Pattern Recognition 71 (2017) 45–59

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Robust-learning fuzzy c-means clustering algorithm with unknown number of clusters

Miin-Shen Yang[a,*], Yessica Nataliani[a,b]

[a] Department of Applied Mathematics, Chung Yuan Christian University, Chung-Li 32023, Taiwan
[b] Department of Information Systems, Satya Wacana Christian University, Salatiga 50711, Indonesia

ARTICLE INFO

Article history:
Received 27 August 2016
Revised 12 May 2017
Accepted 20 May 2017
Available online 22 May 2017

Keywords:
Fuzzy clustering
Fuzzy c-means (FCM)
Robust learning-based schema
Number of clusters
Entropy penalty terms
Robust-learning FCM (RL-FCM)

ABSTRACT

In fuzzy clustering, the fuzzy c-means (FCM) algorithm is the most commonly used clustering method. Various extensions of FCM had been proposed in the literature. However, the FCM algorithm and its extensions are usually affected by initializations and parameter selection with a number of clusters to be given a priori. Although there were some works to solve these problems in FCM, there is no work for FCM to be simultaneously robust to initializations and parameter selection under free of the fuzziness index without a given number of clusters. In this paper, we construct a robust learning-based FCM framework, called a robust-learning FCM (RL-FCM) algorithm, so that it becomes free of the fuzziness index *m* and initializations without parameter selection, and can also automatically find the best number of clusters. We first use entropy-type penalty terms for adjusting bias with free of the fuzziness index, and then create a robust learning-based schema for finding the best number of clusters. The computational complexity of the proposed RL-FCM algorithm is also analyzed. Comparisons between RL-FCM and other existing methods are made. Experimental results and comparisons actually demonstrate these good aspects of the proposed RL-FCM where it exhibits three robust characteristics: 1) robust to initializations with free of the fuzziness index, 2) robust to (without) parameter selection, and 3) robust to number of clusters (with unknown number of clusters).

1. Introduction

Clustering is a useful tool for data analysis. It is a method for finding groups within data with the most similarity in the same cluster and the most dissimilarity between different clusters. The hierarchical clustering was supposed as the earliest clustering method used by biologist and social scientists. Afterwards, cluster analysis becomes a branch in statistical multivariate analysis [1]. It is also an approach to unsupervised learning as one of the major techniques in pattern recognition and machine learning. According to the statistical point of view, clustering methods may be divided as a probability model-based approach and a nonparametric approach. A probability model-based approach assumes that the data set follows a mixture model of probability distributions so that a mixture likelihood approach to clustering is used [2], where the expectation and maximization (EM) algorithm [3] is the most popular. For a nonparametric approach, clustering methods may be based on an objective function of similarity or dissimilarity measures, where partitional methods are the most used. Graph clustering had also been developed and discussed in the literature [4], for example, clustering based on similarity of user-behavior for targeted advertising was investigated in Aggarwal et al. [5].

Partitional clustering methods suppose that the data set can be represented by finite cluster prototypes with their own objective functions. Therefore, to define the dissimilarity (or distance) between data points and cluster prototypes is essential for partitional methods. It is known that the k-means (or called hard c-means) algorithm is the oldest and popular partitional method [6]. For an efficient estimation for a number of clusters, Pelleg and Moore [7] extended k-means, called X-means, by making local decisions for cluster centers in each iteration of k-means with splitting themselves to get better clustering. Users only need to specify a range of cluster numbers in which the true cluster number reasonably lies and then a model selection, such as Bayesian information criterion (BIC) or Akaike information criterion (AIC), is used to do the splitting process. Although the k-means and X-means algorithms are widely used, these crisp clustering methods restricts that each data point belongs to exactly one cluster with crisp cluster memberships so that it can be well fitted for sharp boundaries between clusters in data, but not good for unsharp (or vague) boundaries.

[*] Corresponding author.
E-mail address: msyang@math.cycu.edu.tw (M.-S. Yang).

http://dx.doi.org/10.1016/j.patcog.2017.05.017
0031-3203/© 2017 Elsevier Ltd. All rights reserved.

Since Zadeh [8] proposed fuzzy set that introduced the idea of partial memberships described by membership functions, it was successfully applied in clustering. Fuzzy clustering has been widely studied and applied in a variety of substantive areas more than 45 years [9–12] since Ruspini [13] first proposed fuzzy *c*-partitions as a fuzzy approach to clustering in the 1970s. In fuzzy clustering, the fuzzy c-means (FCM) clustering algorithm proposed by Dunn [14] and Bezdek [9] is the most well-known and used method. There are many extensions and variants of FCM proposed in the literature. The first important extension to FCM was proposed by Gustafson and Kessel (GK) [15] in which the Euclidean distance in the FCM objective function was replaced by the Mahalanobis distance. Afterwards, there are many extensions to FCM, such as extensions to maximum-entropy clustering (MEC) by Karayiannis [16], Miyamoto and Umayahara [17] and Wei and Fahn [18], extensions to L_p norms by Hathaway et al. [19], extension of FCM as alpha-cut implemented fuzzy clustering algorithms by Yang et al. [20], extension of FCM for treating very large data by Havens et al. [21], an augmented FCM for clustering spatiotemporal data by Izakian et al. [22], and so forth. However, these fuzzy clustering algorithms always need to give a number of clusters a priori. In general, the cluster number *c* is unknown. In this case, validity indices can be used to find a cluster number *c* where they are supposed to be independent of clustering algorithms. Many cluster validity indices for fuzzy clustering algorithms had been proposed in the literature, such as partition coefficient (PC) [23], partition entropy (PE) [24], normalization of PC and PE [25–26], fuzzy hypervolume (FHV) [27] and XB (Xie and Beni [28]).

Frigui and Krishnapuram [29] proposed the robust competitive agglomerative (RCA) algorithm by adding a loss function of clusters and a weight function of data points to clusters. The RCA algorithm can be used for determining a cluster number. Starting with a large cluster number, RCA reduces the number by discarding clusters with less cardinality. Some parameter initial values are needed in RCA, such as time constant, discarding threshold, tuning factor, etc. Another clustering algorithm was presented by Rodriguez and Laio [30] for clustering by fast search, called C-FS, using a similarity matrix for finding density peaks. They proposed the C-FS algorithm by assigning a cutoff distance $d_c$and selecting a decision window so that it can automatically determine a number of clusters. In [30], the cutoff distance d_c becomes another parameter in which clustering results are heavily dependent of the cutoff parameter d_c. Recently, Fazendeiro and Oliveira [31] presented a fuzzy clustering algorithm with an unknown number of clusters based on observer position, called focal point. With this point, observer can select a suitable point while searching for clusters that is actually appropriate to the underlying data structure. After the focal point is chosen, the initialization of cluster centers must be generated randomly. The inverse of XB index is used to compute the validity measure. The maximal value is chosen to get the best number of clusters. Although these algorithms can find a number of clusters during iteration procedures, they are still dependent of initializations and parameter selections.

Up to now, there is no work in the literature for FCM to be simultaneously robust to initializations and parameter selection under free of the fuzziness index without a given number of clusters. We think that this may be due to its difficulty for constructing this kind of robust FCM. In this paper, we try to construct a robust learning-based framework for fuzzy clustering, especially for the FCM algorithm. This framework can automatically find the best number of clusters, without any initialization and parameter selection, and it is also free of the fuzziness index *m*. We first consider some entropy-type penalty terms for adjusting the bias, and then create a robust-learning mechanism for finding the best number of clusters. The organization of this paper is as follows. In Section 2, we construct a robust learning-based framework for fuzzy clustering. The robust-learning FCM (RL-FCM) clustering algorithm is also presented in this section. In Section 3, several experimental examples and comparisons with numeric and real data sets are provided to demonstrate the effectiveness of the proposed RL-FCM, which can automatically find the best number of clusters. Finally, conclusions are stated in Section 4.

