) \exp(-1) \exp\left(\frac{-\hat{\alpha}_i}{\lambda}\right). \tag{6}$$

By substituting (6) into (5), we have

$$\sum_{j=1}^{k} \hat{u}_{i,j} = \sum_{j=1}^{k} \exp\left(\frac{-D_{i,j}}{\lambda}\right) \exp(-1) \exp\left(\frac{-\hat{\alpha}_i}{\lambda}\right) = \exp(-1) \exp\left(\frac{-\hat{\alpha}_i}{\lambda}\right) \sum_{j=1}^{k} \exp\left(\frac{-D_{i,j}}{\lambda}\right) = 1.$$

It follows that

$$\hat{u}_{i,j} = \frac{\exp\left(\frac{-D_{i,j}}{\lambda}\right)}{\sum_{l=1}^{k} \exp\left(\frac{-D_{l,j}}{\lambda}\right)}, \tag{7}$$

and U can be updated by (7). The alternating minimization procedure between Z and U can be applied to (1). The optimization procedure to solve (1) is given as follows:

THE AGGLOMERATIVE FUZZY K-MEANS ALGORITHM:

1. SET THE PENALTY FACTOR λ. RANDOMLY CHOOSE INITIAL POINTS $Z^{(0)} = \{Z_1, Z_2, \ldots, Z_k\}$. DETERMINE $U^{(0)}$ SUCH THAT $P(U^{(0)}, Z^{(0)})$ IS MINIMIZED BY USING (7). SET $t = 0$.
2. LET $\hat{Z} = Z^t$, SOLVE PROBLEM $\mathrm{P}(\mathrm{U}, \hat{Z})$ TO OBTAIN U^{t+1}. IF $\mathrm{P}(U^{t+1}, \hat{Z}) = \mathrm{P}(U^t, \hat{Z})$, OUTPUT $(U^t, \hat{Z})$ AND STOP; OTHERWISE, GO TO STEP 3.
3. LET $\hat{U} = U^{t+1}$, SOLVE PROBLEM $\mathrm{P}(\hat{U}, Z)$ TO OBTAIN Z^{t+1}. IF $\mathrm{P}(\hat{U}, Z^{t+1}) = \mathrm{P}(\hat{U}, Z^t)$, OUTPUT $(\hat{U}, Z^t)$ AND STOP; OTHERWISE, SET $t = t + 1$ AND GO TO STEP 2.

3.2 The Properties of the Algorithm

In the clustering process, the algorithm tries to minimize the within cluster dispersion and maximize the sum of the negative weight entropies of all objects, so the objective function (1) is minimized. Which part plays a more important role in the minimization process of (1) is balanced by the parameter λ. We know that the weight entropy of an object measures whether the object is assigned to a single cluster, in which case the entropy is equal to zero, or to several clusters, in which case the entropy is positive number. Maximization of the sum of the negative entropies of all objects is to assign each object to more clusters instead of a single cluster. Therefore, the parameter λ has the following properties to control the clustering process.

1. When λ is large such that the value of

$$\sum_{j=1}^{k} \sum_{i=1}^{n} u_{i,j} D_{i,j}$$

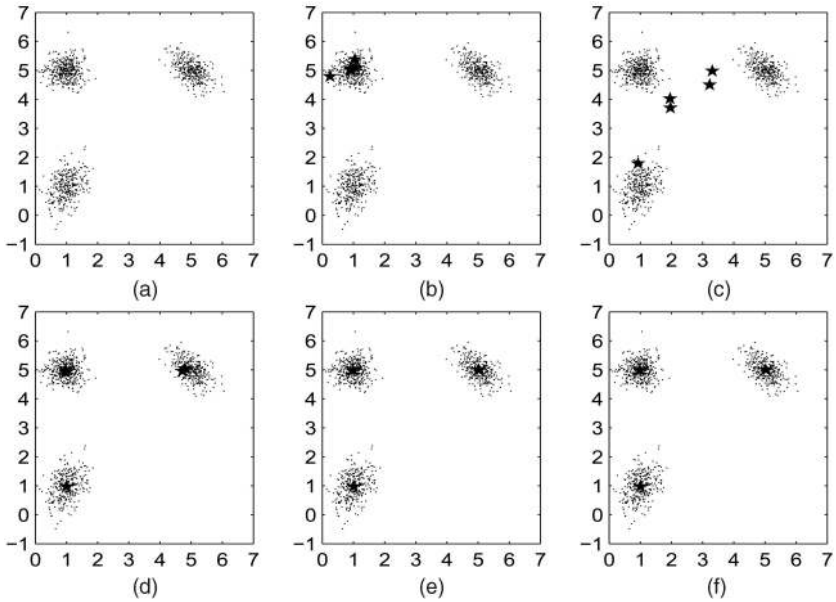


Fig. 1. The results obtained via the agglomerative fuzzy K-Means The positions of the cluster centers are marked by "$\star$." (a) Original data set. (b) Initial seed points as cluster centers. (c) Positions of cluster centers after the first iteration. (d) Second iteration. (e) Third iteration. (f) Eighth iterations.

(the within cluster dispersion term) is much less than the value of $\lambda \sum_{j=1}^{k} \sum_{i=1}^{n} u_{i,j} \log u_{i,j}$ (the entropy term), the entropy term will play a more important role to minimize (1). The clustering process will try to assign each object to more clusters to make the second term more negative. When the weights $u_{i,j}$ of an object to all clusters are equal, the object weight entropy is the largest. Since the locations of objects are fixed, to achieve the largest object entropy, move the cluster centers to the same location. Therefore, when λ is large, the clustering process turns to move some clusters centers to the same locations to maximize the sum of the negative entropies of all objects.

2. When λ is small such that the value of

$$\sum_{j=1}^{k} \sum_{i=1}^{n} u_{i,j} D_{i,j}$$

is much larger than the value of

$$\lambda \sum_{j=1}^{k} \sum_{i=1}^{n} u_{i,j} \log u_{i,j},$$

the within cluster dispersion term will a play a more important role to minimize (1). The clustering process turns to minimize the within cluster dispersion.

In the next two sections, we will present how to select suitable values of λ for numerical data clustering.

3.3 An Example

The properties of the proposed objective function can be demonstrated by the following example. A data set of 1,000 points in a two-dimensional (2D) space is shown Fig. 1a. We can see there are three clusters. We want to use the new algorithm to cluster this data set and discover the three clusters. We started with five initial cluster centers randomly selected from the data set, as shown in Fig. 1b. It happened that these initial seed centers were all selected from the same cluster. Apparently, this was not a good selection from the K-Means clustering point of view. Fig. 2 shows the clustering results of the data set in Fig. 1a by the algorithm with different λ input values. We can see that when λ is very small, the number of clusters generated by the algorithm was equal to the number of initial cluster centers. As λ increased, the number of generated clusters reduced because some initial cluster centers moved to the same locations. As λ increased to certain level, the number of generated clusters was same as the number of the true clusters in the data set. This indicates that the λ setting was right in finding the true clusters by the algorithm. However, as λ further increased, the number of generated clusters became smaller than the number of the true clusters in the data set. Finally, when λ increased to a certain value, the

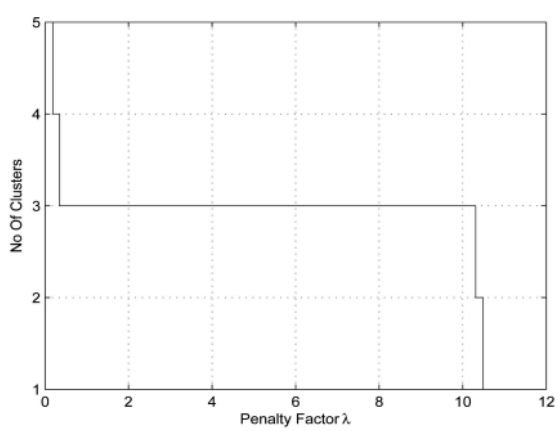


Fig. 2. The numbers of merged cluster centers with respect to different values of λ.

number of generated clusters became one. This indicates that the negative entropy term fully dominated the clustering process.

For instance, when λ is equal to 1 in Fig. 2, Figs. 1c, 1d, 1e, and 1f show the movements of the cluster centers in the subsequent iterations. The clustering process stopped at the eighth iteration. In Fig. 1f, we can see that the five initial cluster centers moved to three locations, which are very close to the three true cluster centers in the data set. The positions of the identified cluster centers by the clustering process and the true cluster centers are given in the table below.

	The positions of true cluster centers	The positions of determined cluster centers
Cluster 1	(1,5)	(0.9817,4.9809)
Cluster 2	(1,1)	(1.0079,0.9756)
Cluster 3	(5,5)	(5.0169,4.9998)

Repeated experiments with different numbers of initial cluster centers produced the same clustering result, as shown in Fig. 1f.

The above experiments show that by introducing the negative entropy term to the K-Means objective function and by using different values of λ in the clustering process, we can discover the centers of true clusters on the data set but also find the correct number of the true clusters. These are the two well-known problems in K-Means clustering stated in Section 1, which can be solved by the new algorithm.