2. Robust-learning fuzzy c-means clustering algorithm

Let $\mathbf{X} = \{x_1, \ldots, x_n\}$ be a data set in a *d*-dimensional Euclidean space $\mathbb{R}^d$ and $\mathbf{V} = \{v_1, \ldots, v_c\}$ be the *c* cluster centers with its Euclidean norm denoted by $d_{ik} = \|x_i - v_k\|_2 = \sqrt{\sum_{j=1}^{d} (x_{ij} - v_{kj})^2}$. The fuzzy c-means (FCM) objective function [9–10] is given with $J_m(\mathbf{U}, \mathbf{V}) = \sum_{k=1}^{c} \sum_{i=1}^{n} \mu_{ik}^m d_{ik}^2$ where $m > 1$ is the fuzziness index, $\mu = \{\mu_{ik}\}_{n\times c} \in M_{fcn}$ is a fuzzy partition matrix with $M_{fcn} = \{\mu = [\mu_{ik}]_{nc} | \forall i, \forall k, 0 \le \mu_{ik} \le 1, \sum_{k=1}^{c} \mu_{ik} = 1, 0 < \sum_{i=1}^{n} \mu_{ik} < n\}$, and $d_{ik} = \|x_i - v_k\|_2$ is the Euclidean distance. The FCM algorithm is iterated through necessary conditions for minimizing $J_m(\mathbf{U}, \mathbf{V})$with the updating equations for cluster centers and memberships as: $v_k = \sum_{i=1}^{n} \mu_{ik}^m x_i / \sum_{i=1}^{n} \mu_{ik}^m$ and $\mu_{ik} = (d_{ik})^{-2/(m-1)} / \sum_{t=1}^{c} (d_{it})^{-2/(m-1)}$.

We know that the FCM algorithm is dependent on initial values and some parameters need to be given a priori, such as a fuzziness index *m*, cluster center initialization and also a number of clusters. Although there exist some works in the literature to solve some problems in FCM, such as Dembélé and Kastner [32] and Schwämmle and Jensen [33] on estimating the fuzziness index *m* for clustering microarray data, there is no work for FCM to be simultaneously robust to initializations and parameter selection under free of the fuzziness index *m* without a given number of clusters. Next, we construct a robust learning-based schema for FCM to simultaneously solve these problems. Our basic idea is that, we first consider all data points as initial cluster centers, i.e., the number of data points is the initial number of clusters. After that, use the mixing proportion α_k of the cluster *k*, which is like a cluster weight, and discard these clusters that have values of α_k less than one over the number of data points. The proposed algorithm can iteratively obtain the best number of clusters until it converges.

For a data set $\mathbf{X} = \{x_1, \ldots, x_n\}$ in $\mathbb{R}^d$ with *c* cluster centers to have FCM be simultaneously robust to initializations and parameter selection under free of the fuzziness index *m* that can automatically find the best number of clusters, we add several entropy terms in the FCM objective function. First, to construct an algorithm free of the fuzziness index *m*, we replace *m* by adding an extra term with a function of μ_{ik}. In this sense, we consider the concept of MEC [16–18] by adding the entropy term of memberships with $\sum_{k=1}^{c} \sum_{i=1}^{n} \mu_{ik} \ln \mu_{ik}$. Moreover, we use a learning function *r*, i.e. $r \sum_{k=1}^{c} \sum_{i=1}^{n} \mu_{ik} \ln \mu_{ik}$, to learn the effects of the entropy term for adjusting bias. We next use the mixing proportion $\alpha = (\alpha_1, \cdots, \alpha_c)$ of clusters, where α_k presents the probability of one data point belonged to the *k*th cluster with the constraint $\sum_{k=1}^{c} \alpha_k = 1$. Hence, $-\ln \alpha_k$ is the information in the occurrence of a data point belonged to the *k*th cluster. Thus, we add the entropy term $\sum_{k=1}^{c} \sum_{i=1}^{n} \mu_{ik} \ln \alpha_k$ to summarize the average of information for the occurrence of a data point belonged to the corresponding cluster over fuzzy memberships. Furthermore, we borrow the idea of Yang et al. [34] in the EM algorithm by using the entropy term, $\sum_{k=1}^{c} \alpha_k \ln \alpha_k$, to represent the average of information for the occurrence of each data point belonged to the corresponding cluster. Totally, the entropy terms of mixing proportion in probability and the average of occurrence in probability over fuzzy memberships are used for learning to find the best number of clusters.

According to the above construction for FCM, we propose a robust-learning FCM (RL-FCM) objective function as follows:

$$J(\mathbf{U},\alpha,\mathbf{V}) = \sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}d_{ik}^2 - r_1\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\ln\alpha_k + r_2\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\ln\mu_{ik} - r_3 n\sum_{k=1}^{c}\alpha_k\ln\alpha_k, \tag{1}$$

where $r_1,\, r_2,\, r_3 \geq 0$ and $d_{ik} = \|x_i - v_k\|_2 = \sqrt{\sum_{j=1}^{d}(x_{ij}-v_{kj})^2}$. The Lagrangian function of (1) is

$$\tilde{J}(\mathbf{U},\alpha,\lambda_1,\lambda_2,\mathbf{V}) = \sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}d_{ik}^2 - r_1\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\ln\alpha_k + r_2\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\ln\mu_{ik} - r_3 n\sum_{k=1}^{c}\alpha_k\ln\alpha_k - \lambda_1\left(\sum_{k=1}^{c}\mu_{ik}-1\right) - \lambda_2\left(\sum_{k=1}^{c}\alpha_k - 1\right). \tag{2}$$

By considering the Lagrangian function in (2), the updating equation for membership function, cluster center, and mixing proportion are as follows. The updating equation for the RL-FCM objective function$J(\mathbf{U},\,\alpha,\,\mathbf{V})$ with respective to v_k is as Eq. (3).

$$v_k = \sum_{i=1}^{n}\mu_{ik}x_i \Bigg/ \sum_{i=1}^{n}\mu_{ik}. \tag{3}$$

By taking the partial derivative of the Lagrangian in Eq. (2) with respect to μ_{ik} and setting them to be zero, it becomes $\frac{\partial\tilde{J}}{\partial\mu_{ik}} = d_{ik}^2 - r_1\ln\alpha_k + r_2(\ln\mu_{ik}+1) - \lambda_1 = 0$ and then $\ln\mu_{ik} = \frac{(-d_{ik}^2 + r_1\ln\alpha_k + \lambda_1 - r_2)}{r_2}$. Thus, the updating equation for μ_{ik} is obtained as follows:

$$\mu_{ik} = \exp\left(\frac{-d_{ik}^2 + r_1\ln\alpha_k}{r_2}\right) \Bigg/ \sum_{t=1}^{c}\exp\left(\frac{-d_{it}^2 + r_1\ln\alpha_t}{r_2}\right). \tag{4}$$

Similarly, we have $\frac{\partial\tilde{J}}{\partial\alpha_k} = -r_1\sum_{i=1}^{n}\frac{\mu_{ik}}{\alpha_k} - r_3 n(\ln\alpha_k + 1) - \lambda_2 = 0$. By multiplying with α_k, we obtain

$$-r_1\sum_{i=1}^{n}\mu_{ik} - r_3 n\alpha_k(\ln\alpha_k + 1) - \lambda_2\alpha_k = 0 \tag{5}$$

and then $-r_1\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik} - \sum_{k=1}^{c} nr_3\alpha_k\ln\alpha_k - \sum_{k=1}^{c} nr_3\alpha_k - \sum_{k=1}^{c}\lambda_2\alpha_k = 0$. We get

$$\lambda_2 = -nr_1 - nr_3\sum_{k=1}^{c}\alpha_k\ln\alpha_k - nr_3. \tag{6}$$

By substituting (6) to (5), we have $-r_1\sum_{i=1}^{n}\mu_{ik} - nr_3\alpha_k(\ln\alpha_k + 1) - (-nr_1 - nr_3\sum_{k=1}^{c}\alpha_k\ln\alpha_k - nr_3)\alpha_k = 0$.