The agglomerative fuzzy K-Means algorithm has the following advantages:

1. It is not sensitive to the initial cluster centers. This algorithm can improve the traditional K-Means algorithm by generating more consistent clustering results in different clustering runs. These consistent clustering results can be effectively used with cluster validation techniques to determine the number of clusters in a data set.
2. It is not necessary to know the exact number of clusters in advance. The number of clusters will be determined by counting how many finally merged cluster centers. For instance, as shown in Fig. 1f, the number of merged cluster centers is three. This is exactly the same as the number of "true" clusters in the data set.

3.4 The Overall Implementation

The overall algorithm is implemented in the framework shown in Fig. 3, which automatically run agglomerative fuzzy K-Means algorithm to discover the "best" number of clusters. In the implementation, there is only one input parameter, the number of initial cluster centers $k^\star$. This input number should be larger than the possible number of clusters in the given data set.

There are two loops in the implementation. In the first loop, we find the penalty factor $\lambda_{\min}$ such that the agglomerative fuzzy K-Means algorithm will produce exactly $k^\star$ clusters in the output. The first loop guarantees the "best" number of clusters will not be missed. In the second loop, the number of clusters k is changed in an decreasing order while λ changes in an increasing order. We consider that the values of λ increase from $\lambda_{\min} : \lambda := \lambda_{\min} \times t$, where $t = 2, 3, \ldots$, and run the agglomerative fuzzy K-Means algorithm for each λ. In the loop, the generated cluster centers are checked and the k_{share} cluster centers, which share the same locations with other cluster centers, are removed. Therefore, the output number of clusters become $k = k^\star - k_{share}$. The whole procedure is stopped when k is equal to 1, i.e., the value of λ is large enough such that all the objects associate to one merged cluster center.

In using this algorithm, when the number of clusters k stays unchanged in a few iterations, where λ has increased a few times, this indicates that the right number of clusters may have been found. For example, in Fig. 2, the number of clusters stays 3 as λ has changed from 2 to 10. This indicates that 3 is the true number of clusters in the data set.

In this iterative process, we further add a cluster validation step to validate the clustering result, where a cluster that shares its center with other clusters is identified. A cluster validation index will be defined and studied in Section 4. This loop stops when $k = 1$, i.e., all cluster centers have moved to the same location. The output of the implementation is the clustering results with the least validation index value.

4 Experimental Results

In this section, we present five experiments to show the effectiveness of the proposed algorithm. The first three experiments were conducted on synthetic data sets containing overlapping clusters. The last experiment used a real data set. The experiment results demonstrated that starting with different numbers of initial cluster centers, the algorithm was able to consistently discover the genuine clusters.

In the experiments, we used a validation index proposed by Sun et al. [20]. This validation index is constructed based on the average scattering (compactness) of clusters and distances (separations) between clusters. However, we would like to remark that other validation indices can also be used in our framework since the proposed algorithm can provide more consistent and effective clustering solutions in different clustering runs for cluster validation. The validation index [20] is given as

$$\mathcal{V}(U, Z, k) = \mathcal{SCATTER}(k) + \frac{\mathcal{DISTANCE}(k)}{\mathcal{DISTANCE}(k_{max})},$$

where

$$\mathcal{SCATTER}(k) = \frac{\frac{1}{k}\sum_{i=1}^{k} \|\sigma(z_i)\|}{\|\sigma(X)\|}$$

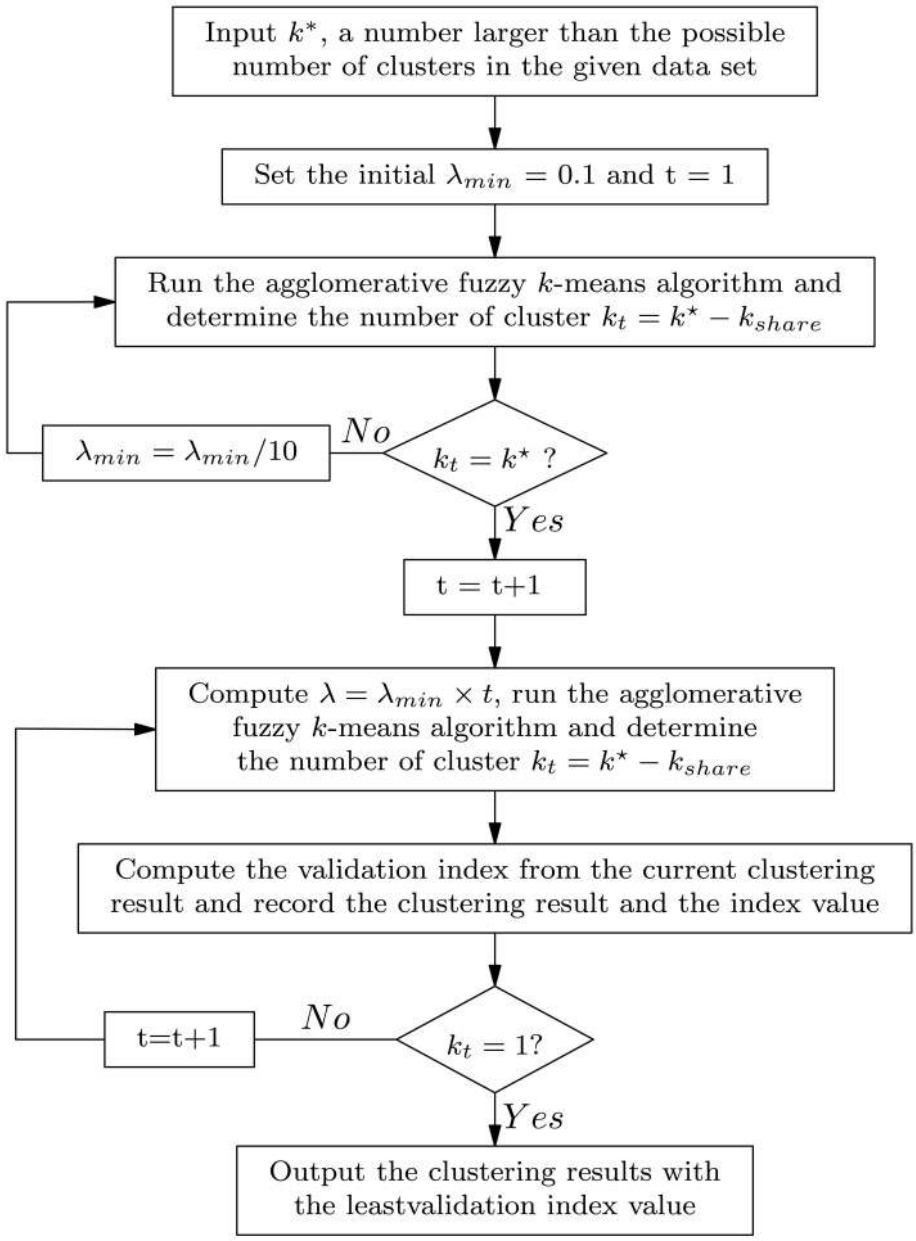


Fig. 3. The flowchart of the overall implementation.

shows the average scattering of the clusters. Here,

$$\sigma(X) = [\sigma_1(X), \sigma_2(X), \ldots, \sigma_m(X)]^T,$$
$$\sigma_j(X) = \frac{1}{n}\sum_{i=1}^{n}(x_{i,j} - \bar{x}_j)^2,$$
$$\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{i,j},$$
$$\sigma(z_l) = [\sigma_1(z_l), \sigma_2(z_l), \ldots, \sigma_m(z_l)]^T, \text{ and}$$
$$\sigma_m(z_l) = \frac{1}{n}\sum_{i=1}^{n} u_{i,l}(x_{i,m} - z_{l,m})^2.$$

When the number of clusters k is large, the value of $\mathcal{SCATTER}(k)$ is small. The second term $\mathcal{DISTANCE}(k)$ measures the separation between clusters:

$$\mathcal{DISTANCE}(k) = \frac{d_{max}^2}{d_{min}^2}\sum_{i=1}^{k}\sum_{j=1}^{k}\frac{1}{\|z_i - z_j\|^2},$$
$$d_{min} = \min_{i\neq j}\|z_i - z_j\| \text{ and } d_{max} = \max_{i\neq j}\|z_i - z_j\|.$$

We note that the smaller the $\mathcal{V}$, the better the clustering result.

In addition, the performance metric to evaluate clustering results is the Rand index. It measures how similar the partitions of objects are according to the real clusters (A) and a clustering result (B). Denote a and b as the number of object pairs that are in the same cluster in both A and B and in the same cluster in A but not B, respectively. The Rand index is defined as follows:

$$\mathcal{RAND}(k) = \frac{a+b}{n},$$

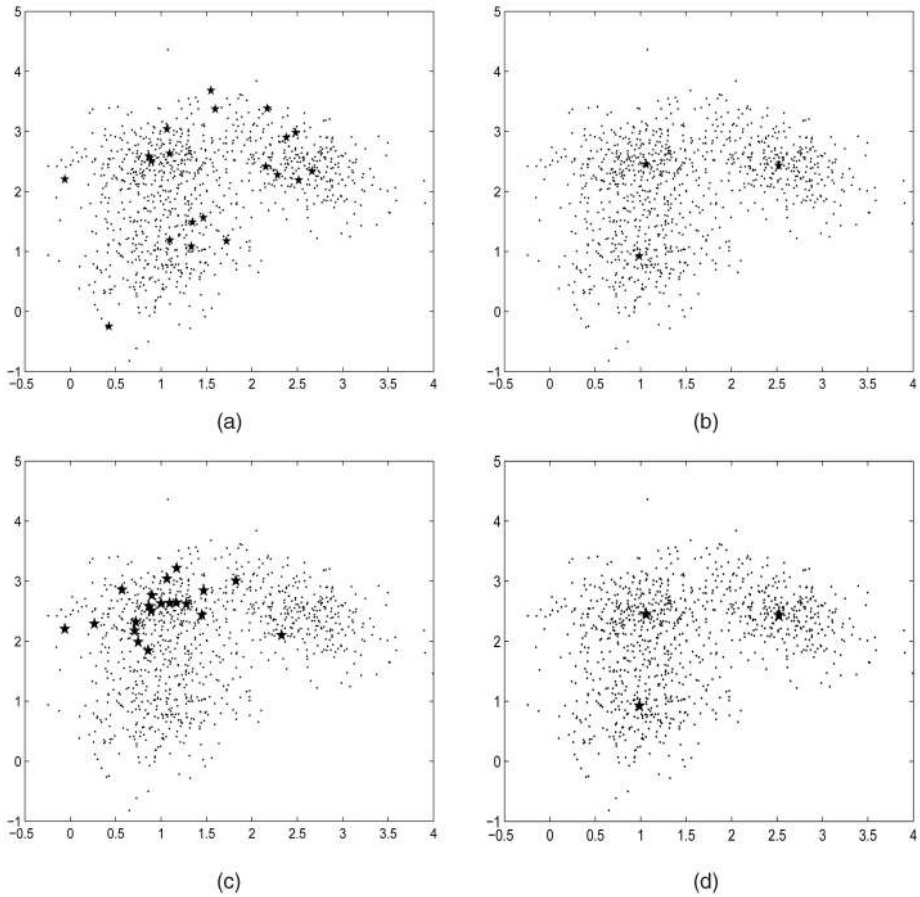


Fig. 4. The clustering results by the proposed algorithm (a) a random chosen initial seed centers, (b) the final positions of the merged cluster centers from (a), (c) a bad initial seed centers, and (d) the final positions of the merged cluster centers from (c).

where n is the total number of objects. We note that the larger the $\mathcal{RAND}$, the better the clustering result.

4.1 Experiment 1

The example in Fig. 1 has shown that starting from different initial cluster centers, the algorithm was able to move the initial cluster centers to the center locations of the genuine clusters in the date set with some initial centers merged to the same genuine centers. However, the three clusters are well separated without overlapping in the data set.

In this experiment, we investigated the performance of the agglomerative fuzzy K-Means algorithm in clustering data with overlapping and nonspherical clusters. We generated 1,000 synthetic data points from a mixture of three bivariate Gaussian densities given by

$$\begin{aligned} &0.33\ \text{Gaussian}\left[\begin{pmatrix}1.0\\1.0\end{pmatrix},\begin{pmatrix}0.20 & 0.05\\0.05 & 0.30\end{pmatrix}\right]\\ &+0.34\ \text{Gaussian}\left[\begin{pmatrix}1.0\\2.5\end{pmatrix},\begin{pmatrix}0.20 & 0.00\\0.00 & 0.20\end{pmatrix}\right]\\ &+0.33\ \text{Gaussian}\left[\begin{pmatrix}2.5\\2.5\end{pmatrix},\begin{pmatrix}0.20 & -0.10\\-0.10 & 0.30\end{pmatrix}\right], \end{aligned}$$

where $\text{Gaussian}[X, Y]$ is a Gaussian normal distribution with the mean X and the covariance matrix Y. The generated data set by this density function is shown in Fig. 4a. In the experiment, we randomly allocated 20 initial centers shown as stars in Fig. 4a. Here, the number of the initial centers is much larger than the number of the genuine clusters in the data set. After several iterations, the algorithm merged the initial centers to three locations, as shown in Fig. 4b. These three locations are very close to the "real" centers of the three genuine clusters in the data set.

Fig. 5 shows the results of the same data set with different values of λ and the same initial centers in Fig. 4a. From these results, we can observe that the number of the merged clusters reduced as λ increased. Fig. 6 shows the relationship between the number of the merged cluster centers and λ. We can see that the number of the merged clusters approached 1 as λ increased to 2. However, when the number of the merged cluster centers is equal to the number of the clusters in the data set, it kept the same for a long range of λ values. Therefore, this long range can be used as an indicator for the right number of clusters if the number of clusters in a data set is not known.

We also tested with different sets of initial cluster centers and found that when the number of the merged cluster centers became close to the number of clusters in the data set, it kept unchanged for a big range of λ values. Figs. 4a and 4c show two different sets of initial cluster centers for the same data set. Figs. 4b and 4d show their corresponding final locations of the merged cluster centers. The locations of the three clusters and the merged clusters from two different sets of initial cluster centers are given in the table below. We can see that they are very close. This implies that the "real" locations of the cluster centers can be discovered

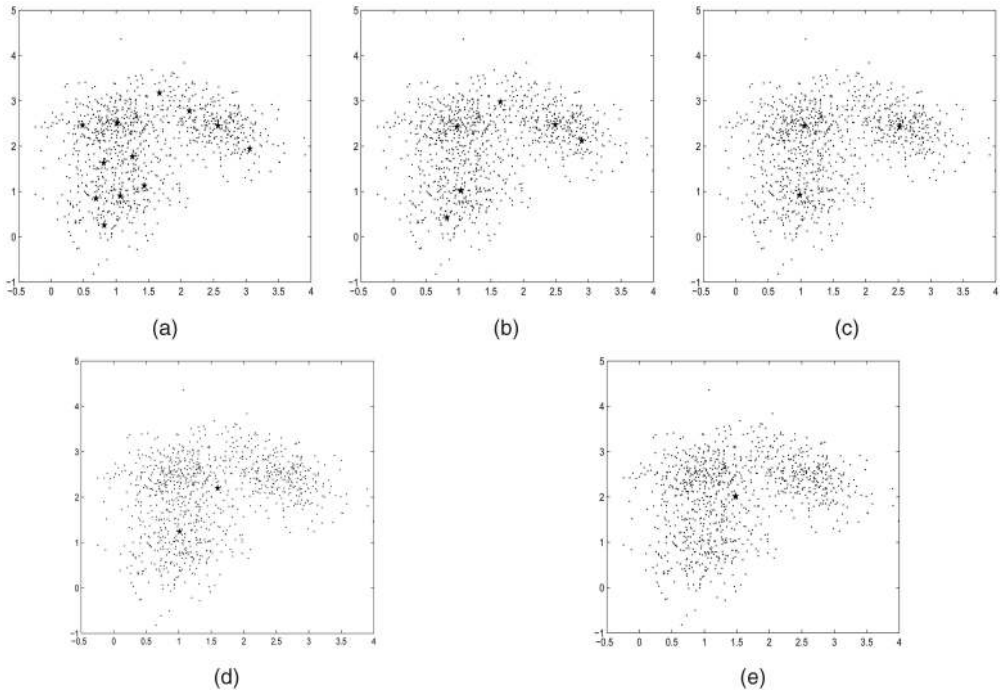


Fig. 5. The positions of the merged cluster centers (a) $\lambda = 0.27$, (b) $\lambda = 0.47$, (c) $\lambda = 0.63$, (d) $\lambda = 1.60$, and (e) $\lambda = 1.86$.

with the new algorithm, and the final result is not sensitive to the selection of the initial cluster centers.

	The positions of true cluster centers	The positions of determined cluster centers in Figure 4(b) using initial centers in Figure 4(a)	The positions of determined cluster centers in Figure 4(d) using initial centers in Figure 4(c)
Cluster 1	(2.5,2.5)	(2.5197,2.4327)	(2.5197,2.4327)
Cluster 2	(1,2.5)	(1.0603,2.4501)	(1.0603,2.4505)
Cluster 3	(1,1)	(0.9865,0.9252)	(0.9864,0.9252)

Cluster validation techniques can be further applied to these clustering results to verify the number of clusters in the data set. Fig. 7a shows the values of $\mathcal{V}$ with respect to different values of λ. The numbers in the brackets refer to the number of the merged cluster centers for a given value of λ. As the data set was heavily overlapped, the proposed algorithm selected many possible numbers of clusters in the data set (cf., (6)). We find in Fig. 7a that the case of the three merged clusters give the smallest value of $\mathcal{V}$. Similarly, we show in Fig. 7b the values of $\mathcal{RAND}$ with respect to different values of λ. Again, it is clear that the case of three merged clusters give the largest value of $\mathcal{RAND}$. Both indices support that the data set contains three clusters. The proposed algorithm determined the number of clusters accurately.