Thus, the updating equation for α_k can be obtain as follows:

$$\alpha_k^{(new)} = \frac{1}{n}\sum_{i=1}^{n}\mu_{ik} + \frac{r_3}{r_1}\alpha_k^{(old)}\left(\ln\alpha_k^{(old)} - \sum_{t=1}^{c}\alpha_t^{(old)}\ln\alpha_t^{(old)}\right). \tag{7}$$

For solving the initialization problem, all data points is assigned as initial clusters for the first iteration. That is, $c^{(0)} = n$ and $\alpha_k^{(0)} = 1/c = 1/n,\ k = 1,\cdots,c$. There are competitions between these mixing proportions according to Eq. (7). Iteratively, the algorithm can find the final number of clusters c by utilizing the following Eq. (8). When $\alpha_k^{(new)} < 1/n$, we discard illegitimate mixing proportion $\alpha_k^{(new)}$. Therefore, the updating number of clusters $c^{(new)}$ is

$$c^{(new)} = c^{(old)} - \left|\left\{\alpha_k{}^{(new)}\,\middle|\,\alpha_k{}^{(new)} < 1/n,\ k = 1,2,..,c^{(old)}\right\}\right| \tag{8}$$

where $|\{\}|$ denotes the cardinality of the set $\{\}$. After updating the number of clusters c, the remaining mixing proportion α_k^* and corresponding μ_{ik}^* need to be re-normalized by,

$$\alpha_k^* = \alpha_k^* \Bigg/ \sum_{t=1}^{c^{(new)}}\alpha_t^* \tag{9}$$

$$\mu_{ik}^* = \mu_{ik}^* \Bigg/ \sum_{t=1}^{c^{(new)}}\mu_{it}^* \tag{10}$$

Eqs. (9) and (10) keep the constraints $\sum_{k=1}^{c^{(new)}}\alpha_k^* = 1$ and $\sum_{k=1}^{c^{(new)}}\mu_{ik}^* = 1$. We utilize this concept to estimate the best number of clusters c^*.

A new problem is how to learn the values of the three parameters, r_1, r_2, and r_3 for the three penalty terms $\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\ln\alpha_k$, $\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\ln\mu_{ik}$, and $\sum_{k=1}^{c}\alpha_k\ln\alpha_k$, respectively. By considering some decrease learning rates, such as e^{-t}, $e^{-t/10}$, $e^{-t/100}$, and $e^{-t/1000}$, we know that $y = e^{-t/1000}$ decreases slower, but $y = e^{-t}$ decreases faster. Since $\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\ln\alpha_k$ has effect on membership partition and mixing proportion, we assume that r_1 is not set to decrease too slow or too fast. Therefore, we set r_1 as

$$r_1^{(t)} = e^{-t/10}. \tag{11}$$

On the other hand, because the term $\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\ln\mu_{ik}$ is the entropy to the partition membership μ_{ik} and has effect on the clustering results, the parameter r_2 should maintain large value and does not need too much variation in iterative process. In this sense, we consider the decreasing learning rate for r_2 by assigning it with

$$r_2^{(t)} = e^{-t/100}. \tag{12}$$

In order to avoid that $\sum_{k=1}^{c}\alpha_k\ln\alpha_k$ interferes with $\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ik}\ln\mu_{ik}$ when the algorithm is stable, the term $\sum_{k=1}^{c}\alpha_k\ln\alpha_k$ needs large effect in initially iterative process and small effect when the algorithm is stable. Since r_3 is a control scale for the entropy to α_k, we consider that r_3 is related to the variation of the mixing proportion $|\alpha_k^{(new)} - \alpha_k^{(old)}|$. Our goal is that r_3 can control competition of the mixing proportions. Therefore, first r_3 is defined with

$$r_3 = \frac{\sum_{k=1}^{c}\exp\left(-\eta n\left|\alpha_k^{(new)} - \alpha_k^{(old)}\right|\right)}{c}, \tag{13}$$

where $\eta = \min\{1, 2/d^{\lfloor d/2-1\rfloor}\}$ and the notation $\lfloor a\rfloor$ denotes the largest integer no more than a. In Eq. (13), if $|\alpha_k^{(new)} - \alpha_k^{(old)}|$ is small, then r_3 will become large to enhance its competition. If $|\alpha_k^{(new)} - \alpha_k^{(old)}|$ is large, then r_3 is small to maintain stability. In addition, the competition of the mixing proportions for the higher dimensional data needs the larger value of r_3. Therefore, $\eta = \min\{1, 2/d^{\lfloor d/2-1\rfloor}\}$ is to adjust r_3. Furthermore, we need to consider the restriction of $\max\limits_{1\leq k\leq c}\alpha_k^{(new)} \leq 1$. However, $\max\limits_{1\leq k\leq c}\alpha_k^{(new)} \leq \max\limits_{1\leq k\leq c}(\frac{1}{n}\sum_{i=1}^{n}\mu_{ik}) + \frac{r_3}{r_1}\max\limits_{1\leq k\leq c}\alpha_k^{(old)}(\ln\max\limits_{1\leq k\leq c}\alpha_k^{(old)} - \sum_{t=1}^{c}\alpha_t^{(old)}\ln\alpha_t^{(old)})$ and $\max\limits_{1\leq k\leq c}(\frac{1}{n}\sum_{i=1}^{n}\mu_{ik}) + \frac{r_3}{r_1}\max\limits_{1\leq k\leq c}\alpha_k^{(old)}(\ln\max\limits_{1\leq i\leq c}\alpha_i^{(old)} - \sum_{t=1}^{c}\alpha_t^{(old)}\ln\alpha_t^{(old)}) < \max\limits_{1\leq k\leq c}(\frac{1}{n}\sum_{i=1}^{n}\mu_{ik}) + r_3(-(\max\limits_{1\leq k\leq c}\alpha_k^{(old)}\sum_{t=1}^{c}\alpha_t^{(old)}\ln\alpha_t^{(old)}))$.

Therefore, if $\max\limits_{1\leq k\leq c}(\frac{1}{n}\sum_{i=1}^{n}\mu_{ik}) - r_3\max\limits_{1\leq k\leq c}\alpha_k^{(old)}\sum_{t=1}^{c}\alpha_t^{(old)}\ln\alpha_t^{(old)} \leq 1$, then the restriction will be held. It follows that

$$r_3 \le \left(1 - \max_{1\le k\le c}\left(\frac{1}{n}\sum_{i=1}^{n}\mu_{ik}\right)\right) \Bigg/ \left(-\max_{1\le k\le c}\alpha_k^{(old)}\sum_{t=1}^{c}\alpha_t^{(old)}\ln\alpha_t^{(old)}\right). \tag{14}$$

To combine Eq. (13) and Eq. (14), we obtain

$$r_3 = \min\left(\frac{\sum_{k=1}^{c}\exp\left(-\eta n\left|\alpha_k^{(new)}-\alpha_k^{(old)}\right|\right)}{c}, \frac{1-\max\limits_{1\le k\le c}\left(\frac{1}{n}\sum_{i=1}^{n}\mu_{ik}\right)}{\left(-\max\limits_{1\le k\le c}\alpha_k^{(old)}\sum_{t=1}^{c}\alpha_t^{(old)}\ln\alpha_t^{(old)}\right)}\right). \tag{15}$$

When the number of clusters c is stable, the competition of mixing proportions will be stopped. The parameter r_3 can be set as 0 at this point. In our experiments, the number of clusters c is usually stable if iteration t large than or equal to 100. Thus, we give the flowchart of the proposed robust-learning FCM (RL-FCM) as shown in Fig. 1, and the RL-FCM clustering algorithm is summarized as follows:

RL-FCM Algorithm

Fix $\varepsilon > 0$. Give initials $c^{(0)} = n$, $v_k^{(0)} = x_i$, $\alpha_k^{(0)} = \frac{1}{n}$, and initial learning rates $r_1^{(0)} = r_2^{(0)} = r_3^{(0)} = 1$. Let $t = 1$.

Step 1: Compute $\mu_{ik}^{(t)}$ using $v_k^{(t-1)}$, $\alpha_k^{(t-1)}$, $c^{(t-1)}$, $r_1^{(t-1)}$, $r_2^{(t-1)}$ by Eq. (4).

Step 2: Update $r_1^{(t)}$ and $r_2^{(t)}$ by Eqs. (11) and (12), respectively.

Step 3: Update $\alpha_k^{(t)}$ with $\mu_{ik}^{(t)}$ and $\alpha_k^{(t-1)}$ by Eq. (7).

Step 4: Update $r_3^{(t)}$ with $\alpha_k^{(t)}$ and $\alpha_k^{(t-1)}$ by Eq. (15).

Step 5: Update $c^{(t-1)}$ to $c^{(t)}$ by discarding those clusters with $\alpha_k^{(t)} \le 1/n$ using Eq. (8) and normalize $\alpha_k^{(t)}$ and $\mu_{ik}^{(t)}$ by Eqs. (9) and (10), respectively.
IF $t \ge 100$ and $c^{(t-100)} - c^{(t)} = 0$, THEN let $r_3^{(t)} = 0$.

Step 6: Update $v_k^{(t)}$ using $c^{(t)}$ and $\mu_{ik}^{(t)}$ by Eq. (3).

Step 7: Compare $v_k^{(t)}$ and $v_k^{(t-1)}$.
IF $\max\limits_{1\le k\le c^{(t-1)}} \|v_k^{(t)} - v_k^{(t-1)}\| < \varepsilon$, STOP.
ELSE $t = t + 1$ and return to Step 1.

We analyze the computational complexity for the RL-FCM algorithm. The computational complexity is calculated as follows. The RL-FCM algorithm can be divided into three parts: (1) compute the membership partition,μ_{ik}, which needsO(nc^2d); (2) compute the mixing proportion α_k, which needs O(nc^2); (3) update cluster center, v_k, which needs O(nc). Because the notation of big O (i.e.,O(·)) only considers the upper bound on the growth rate of the function, the total computational complexity for the RL-FCM algorithm is O(nc^2d), where n is the number of data, c is the number of clusters, and d is the dimension of data points. In fact, the RL-FCM has the same computational complexity as the FCM, i.e.O(nc^2d). The most difference between the RL-FCM and FCM is the initial number c of clusters, where c is given a priori in the FCM, but c decreases from n to c_{final}, $n >> c_{final}$, in the RL-FCM. We should mention that, even though the RL-FCM algorithm uses the number n of data points as the number c of clusters (i.e., $c = n$) in the beginning of iterations, the time per iteration will decrease rapidly after several iterations. This situation occurs because the clusters with $\alpha_k \le 1/c = 1/n$ will be discarded during iterations, so that the number c of clusters will decrease rapidly after some iterations. We will demonstrate this behavior of the RL-FCM in the next section.

In the next Example 1, we demonstrate these robust learning behaviors to get the best number c^* of clusters for the RL-FCM. For measuring clustering performance, we use an accuracy rate (AR) with AR $= \sum_{k=1}^{c} n(c_k)/n$, where $n(c_k)$ is the number of data points that obtained correct clustering for the cluster k, and n is the total number of data points. The larger AR is, the better clustering performance is.

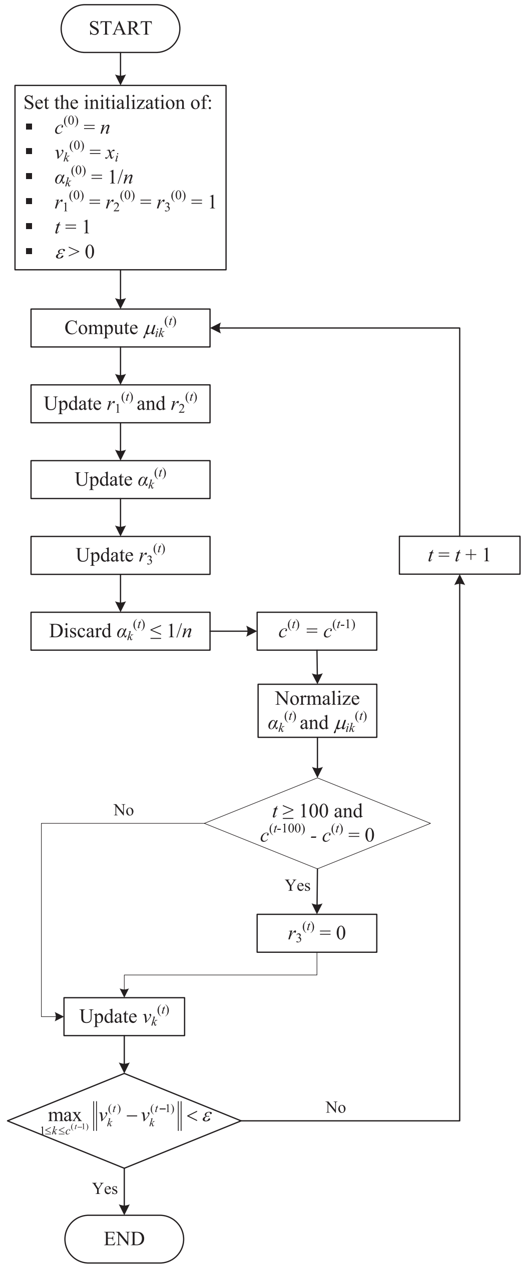
START
Set the initialization of:
▪ $c^{(0)} = n$
▪ $v_k^{(0)} = x_i$
▪ $\alpha_k^{(0)} = 1/n$
▪ $r_1^{(0)} = r_2^{(0)} = r_3^{(0)} = 1$
▪ $t = 1$
▪ $\varepsilon > 0$
Compute $\mu_{ik}^{(t)}$
Update $r_1^{(t)}$ and $r_2^{(t)}$
Update $\alpha_k^{(t)}$
Update $r_3^{(t)}$
Discard $\alpha_k^{(t)} \le 1/n$
$c^{(t)} = c^{(t-1)}$
Normalize $\alpha_k^{(t)}$ and $\mu_{ik}^{(t)}$
$t \ge 100$ and $c^{(t-100)} - c^{(t)} = 0$
No
Yes
$r_3^{(t)} = 0$
Update $v_k^{(t)}$
$\max_{1\le k\le c^{(t-1)}} \|v_k^{(t)} - v_k^{(t-1)}\| < \varepsilon$
No
Yes
$t = t + 1$
END

Fig. 1. Flowchart of RL-FCM.

For Gaussian mixture model in Example 1, data is generated from the d-variate normal mixture model $f(x;\alpha,\theta) = \sum_{k=1}^{c}\alpha_k f(x;\theta_k) = \sum_{k=1}^{c}\alpha_k (2\pi)^{-(d/2)} |\Sigma_k|^{-(1/2)} e^{-(1/2)(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k)}$, where $\alpha_k > 0$ denotes mixing proportions with $\sum_{k=1}^{c}\alpha_k = 1$, and $f(x;\ \theta_k)$ denotes the density of x from kth group with the corresponding parameter θ_k consisted of a mean vector μ_k and a covariance matrix Σ_k.

Example 1. In this example, we use a data set with 500 data points generated from a three-component Gaussian mixture distribution, with parameters $\alpha_1 = \alpha_2 = \alpha_3 = 1/3$, $\mu_1 = (0\quad 0)^T$, $\mu_2 = (7\quad 0)^T$, $\mu_3 = (14\quad 0)^T$, and $\Sigma_1 = \Sigma_2 = \Sigma_3 = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$, as shown in Fig. 2(a). After two iterations, the number of cluster decreases