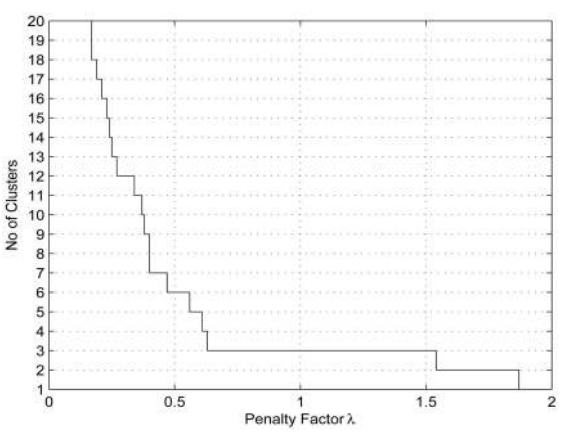


Fig. 6. The numbers of merged cluster centers with respect to different values of λ.

4.2 Experiment 2

In this experiment, we randomly generated synthetic data sets with different numbers of dimensions (2/3/4/5), objects (500/1,000/5,000), and clusters (3/5/7) and different degrees of overlapping between clusters. Each dimension of each group of data sets was generated as the normal distribution with the controlled standard derivation σ (the standard derivation of the distance between each object value and its assigned center value in a dimension). The number of objects in each cluster was the same in each generated data set. We designed two cases of overlapping between two clusters, namely, 1) well separated ($\delta = 6\sigma$) and 2) heavily overlapped ($\delta = 3\sigma$). Here, δ refers to the distance between two centers values in a dimension. For the well-separated case, there were no overlapping points between clusters (cf., Fig. 8). For the overlapping case, points in different clusters were overlapped (cf., Fig. 10). Algorithm 1 gives the description for synthetic data generation.

Algorithm 1. Synthetic data generation

Specify the number of cluster k, objects n, dimensions m, the standard derivation of the clusters σ, the distance δ between two centers (3σ or 6σ in our tested data sets), and the output a set of objects $\boldsymbol{X} = \{X_1, X_2, \ldots, X_n\}$

Set num to be the smallest integer being greater than or equal to n/k

{Randomly choose the centers}

Randomly choose the first center z_1

for $j = 2$ to k **do**

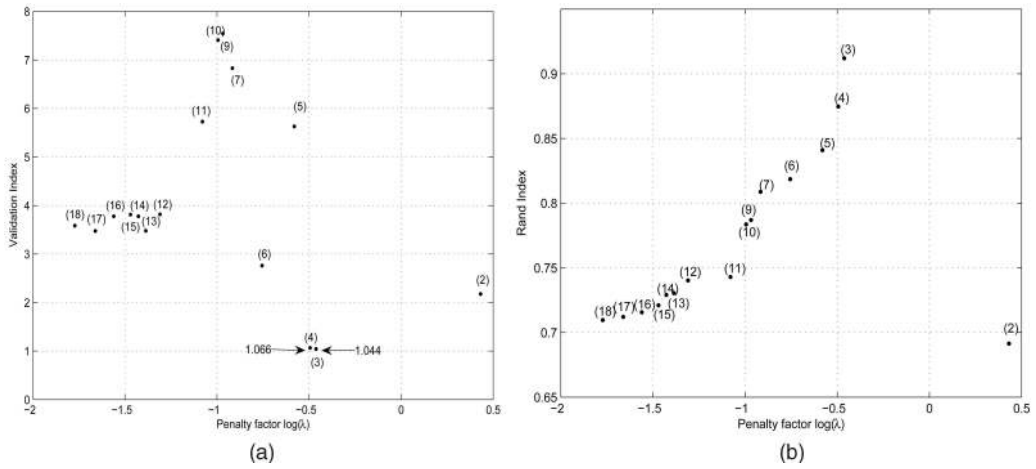


Fig. 7. The evaluation results by the proposed algorithm (a) the index $\mathcal{V}$ and (b) the index $\mathcal{RAND}$ for Experiment 1.

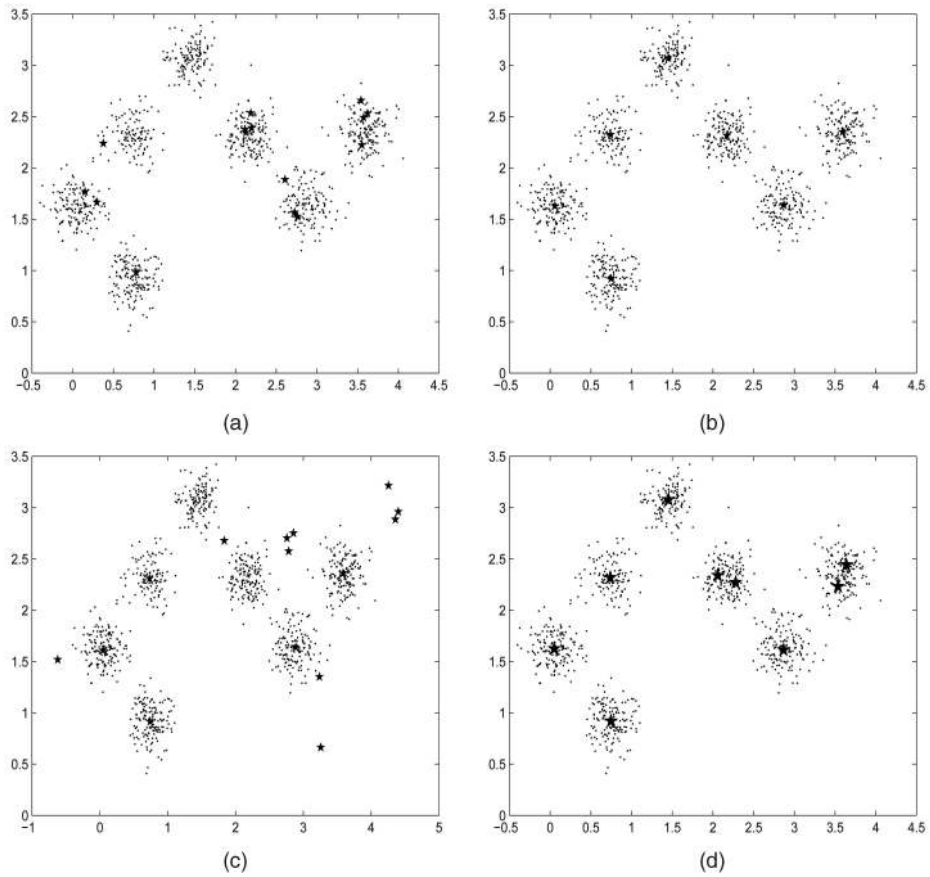


Fig. 8. (a) The initial seed centers, (b) the final positions of the merged cluster centers by the proposed algorithm, (c) the final positions of cluster centers by the rival penalized competitive learning algorithm, and (d) the final positions of the cluster centers by the classical fuzzy K-Means algorithm.

Choose the center z_j such that for each dimension l, $|z_{j,l} - z_{j-1,l}| = \delta$ and $|z_{i,l} - z_{j,l}| \geq \delta$ for $i > j$

end for

{Generate about num objects for each cluster}

for $i = 1$ to n **do**

Set j to be the smallest integer being greater than or equal to i/num

for $l = 1$ to m **do**

Set a random number r is generated by a Gaussian distribution with the mean zero and the standard derivation one

$x_{i,l} = z_{j,l} + r * \sigma$

end for

end for

In conducting this experiment, the number of the initial cluster centers was set to 15 (the number larger than the number of clusters in the generated data sets). There are 10 runs of the agglomerative fuzzy K-Means algorithm. Each run used different randomly generated initial cluster centers. In each data set, we checked the proposed

TABLE 1
The Degree of Overlapping between Two Clusters Is 6σ

Dimensions	Objects	k_{true}	k_{new}	$k_{classical}$
2	500	3	3	3,4
		5	5	7
		7	7	9,11,13
	1000	3	3	5,6
		5	5	7,9
		7	7	9
	5000	3	3	3
		5	5	6,7,9
		7	7	8,10,13
3	500	3	3	3
		5	5	5,11,12
		7	7	11
	1000	3	3	3
		5	5	7,8,10
		7	7	8
	5000	3	3	4
		5	5	5,11,12
		7	7	11
4	500	3	3	3
		5	5	6
		7	7	11
	1000	3	3	4
		5	5	10
		7	7	8
	5000	3	3	4,6,12
		5	5	7
		7	7	12
5	500	3	3	13,14
		5	5	13
		7	7	13
	1000	3	3	14
		5	5	10
		7	7	14
	5000	3	3	7,9,10
		5	5	14
		7	7	12

TABLE 2
The Degree of Overlapping between Two Clusters Is 3σ

Dimensions	Objects	k_{true}	k_{new}	$k_{classical}$
2	500	3	3	3,4
		5	5	8
		7	7,8	9,11,13
	1000	3	3	3,5
		5	5	5
		7	7	5,6,7,8,9
	5000	3	3	4
		5	5	8
		7	7	5,8,11
3	500	3	3	5
		5	5	6
		7	5,7	12
	1000	3	3	5
		5	5	8
		7	7	10
	5000	3	3	3
		5	5	6
		7	7	5
4	500	3	3	4
		5	5	7
		7	7	11
	1000	3	3	4
		5	5	8
		7	7	7,9,11
	5000	3	3	5
		5	5	6
		7	7	8,11
5	500	3	3	13
		5	5	7,12
		7	8	12
	1000	3	3	13
		5	5	12,14
		7	7	8
	5000	3	3	12,14
		5	5	13
		7	7	14

algorithm a long range of λ values keeping the same number of merged cluster centers to estimate the number of true clusters. We also employed the index $\mathcal{V}$ to validate the best clustering result and the estimate of the number of the true clusters by the proposed algorithm.