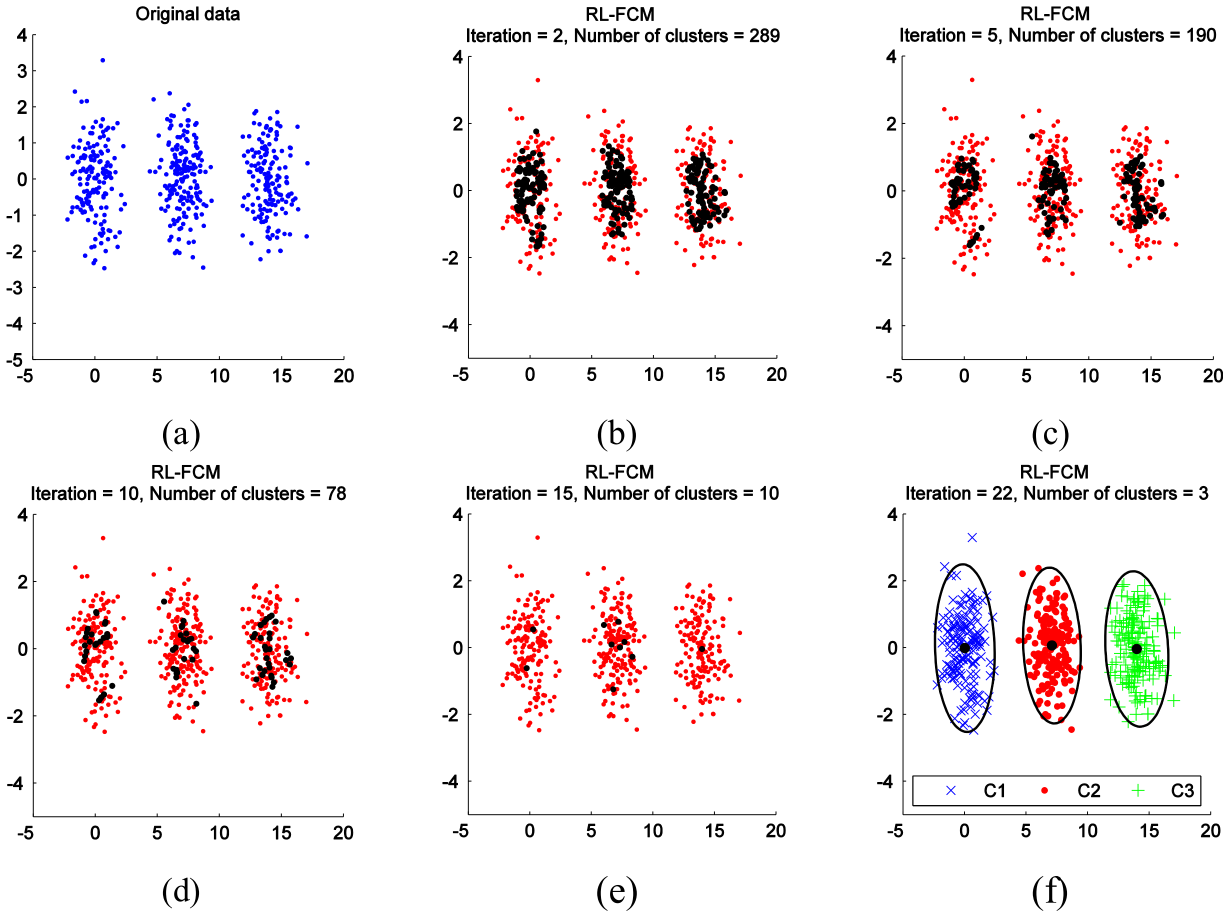

Fig. 2. (a) 3-cluster dataset; (b) Clustering result after 2 iterations with 289 clusters; (c) Clustering result after 5 iterations with 190 clusters; (d) Clustering result after 10 iterations with 78 clusters; (e) Clustering result after 15 iterations with 10 clusters; (f) Final result from RL-FCM with 3 clusters.

rapidly from 500 to 289, as shown in Fig. 2(b). The RL-FCM algorithm decreases the number of cluster to 190, 78, and 10 clusters after 5, 10, and 15 iterations (see Fig. 2(c)–(e)), respectively. Finally, after 22 iterations (see Fig. 2(f)), the RL-FCM algorithm obtains its convergence where three clusters are formed with $c^* = 3$ and AR = 1.00.

3. Experimental results and comparisons

In this section we present some experimental examples with artificial and real data sets, also image segmentations to show the performance of the proposed RL-FCM algorithm. The validity indices of partition coefficient (PC) [23], partition entropy (PE) [24], modified PC (MPC) [25], modified PE (MPE) [26], fuzzy hypervolume (FHV) [27], and Xie and Beni (XB) [28] are computed to compare with the number of clusters obtained from RL-FCM. The comparisons between RL-FCM, FCM, robust competitive agglomeration (RCA) [29], clustering by fast search (C-FS) [30], observer-biased (OB) [31], and robust EM (R-EM) [34] are also made, by using fuzziness index $m = 2$.

Example 2. In this example, we use a data set with 500 data points generated from the five-component Gaussian mixture distribution, with parameters $\alpha_k = 1/5, k = 1, \ldots, 5$, $\mu_1 = (0 \quad 0)^T$, $\mu_2 = (0 \quad 6)^T$, $\mu_3 = (10 \quad 0)^T$, $\mu_4 = (10 \quad 6)^T$, $\mu_5 = (5 \quad 3)^T$, $\Sigma_1 = \Sigma_2 = \Sigma_3 = \Sigma_4 = \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$, and $\Sigma_5 = \begin{pmatrix}0.01 & 0\\ 0 & 10\end{pmatrix}$, as shown in Fig. 3(a). The data set is seen as two circles at the top side, two circles at the bottom side, and a separated line horizontal shape in the center. The proposed RL-FCM algorithm is applied for this data set and have the number of clusters decrease from 500 to 5 after 26 iterations, as shown in Fig. 3(b). We find that the RL-FCM can detect five clusters with $c^* = 5$ and AR = 0.9960. While using the FCM by assigning the number of clusters, $c = 5$, yields the clustering results as shown in Fig. 3(c) with the same AR.

Table 1
Validity index values of PC, PE, MPC, MPE, FHV and XB for the data set in Fig. 3(a).

c	PC	PE	MPC	MPE	FHV	XB
2	**0.7497**	**0.4013**	0.4994	0.4210	13.3712	0.1482
3	0.6414	0.6335	0.4621	0.4234	13.3343	0.2399
4	0.7219	0.5572	0.6292	0.5980	7.6920	0.0990
5	0.7643	0.5202	**0.7053**	**0.6768**	**5.2695**	**0.0553**
6	0.7280	0.6134	0.6736	0.6577	5.6192	0.1313

As known, cluster validity indices are usually used to find the number of clusters. In this example, PC [23], PE [24], MPC [25], MPE [26], FHV [27], and XB [28] are computed to check whether RL-FCM is valid or not. All validity index values are shown in Table 1. The indices PC and PE give the best number of clusters with $c^* = 2$, but the indices MPC, MPE, FHV, and XB give the best number of clusters with $c^* = 5$.

Furthermore, to demonstrate the performance of RL-FCM for noisy data, we add 50 uniformly noisy points (i.e. background outliers) in the data set of Fig. 3(a). The noisy data set is shown in

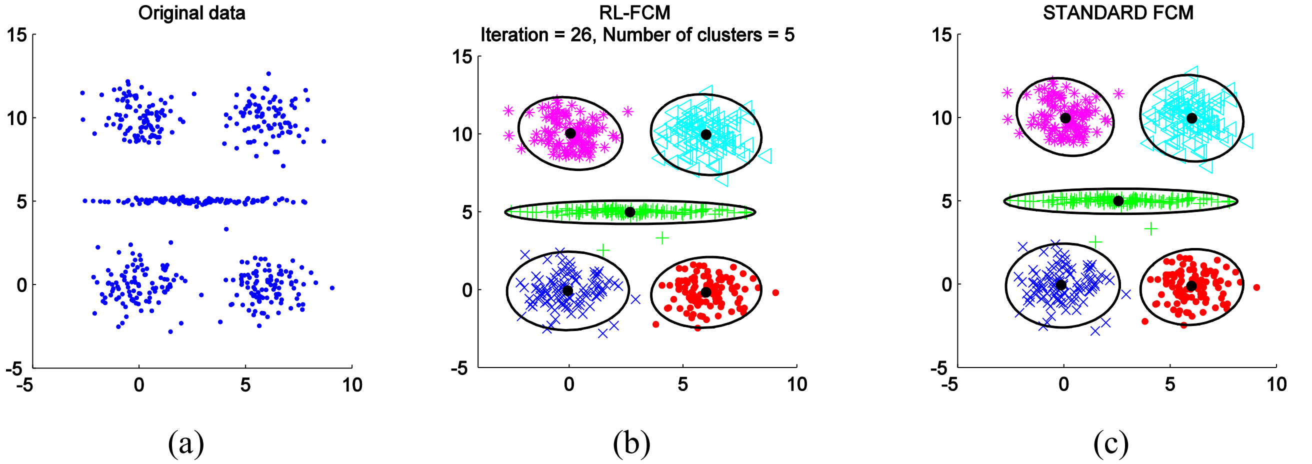

Fig. 3. (a) 5-cluster dataset; (b) Final result from RL-FCM; (c) Final result from FCM with $c = 5$.

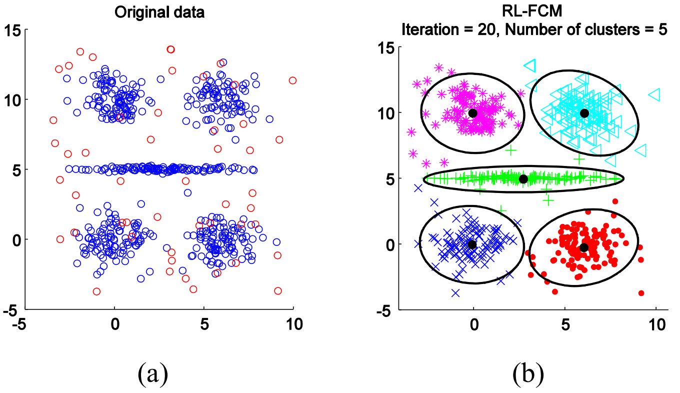

Fig. 4. (a) 5-cluster dataset with noisy points; (b) Final result from RL-FCM.

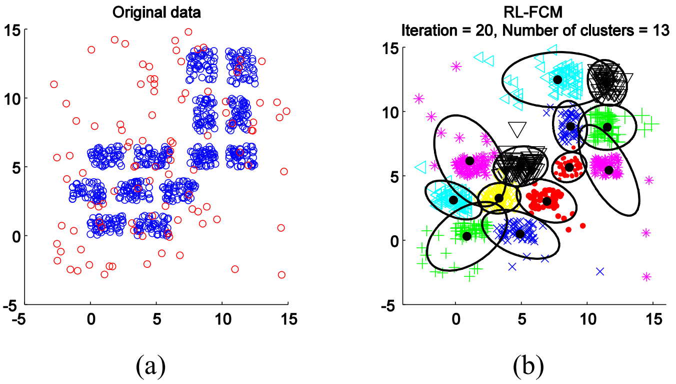

Fig. 6. (a) 13-cluster dataset with 100 noisy points; (b) Final results from RL-FCM.

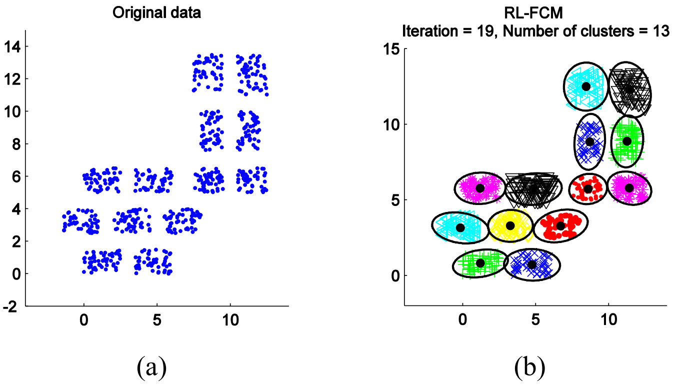

Fig. 5. (a) 13-cluster data set; (b) Final result from RL-FCM.

Table 2
Validity index values of PC, PE, MPC, MPE, FHV and XB for the data set in Fig. 5(a).

c	PC	PE	MPC	MPE	FHV	XB
2	**0.8173**	**0.3109**	**0.6347**	0.5514	9.4148	0.0985
3	0.6920	0.5526	0.5380	0.4970	9.9570	0.1395
4	0.6201	0.7160	0.4935	0.4835	10.0563	0.1968
5	0.5714	0.8538	0.4642	0.4695	10.2332	0.1750
6	0.5571	0.9122	0.4685	0.4909	9.6476	0.1611
7	0.5478	0.9715	0.4725	0.5008	9.0646	0.1420
8	0.5553	0.9845	0.4917	0.5266	8.3539	0.1419
9	0.5530	1.0206	0.4971	0.5355	7.9127	0.1121
10	0.5773	0.9970	0.5303	0.5670	6.5653	0.1184
11	0.5866	0.9987	0.5453	0.5835	5.9894	0.0910
12	0.6072	0.9707	0.5715	0.6094	5.1708	0.0826
13	0.6201	0.9540	0.5885	**0.6281**	**4.6007**	**0.0683**
14	0.6055	1.0003	0.5751	0.6210	4.7423	0.2000

Fig. 4(a). We implement the RL-FCM algorithm for the noisy data set with clustering results shown in Fig. 4(b). We find that the RL-FCM algorithm still obtains five clusters after 20 iterations with AR = 0.9940. In this case, the RL-FCM algorithm is quite robust in this noisy environment.

Example 3. In this example, we show another data set with 13 blocks generated from continuously uniform distribution, where each block contains 50 data points, as shown in Fig. 5(a). Using the RL-FCM algorithm, we get 13 clusters after 19 iterations. We find that the RL-FCM can detect 13 clusters with $c^* = 13$ and AR = 1.00, as shown in Fig. 5(b). The validity index values of PC, PE, MPC, MPE, FHV and XB are shown in Table 2, where the validity indices of PC, PE and MPC give the best number of clusters with $c^* = 2$, the validity indices of MPE, FHV, and XB give the best number of clusters with $c^* = 13$.

We continue adding 100 noisy points in the data set of Fig. 5(a). The noisy data set is shown in Fig. 6(a). The effectiveness of RL-FCM in handling noise is demonstrated in Fig. 6(b) where we can see the RL-FCM algorithm still obtains 13 clusters after

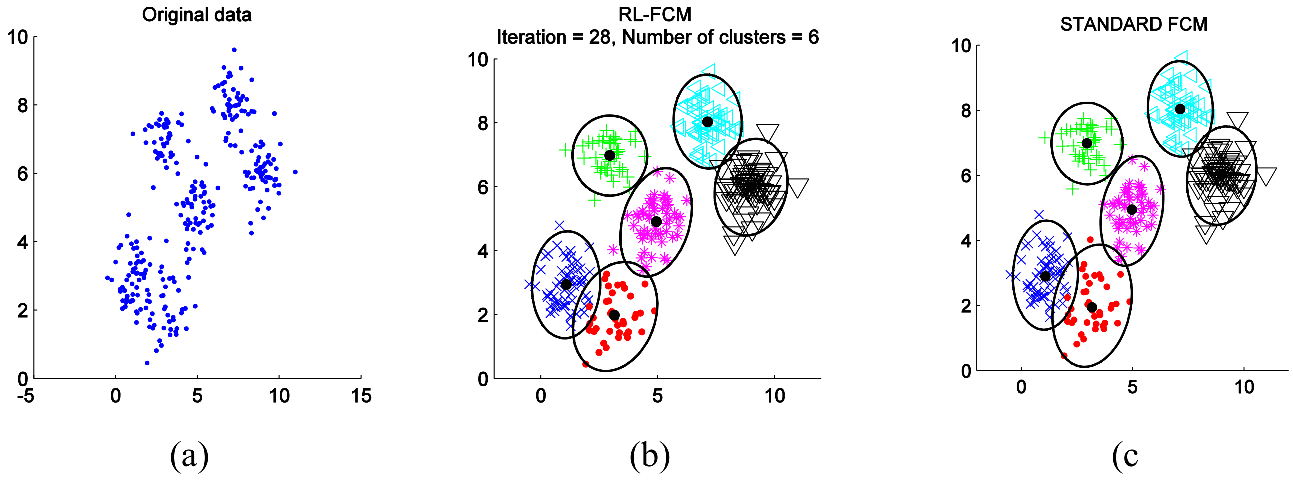

Fig. 7. (a) Original data set; (b) Final result from RL-FCM; (c) Final result from FCM with $c = 6$.

Table 3
Validity index values of PC, PE, MPC, MPE, FHV and XB for the data set in Fig. 7(a).

c	PC	PE	MPC	MPE	FHV	XB
2	**0.7856**	**0.3458**	0.5713	0.5010	4.6115	0.1063
3	0.7293	0.5035	0.5940	0.5417	3.6960	0.1178
4	0.7016	0.5935	0.6021	0.5719	3.2528	0.1744
5	0.6727	0.6809	0.5909	0.5769	3.0853	0.1804
6	0.7142	0.6284	**0.6571**	**0.6493**	**2.2999**	**0.1023**
7	0.6622	0.7420	0.6059	0.6187	2.5522	0.5717

20 iterations with AR = 1.00 from these clustering results shown in Fig. 6(b).

Example 4. Modifying from Fazendeiro and Oliveira [31], 300 data points of a 2D Gaussian mixture distribution are constructed as seen in Fig. 7(a), with parameters $\alpha_k = 1/6$, $\mu_1 = (1 \quad 3)^T$, $\mu_2 = (3 \quad 2)^T$, $\mu_3 = (3 \quad 7)^T$, $\mu_4 = (5 \quad 5)^T$, $\mu_5 = (7 \quad 8)^T$, $\mu_6 = (9 \quad 6)^T$, and $\Sigma_k = \begin{pmatrix} 0.4 & 0 \\ 0 & 0.4 \end{pmatrix}$, $k = 1, \ldots, 6$, where two clusters are overlapping. The proposed RL-FCM algorithm can obtain six clusters as well as the clustering results obtained from Fazendeiro and Oliveira [31] after 28 iterations, as shown in Fig. 7(b), with AR = 0.9667. Using FCM with $c = 6$, the average AR is 0.8910 (Fig. 7(c)). The validity index values of PC, PE, MPC, MPE, FHV and XB are shown in Table 3, where the validity indices MPC, MPE, FHV, and XB give the number of clusters with $c^* = 6$ and the validity indices PC and PE give the number of clusters with $c^* = 2$.