For comparison, we used the classical fuzzy K-Means algorithm to generate the clustering results. We remark that the objective function in the classical fuzzy K-Means algorithm is the same as (1) except without the second entropy term. Different values of k to the classical fuzzy

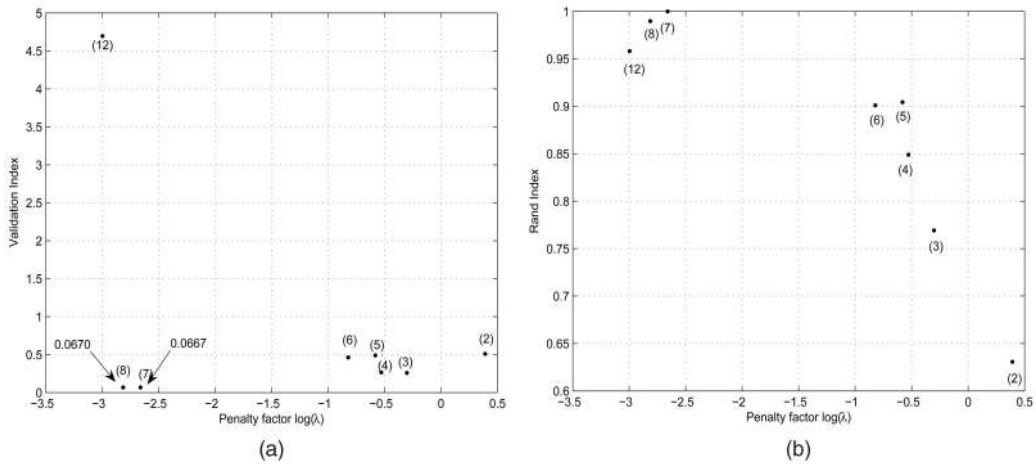


Fig. 9. The evaluation results by the proposed algorithm: (a) the index $\mathcal{V}$ and (b) the index $\mathcal{RAND}$ for Experiment 3.

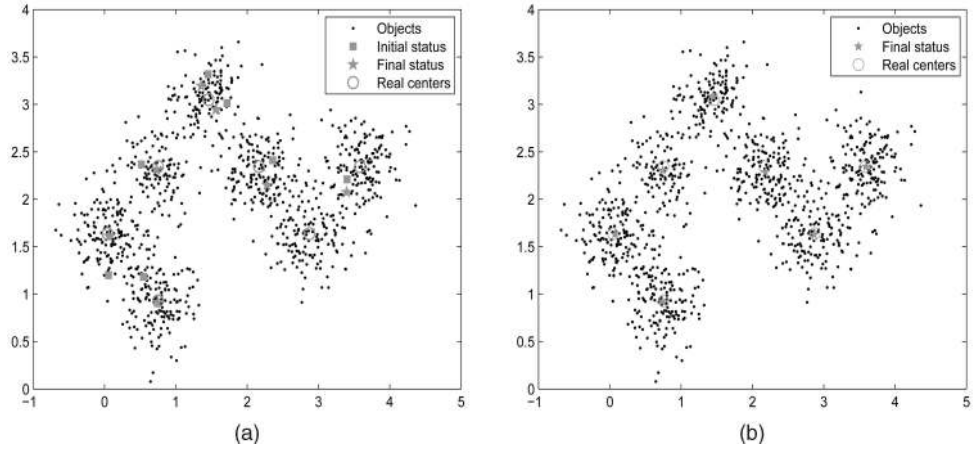


Fig. 10. (a) The clustering results (a) by using the "K-Means ++" procedure and (b) by using the proposed algorithm.

TABLE 3
The Determined Cluster Centers by the Proposed Algorithm, K-Means ++ Algorithm, and the Proposed Algorithm with "K-Means ++" Initialization Procedure

	The positions of true cluster centers	The positions of determined cluster centers using the proposed algorithm	The positions of determined cluster centers using k-means++ algorithm	The positions of determined cluster centers using the proposed algorithm with k-means++ initialization procedure
Cluster 1	(3.580,2.344)	(3.603,2.355)	(3.394,2.076)	(3.595,2.345)
Cluster 2	(2.873,1.637)	(2.881,1.634)	(2.279,2.149)	(2.881,1.604)
Cluster 3	(2.166,2.344)	(2.183,2.287)	(1.564,2.948)	(2.201,2.273)
Cluster 4	(1.459,3.051)	(1.452,3.083)	(1.356,3.196)	(1.453,3.082)
Cluster 5	(0.752,0.930)	(0.749,0.913)	(0.748,0.915)	(0.741,0.925)
Cluster 6	(0.752,2.344)	(0.756,2.318)	(0.730,2.312)	(0.756,2.317)
Cluster 7	(0.044,1.637)	(0.092,1.626)	(0.061,1.613)	(0.076,1.637)
Clustering Accuracy		95.0 %	82.1 %	95.0 %

K-Means algorithm were tested. For each k ($k = 2, 3, \ldots, 15$), we performed 10 runs with different initial cluster centers randomly generated. We found that the clustering results of the classical fuzzy K-Means algorithm were quite different in different runs. The validation index $\mathcal{V}$ was used to determine the number of clusters generated by the classical fuzzy K-Means algorithm in the data set.

Tables 1 and 2 list the true number k_{true} of clusters in the generated data sets, the number k_{new} of the merged clusters found by the agglomerative fuzzy K-Means algorithm that is most frequently generated by the algorithm, and the number $k_{classical}$ of clusters found by the classical fuzzy K-Means algorithm. Here, $k_{classical}$ refers to the result (among 10 runs) selected from the minimum validation

TABLE 4
Clustering Results for Different Algorithms Using the Forgy and Random Partition Initializations

	$\sqrt{KM}$		Clusters found	
	Foryg	Random Partition	Forgy	Random Partition
KM	198.58	242.03	95	79
H1	207.68	262.65	92	74
FZKM	208.05	218.07	93	90
H2	196.92	205.82	96	92
KHM	194.43	215.60	97	88
Proposed	189.62	191.53	100	102

index $\mathcal{V}$ in each run. We can see in Table 1 that the proposed algorithm performed very well, and the performance of the classical fuzzy K-Means algorithm was slightly poor. For some data sets, different k numbers could be selected from the results generated by the classical fuzzy K-Means algorithm according to the minimum validation index $\mathcal{V}$. If the true number of clusters were not known, it would be difficult to determine which k was right. The proposed algorithm did not have such problem.

For heavily overlapping data sets, the performance of the proposed algorithm was much better than that of the classical fuzzy K-Means algorithm. In particular, the determined numbers of clusters were more accurate than those by the classical fuzzy K-Means algorithm. The number of correctly determined clusters by the proposed algorithm was 35 out of 36. For the incorrect case, the number of determined clusters by the proposed algorithm was 8, which was very close to the true number 7 in the data set (the case of the number of dimensions = 5, the number of objects = 500, and the number of true clusters = 7). The number of correctly determined clusters by the classical fuzzy K-Means algorithm was 5 out of 36.

These results show the effectiveness of the proposed algorithm when clustering data with overlapping clusters and the consistency of the clustering results from different initial cluster centers.

4.3 Experiment 3

In this experiment, we show an example to demonstrate the usefulness of the proposed algorithm. We consider the data set (the number of dimensions = 2, the number of objects = 1, 000, and the number of clusters = 7) in Table 1. Fig. 8a shows the data set and the initial cluster centers. In the test, we compared the proposed algorithm with the rival penalized competitive learning algorithm [30]. The number of initial cluster centers randomly generated was set to 15 in the two

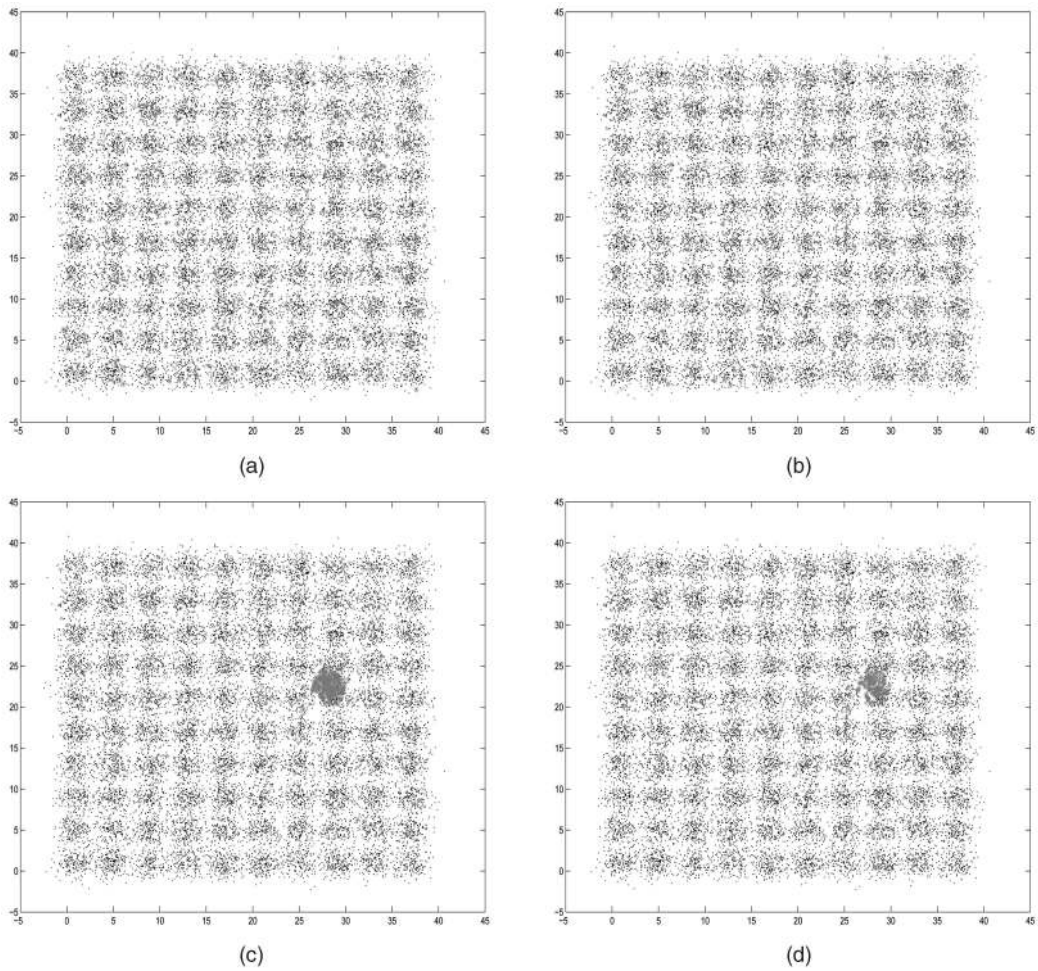


Fig. 11. (a) The Forgy initialization for the proposed algorithm. (b) The Forgy initialization for the other algorithms. (c) The random partition initialization for the proposed algorithm. (d) The random partition initialization for the other algorithms.

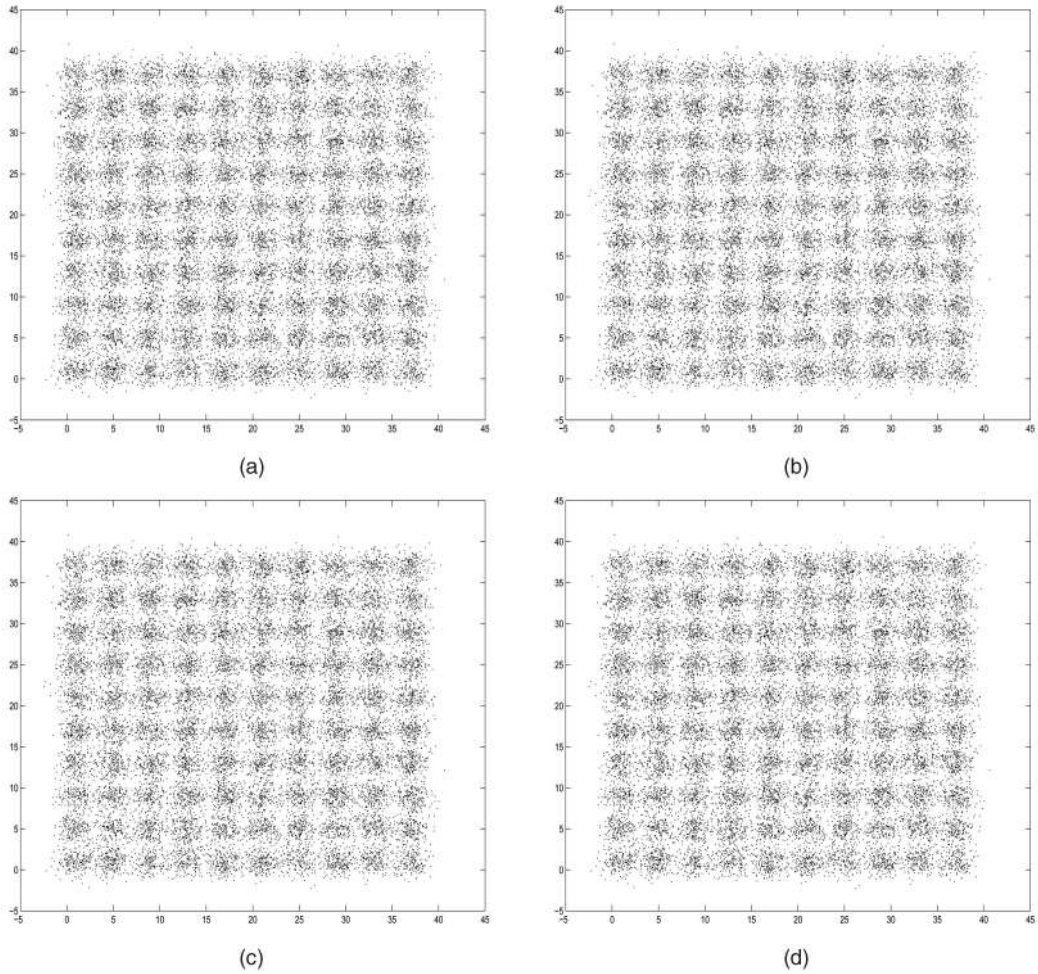


Fig. 12. The clustering results of the proposed algorithm using (a) the Forgy initialization and (b) the random partition initialization. The clustering results of KHM using (c) the Forgy initialization and (d) the random partition initialization.

algorithms. Each algorithm was run 100 times. The proposed algorithm showed consistent results in 100 runs on the final locations of the merged cluster centers, as shown in Fig. 8b. It is clear from the figure that the number of the merged cluster centers is 7, which is exactly the same as the number of the true clusters. Both validation indices $\mathcal{V}$ and $\mathcal{RAND}$ have the smallest and the largest values, respectively, when the number of the merged clusters was 7 (see Fig. 9). However, the rival penalized competitive learning algorithm was sensitive to the initial cluster centers. The best result produced by this algorithm is shown in Fig. 8c. We can see from the figure that one cluster center is situated between two clusters. In Fig. 8d, we show the clustering results by the classical fuzzy K-Means algorithm. The best clustering result is attained when the number of clusters is 9. We can see from the figure that one cluster contains two centers. However, these two cluster centers were not merged together. We cannot interpret them as a single cluster. In our proposed algorithm, we moved the merged cluster centers to the same cluster and grouped all the objects in the corresponding clusters as one cluster.

4.4 Experiment 4

In this experiment, we demonstrate the insensitive capability of the proposed algorithm to the centers' initialization. David et al. [16] provide a new algorithm, "K-Means ++," to estimate better initial centers for K-Means algorithms. We consider the data set (the number of dimensions = 2, the number of objects = 1, 000, and the number of clusters = 7) in Table 2. The data set is shown, as in Fig. 10. The clustering result of the proposed algorithm for this data set in shown, as in Fig. 10b. We remark that the number of initial cluster centers is set to 15 (cf., Experiment 2). It is clear from the figure that the merged cluster centers match the locations of the true centers. We also employ the "K-Means ++" algorithm for this data set. In this test, we assume the number of clusters is known (i.e., $k = 7$) and would like to check the performance of the initialization procedure. The clustering result is shown, as in Fig. 10a. We see from the figure that two initial centers determined by the "K-Means ++" procedure are located in the same cluster, and there is one true center that cannot be identified by the final centers. In Table 3, we summarize the clustering results of the proposed algorithm and the "K-Means ++" algorithm and further add the clustering result by using the proposed algorithm with the "K-Means ++" initialization procedure (i.e., using the initial seed centers generated by the "K-Means ++" algorithm only). In the table, the clustering accuracy refers to the percentage of objects in the data set that are correctly clustered together. According to the table, it is clear that the performance of the proposed