Moreover, to show the performance of RL-FCM with more overlapping data set, we generate 500 data points of a 2D Gaussian mixture distribution as seen in Fig. 8(a), with parameters $\alpha_k = 1/6$, $\mu_1 = (1.5 \quad 3)^T$, $\mu_2 = (3 \quad 3)^T$, $\mu_3 = (3 \quad 6)^T$, $\mu_4 = (5 \quad 5)^T$, $\mu_5 = (6 \quad 7)^T$, $\mu_6 = (8 \quad 6.5)^T$, and $\Sigma_k = \begin{pmatrix} 0.4 & 0 \\ 0 & 0.4 \end{pmatrix}$, $k = 1, \ldots, 6$. The proposed RL-FCM algorithm still can obtain six clusters after 32 iterations, as shown in Fig. 8(b), with AR = 0.9140, while using FCM with $c = 6$, the average AR is 0.8986 (Fig. 8(c)).

Example 5. For considering more number of clusters, in this example, we generate 500 data points from a 2D Gaussian mixture distribution, as shown in Fig. 9(a), where the data set is consisted of 25 clusters with $\Sigma_k = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $k = 1, \ldots, 25$. Using the RL-FCM algorithm, the data set can be well separated into 25 clusters after 31 iterations, as shown in Fig. 9(b), with AR = 1.00. While using FCM with $c = 25$, the average AR is 0.8558, as shown in Fig. 9(c).

Table 4
Cluster number results using different learning functions for r_1 with $r_2 = e^{-t/100}$ (ARs presented inside brackets).

		Ex. 1	Ex. 2	Ex. 3	Ex. 4
	True c	3	5	13	6
r_1	e^{-t}	2	2	4	4
	$e^{-t/10}$	3 (1.0000)	5 (0.9960)	13 (1.0000)	6 (0.9667)
	$e^{-t/100}$	9	12	13 (1.0000)	6 (0.9633)
	$e^{-t/1000}$	9	9	13 (1.0000)	6 (0.9633)

Furthermore, we continue a data set with $\Sigma_k = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$, $k = 1, \ldots, 25$, as shown in Fig. 10(a). We find that RL-FCM still obtain 25 clusters after 36 iterations with AR = 0.9160, as shown in Fig. 10(b). While using FCM with $c = 25$, the average AR is 0.8340, as shown in Fig. 10(c).

Example 6. In this example, we consider 3-dimensional data with 17 blocks generated from uniform distributions, where each block contains 50 data points, as shown in Fig. 11(a). Using the RL-FCM algorithm, it gets 17 clusters after 51 iterations, as shown in Fig. 11(b). RL-FCM can detect 17 clusters with AR = 1.00, while using FCM with $c = 17$, the average AR is 0.8851, as shown in Fig. 11(c).

We continue a more number of clusters that has 21 blocks generated from uniform distributions, as shown in Fig. 12(a). RL-FCM obtains 21 clusters after 49 iterations with AR = 0.9962, as shown in Fig. 12(b). While using FCM with $c = 21$, the average AR = 0.8826, as shown in Fig. 12(c).

We study different learning behaviors for the parameter r_1 and r_2. From all above examples, we implement the RL-FCM algorithm using four learning functions, i.e., e^{-t}, $e^{-t/10}$, $e^{-t/100}$, and $e^{-t/1000}$. Clustering results obtained by using $r_1 = e^{-t}$ indicate that cluster bias is adjusted too fast, so it is very difficult to target optimal cluster centers with correct number of clusters. On the other hand, when $r_1 = e^{-t/1000}$, cluster bias is adjusted slow, so it cannot create better clustering results with correct number of clusters. Tables 4 and 5 show the cluster number results using different learning rates for r_1 and r_2 (ARs are presented inside brackets), respectively. We also can see the behavior of r_1 and r_2 from these tables. Most of them, the greater r_1 obtains the more cluster number. While for r_2, the greater r_2 gives the less cluster number. Using $r_1 = e^{-t/10}$, $r_1 = e^{-t/100}$, $r_1 = e^{-t/1000}$ or $r_2 = e^{-t/10}$, $r_2 = e^{-t/100}$, $r_2 = e^{-t/1000}$, some cluster bias can be adjusted well and targeted to optimal cluster centers. However, $r_1 = e^{-t/10}$ and $r_2 = e^{-t/100}$ can

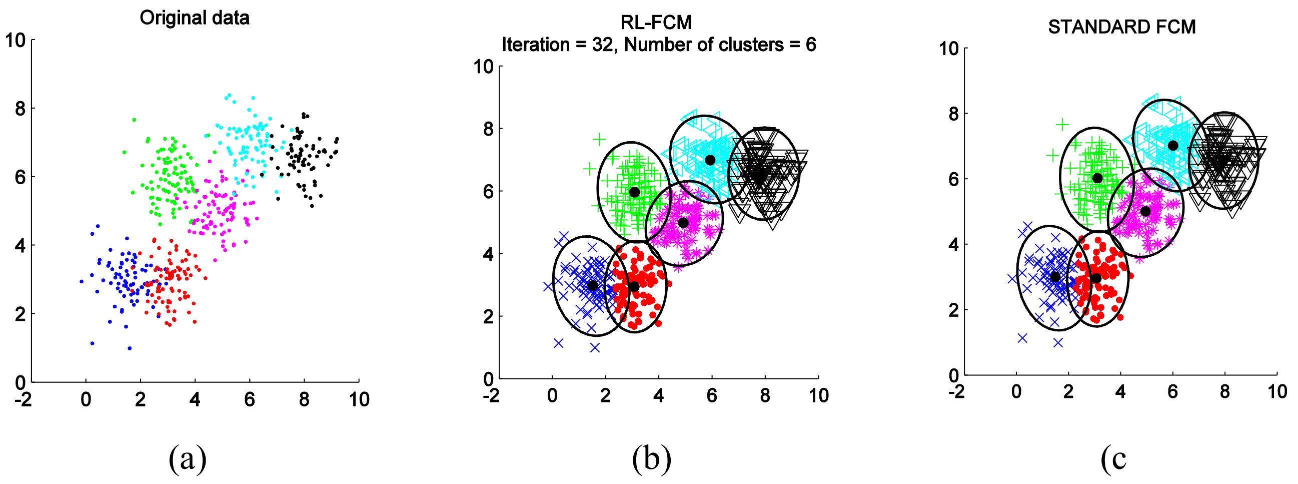

Fig. 8. (a) Original data set; (b) Final result from RL-FCM; (c) Final result from FCM with $c = 6$.

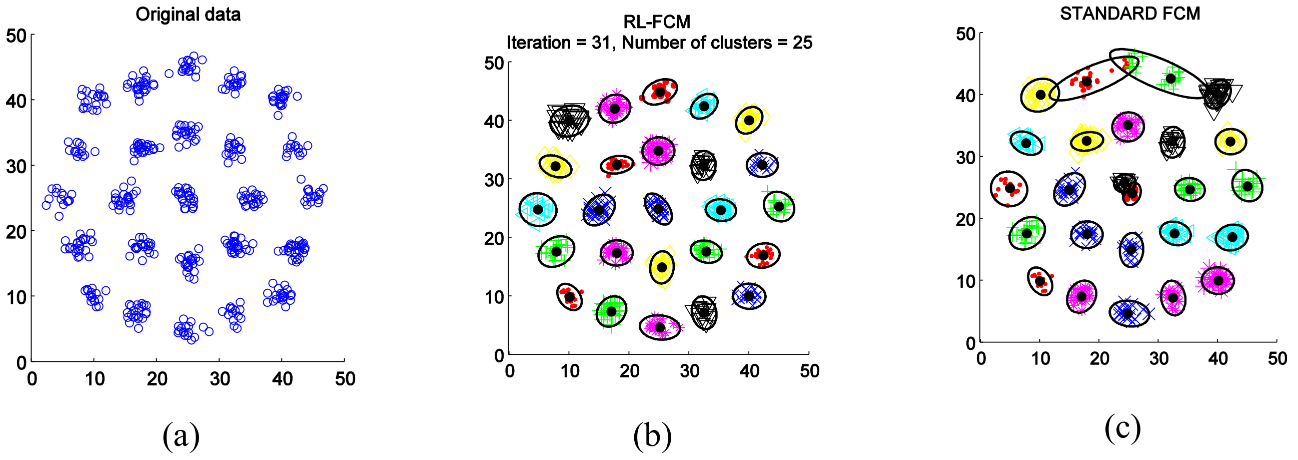

Fig. 9. (a) 25-cluster dataset; (b) Final result from RL-FCM; (c) Final result from FCM with $c = 25$.