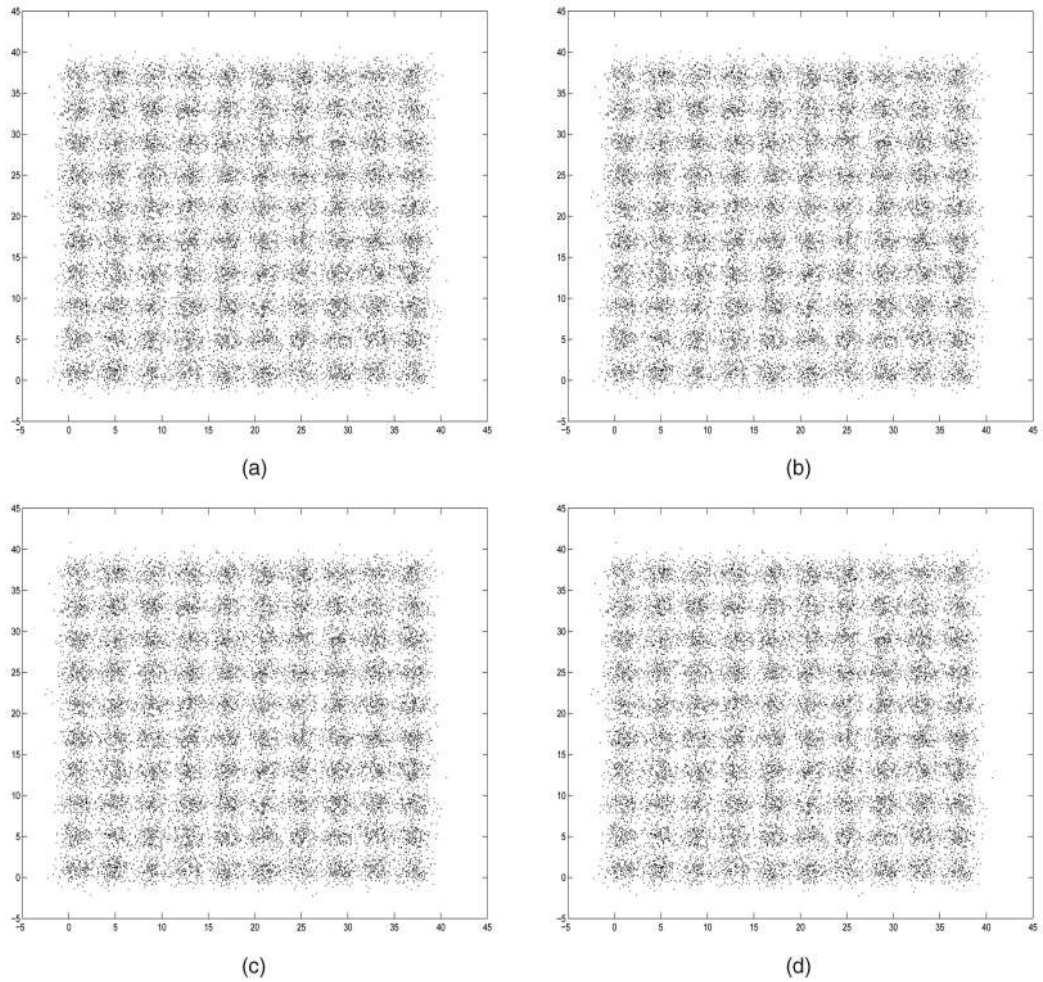


Fig. 13. The clustering results of H2 using (a) the Forgy initialization and (b) the random partition initialization. The clustering results of H1 using (c) the Forgy initialization and (d) the random partition initialization.

algorithm is better than that of the "K-Means ++" algorithm. We also find that the initialization procedure added to the proposed algorithm does not further improve the clustering result.

4.5 Experiment 5

In this experiment, we further investigate the insensitive capability of the proposed algorithm to the centers' initialization. We compare it with other algorithms: k-harmonic means algorithm (KHM) by Zhang et al. [38] and its two new variants (H1 and H2) by Hamerly and Elkan [39]. These algorithms are quite insensitive to initial cluster centers. Here, we used the BIRCH data set [37], which has 100 true clusters arranged in a 10×10 2D grid. Each clusters contains 200 objects generated by Gaussian distributions. The total number of objects is 20,000. The distance between two adjacent generated data cluster centers is $4\sqrt{2}$. The variance of Gaussian distribution is 1. In the tests, we employed two initial cluster centers. One method is called the Forgy initialization [39], and the other method is random partition initialization. These two methods have been tested in [39]. In the proposed algorithm, we set the number of initial cluster centers to be 200. For the other clustering algorithms, we assign the number of initial cluster centers to be 100, which is the same as the number of clusters in the BIRCH data set. As a comparison, we also test K-Means algorithm (KM) and fuzzy K-Means algorithm (FZKM). In Table 4, clustering results for different algorithms using the two different initializations are presented. In the table, the square root of the K-Means objective function values $\sqrt{KM}$ from the output of different algorithms and the number of clusters found by different algorithms are listed. We see from the table that the proposed algorithm is competitive with k-harmonic means algorithm and its two variants. Fig. 11 shows the two initializations of the proposed algorithm and the other algorithms. Figs. 12, 13, and 14 show the clustering results of different algorithms using the two initializations. We see from the figures that the proposed algorithm performs quite well.

4.6 Experiment 6

In this experiment, we used the WINE data set obtained from the UCI machine Learning Repository. This data set represents the results of chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. As such, each data point was labeled as one of the three cultivars. The WINE data set consists of 178 records, each being described by 13 attributes.

We carried out 10 runs of the proposed algorithm and also 10 runs of the classical fuzzy K-Means algorithm with different initial cluster centers. We found that the proposed algorithm with the validation index $\mathcal{V}$ can find the corrected number of clusters, i.e., three clusters. The average value of

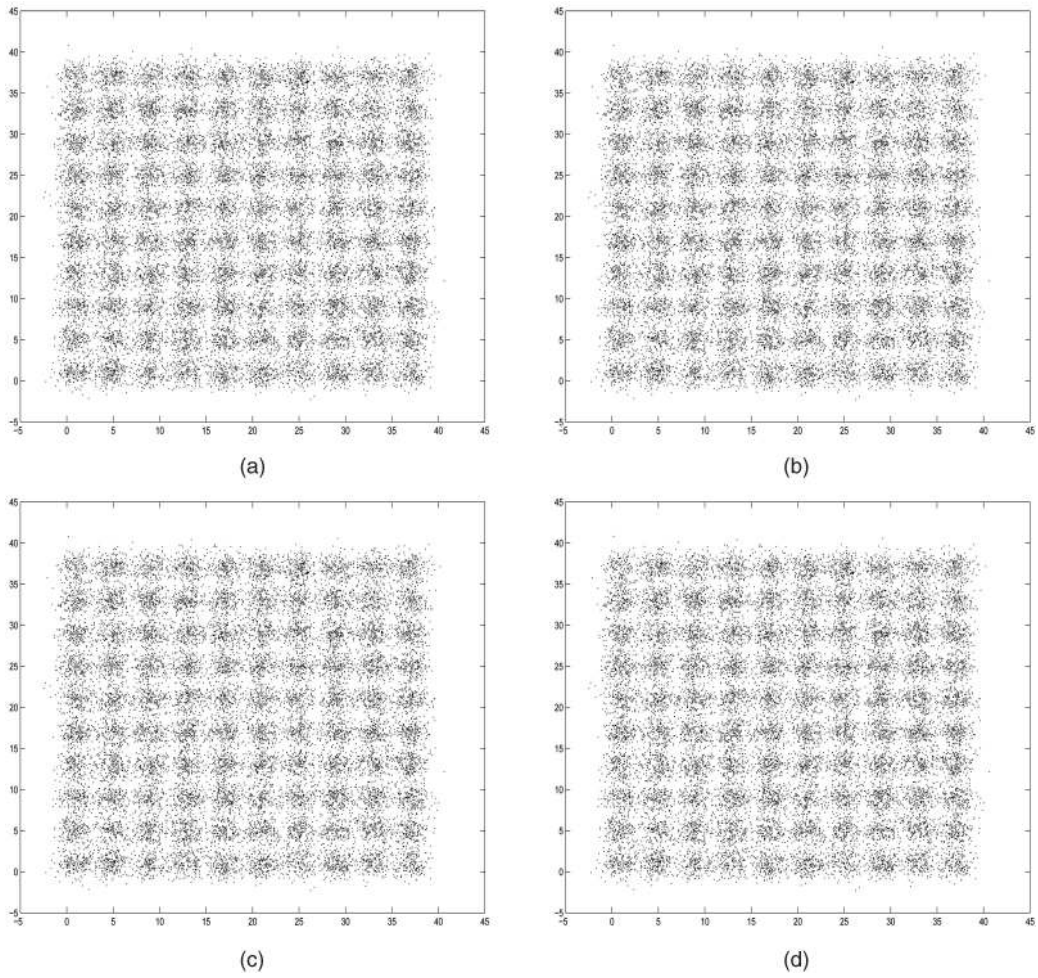


Fig. 14. The clustering results of FZKM using (a) the Forgy initialization and (b) the random partition initialization. The clustering results of KM using (c) the Forgy initialization and (d) the random partition initialization.

the $\mathcal{RAND}$ index is 0.931. On the other hand, using the validation index $\mathcal{V}$, the classical fuzzy K-Means algorithm usually produced the numbers of clusters as 5, 9, or 12 in 10 runs. The corresponding average value of the $\mathcal{RAND}$ index is 0.694, which is significantly smaller than that by the proposed algorithm. Even if we correctly found three clusters in one of 10 runs by the fuzzy K-Means algorithm, the $\mathcal{RAND}$ index value is only 0.720, which is still inferior to the proposed algorithm.

5 Concluding Remarks

In this paper, we have presented a new approach, called the agglomerative fuzzy K-Means clustering algorithm for numerical data to determine the number of clusters. The new approach minimizes the objective function, which is the sum of the objective function of the fuzzy k-mean and the entropy function. The initial number of clusters is set to be larger than the true number of clusters in a data set. With the entropy cost function, each initial cluster centers will move to the dense centers of the clusters in a data set. These initial cluster centers are merged in the same location, and the number of the determined clusters is just the number of the merged clusters in the output of the algorithm. Our experimental results have shown the effectiveness of the proposed algorithm when different initial cluster centers were used and overlapping clusters are contained in data sets.

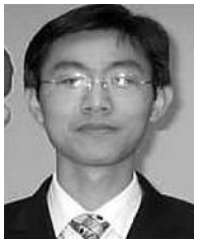
Acknowledgments

The work in this paper was supported by the Research Grant Council of Hong Kong SAR under Projects: 7045/04P, 7045/05P, HKBU 2156/04E, and HKBU 210306, and the Faculty Research Grant of Hong Kong Baptist University under Project: HKBU 05-06/II-42, and supported by the Natural Science Foundation of China under Grant 60603066.

References

[1] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data.* Prentice Hall, 1988.
[2] M.R. Anderberg, *Cluster Analysis for Applications.* Academic, 1973.
[3] G.H. Ball and D.J. Hall, "A Clustering Technique for Summarizing Multivariate Data," *Behavioral Science,* vol. 12, pp. 153-155, 1967.
[4] E.R. Ruspini, "A New Approach to Clustering," *Information Control,* vol. 19, pp. 22-32, 1969.
[5] J.B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. Fifth Symp. Math. Statistics and Probability,* vol. 1, AD 669871, pp. 281-297, 1967.
[6] J.C. Bezdek, "A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 2, pp. 1-8, 1980.

[7] F. Höppner, F. Klawonn, and R. Kruse, "Fuzzy Cluster Analysis: Methods for Classification," *Data Analysis, and Image Recognition.* Wiley, 1999.
[8] R. Krishnapuram, H. Frigui, and O. Nasraoui, "Fuzzy and Possibilistic Shell Clustering Algorithms and Their Application to Boundary Detection and Surface Approximation—Part I and II," *IEEE. Trans. Fuzzy Systems,* vol. 3, no. 4, pp. 29-60, 1995.
[9] F. Hoppner, "Fuzzy Shell Clustering Algorithms in Image Processing: Fuzzy c-Rectangular and 2-Rectangular Shells," *IEEE. Trans. Fuzzy Systems,* vol. 5, no. 4, pp. 599-613, 1997.
[10] G. Hamerly and C. Elkan, "Learning the k in k-Means," *Proc. 17th Ann. Conf. Neural Information Processing Systems (NIPS '03),* Dec. 2003.
[11] Z. Huang, M. Ng, H. Rong, and Z. Li, "Automated Variable Weighting in k-Means Type Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 5, pp. 657-668, May 2005.
[12] G.P. Babu and M.N. Murty, "A Near-optimal Initial Seed Value Selection for K-Means Algorithm Using Genetic Algorithm," *Pattern Recognition Letters,* vol. 14, pp. 763-769, 1993.
[13] K. Krishna and M.N. Murty, "Genetic K-Means Algorithm," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 29, no. 3, 1999.
[14] M. Laszlo and S. Mukherjee, "A Genetic Algorithm Using Hyper-Quadtrees for Low-Dimensional K-Means Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 4, pp. 533-543, Apr. 2006.
[15] M. Laszlo and S. Mukherjee, "A Genetic Algorithm That Exchanges Neighboring Centers for K-Means Clustering," *Pattern Recognition Letters,* vol. 28, no. 16, pp. 2359-2366, 2007.
[16] D. Arthur and S. Vassilvitskii, "K-Means++: The Advantages of Careful Seeding," *Proc. 18th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA '07),* pp. 1027-1035, 2007.
[17] G.W. Millligan and M.C. Cooper, "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Psychometrika,* vol. 50, pp. 159-179, 1985.
[18] R. Nikhil and C. James, "On Cluster Validity for the Fuzzy c-Means Model," *IEEE Trans. Fuzzy Systems,* vol. 3, no. 3, pp. 370-379, 1995.
[19] I. Gath and A. Geve, "Unsupervised Optimal Fuzzy Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 7, pp. 773-781, July 1989.
[20] H. Sun, S. Wang, and Q. Jiang, "FCM-Based Model Selection Algorithms for Determining the Number of Clusters," *Pattern Recognition,* vol. 37, pp. 2027-2037, 2004.
[21] M. Rezaee, B. Lelieveldt, and J. Reiber, "A New Cluster Validity Index for the Fuzzy c-Mean," *Pattern Recognition Letters,* vol. 19, pp. 237-246, 1998.
[22] H. Akaike, "A New Look at the Statistical Model Identification," *IEEE Trans. Automatic Control,* vol. 19, no. 6, pp. 716-722, 1974.
[23] G. Schwarz, "Estimating the Dimension of a Model," *Annals of Statistics,* vol. 6, pp. 461-464, 1978.
[24] B.G. Leroux, "Consistent Estimation of a Mixing Distribution," *Annals of Statistics,* vol. 20, pp. 1350-1360, 1992.
[25] W. Pan, "Booststrapping Likelihood for Model Selection with Small Samples," *J. Computational and Graphical Statistics,* vol. 8, pp. 225-235, 1999.
[26] S. Padhraic, "Model Selection for Probabilistic Clustering Using Cross-Validated Likelihood," *Statistics and Computing,* vol. 10, pp. 63-72, 2000.
[27] M. Windham and A. Cutler, "Information Ratios for Validating Mixture Analyses," *Statistical Assoc.,* vol. 87, pp. 1188-1192, 1992.
[28] H. Bozdogan, "Choosing the Number of Component Clusters in the Mixture-Model Using a New Informational Complexity Criterion of the Inverse-Fisher Information Matrix," *Information and Classification,* pp. 40-54, 1993.
[29] M. Geoffrey and P. David, *Finite Mixture Models,* pp. 202-207, 2000.
[30] Y. Cheung, "Maximum Weighted Likelihood via Rival Penalized EM for Density Mixture Clustering with Automatic Model Selection," *IEEE Trans. Knowledge and Data Eng.,* vol. 17, no. 6, pp. 750-761, June 2005.
[31] L. Xu, "Rival Penalized Competitive Learning, Finite Mixture, and Multisets Clustering," *Pattern Recognition Letters,* vol. 18, nos. 11-13, pp. 1167-1178, 1997.
[32] L. Xu, "How Many Clusters?: A Ying-Yang Machine Based Theory for a Classical Open Problem in Pattern Recognition," *Proc. IEEE Int'l Conf. Neural Networks (ICNN '96),* vol. 3, pp. 1546-1551, 1996.
[33] P. Guo, C.L. Chen, and M.R. Lyu, "Cluster Number Selection for a Small Set of Samples Using the Bayesian Ying-Yang Model," *IEEE Trans. Neural Networks,* vol. 13, no. 3, pp. 757-763, 2002.
[34] H. Frigui and R. Krishnapuram, "Clustering by Competitive Agglomeration," *Pattern Recognition,* vol. 30, no. 7, pp. 1109-1119, 1997.
[35] S. Miyamoto and M. Mukaidono, "Fuzzy c-means as a Regularization and Maximum Entropy Approach," *Proc. Seventh Int'l Fuzzy Systems Assoc. World Congress (IFSA '97),* vol. 2, pp. 86-92, 1997.
[36] W. Rand, "Objective Criteria for the Evaluation of Clustering Methods," *J. Am. Statistical Assoc.,* vol. 66, pp. 846-850, 1971.
[37] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: A New Data Clustering Algorithm and Its Applications," *Data Mining and Knowledge Discovery,* vol. 1, pp. 141-182, 1997.
[38] B. Zhang, M. Hsu, and U. Dayal, "K-Harmonic Means—A Data Clustering Algorithm," Technical Report HPL-1999-124, Hewlett-Packard Labs, 1999.
[39] G. Hamerly and C. Elkan, "Alternatives to the K-Means Algorithm That Find Better Clusterings," *Proc. 11th Int'l Conf. Information and Knowledge Management (IKM '02),* pp. 600-607, 2002.



Mark Junjie Li received the BE degree in electronic engineering from JINAN University, Guangzhou, China, in 2002 and the MSc degree in electronic commerce and Internet computing at the University of Hong Kong in 2005. He is currently working toward the PhD degree in the Department of Mathematics, Hong Kong Baptist University. His research interests include data mining, subspace clustering algorithm, bioinformatics, and business intelligence.

Michael K. Ng is a professor in the Department of Mathematics, Hong Kong Baptist University. As an applied mathematician, his main research interests include bioinformatics, data mining, operations research, and scientific computing. He has published and edited five books and published more than 140 journal papers. He is the principal editor of the *Journal of Computational and Applied Mathematics* and the associate editor of the *SIAM Journal on Scientific Computing*, *Numerical Linear Algebra with Applications*, the *International Journal of Data Mining and Bioinformatics*, and *Multidimensional Systems and Signal Processing*.

Yiu-ming Cheung received the PhD degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong, in 2000. Currently, he is an associate professor in the Department of Computer Science, Hong Kong Baptist University. His research interests include machine learning, information security, signal processing, pattern recognition, and data mining. He is the founding and present chair of the Computational Intelligence Chapter, IEEE, Hong Kong. Also, he is a senior member of the IEEE and the ACM. More details can be found at http://www.comp.hkbu.edu.hk/~ymc.

Joshua Zhexue Huang received the PhD degree from the Royal Institute of Technology, Sweden. He is the assistant director of the E-Business Technology Institute (ETI), University of Hong Kong. Before joining ETI in 2000, he was a senior consultant at the Management Information Principles, Australia, consulting on data mining and business intelligence systems. From 1994 to 1998, he was a research scientist at CSIRO, Australia. His research interests include data mining, machine learning, clustering algorithms, and Grid computing.