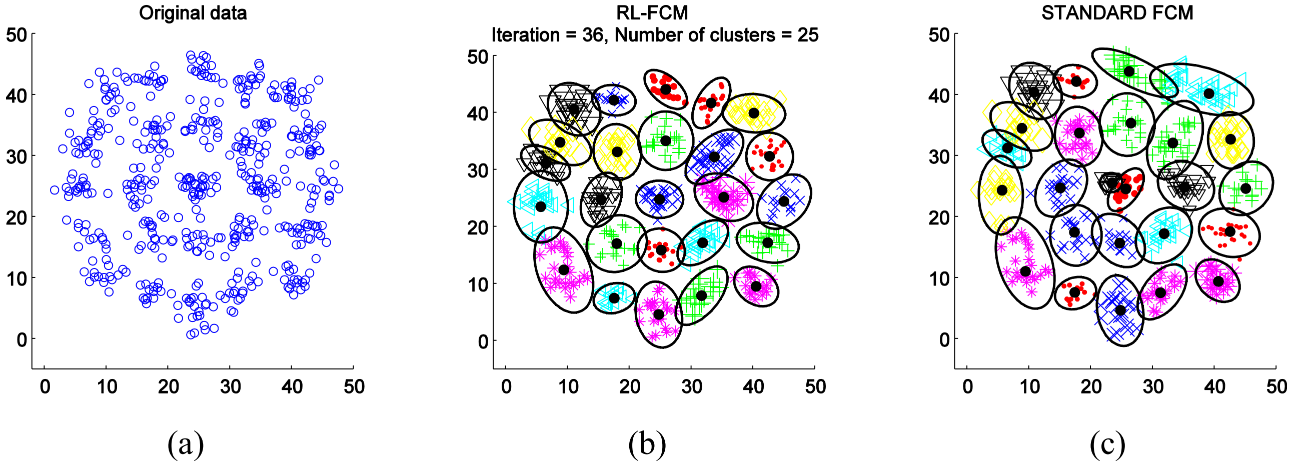

Fig. 10. (a) 25-cluster data set with closer cluster; (b) Final result from RL-FCM; (c) Final result from FCM with $c = 25$.

often give the best number of clusters with the smallest error rate. Overall, we recommend the learning function $r_1 = e^{-t/10}$ and $r_2 = e^{-t/100}$ as a decreasing learning of the parameters r_1 and r_2, respectively, in the proposed RL-FCM algorithm.

Example 7 (Iris data set)**.** In this example, we use the Iris real dataset from UCI Machine Learning Repository [35], which contains 150 data points with four attributes, i.e., sepal length (SL, in cm), sepal width (SW, in cm), petal length (PL, in cm), and petal width (PW, in cm). The Iris data set originally has three clusters (i.e., setosa, versicolor, and virginica). Using the RL-FCM algorithm, we get three clusters in 23 iterations, as shown in Fig. 13. We find that the RL-FCM and R-EM can detect three clusters with $c^* = 3$ and the AR of 0.9067 (or 14 error counts) and 0.5200 (or 72 error counts), respectively. Using the FCM by assigning the number of clusters $c = 3$ for the Iris data set, we generally get an average AR

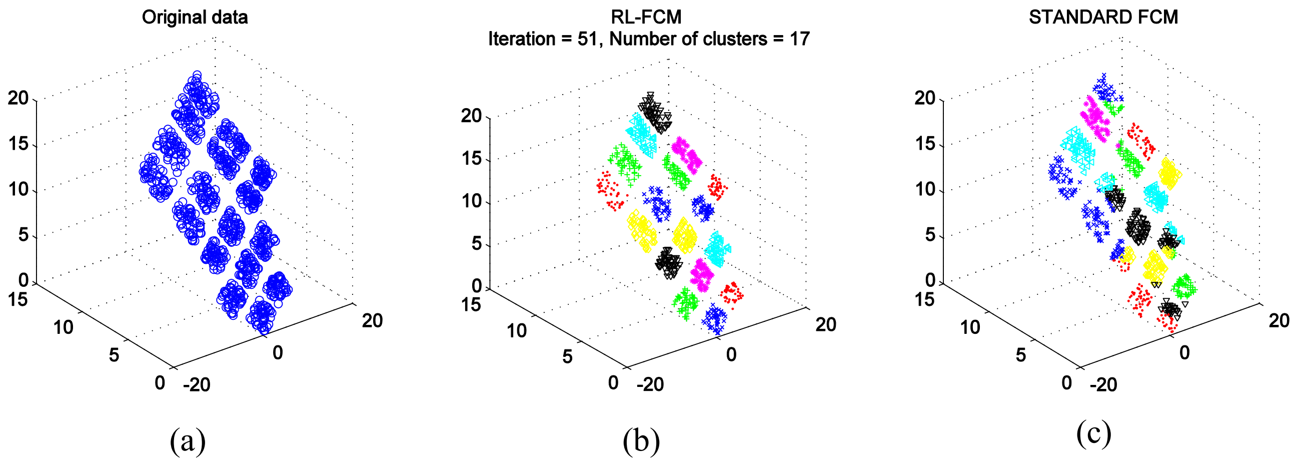

Fig. 11. (a) 17-cluster dataset; (b) Final result from RL-FCM; (c) Final result from FCM with $c = 17$.

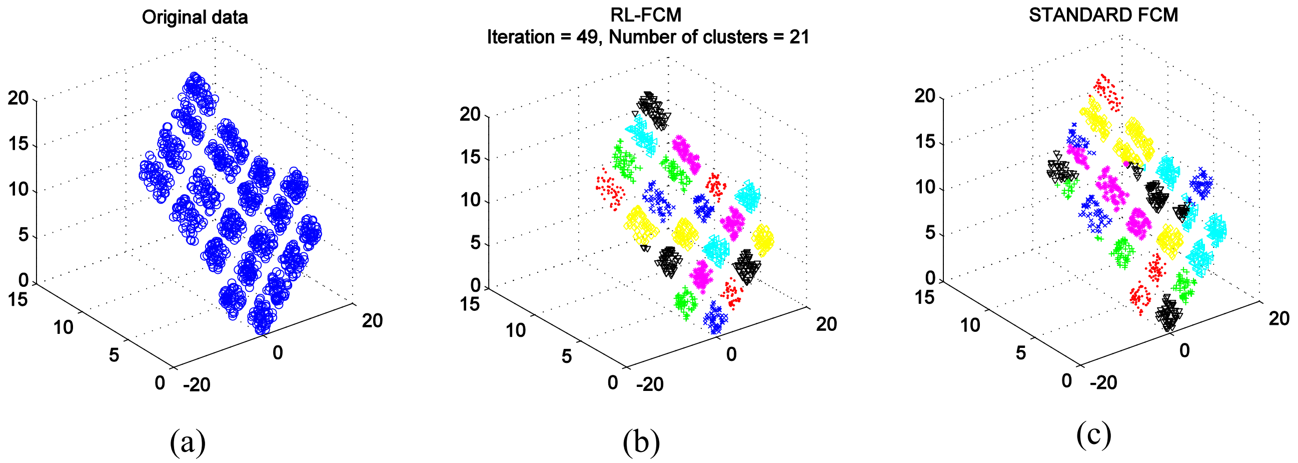

Fig. 12. (a) 21-cluster dataset; (b) Final result from RL-FCM; (c) Final result from FCM with $c = 21$.

Table 5
Cluster number results using different learning functions for r_2 with $r_1 = e^{-t/10}$ (ARs presented inside brackets).

		Ex. 1	Ex. 2	Ex. 3	Ex. 4
	True c	3	5	13	6
r_2	e^{-t}	56	49	46	25
	$e^{-t/10}$	6	4	13 (1.0000)	5
	$e^{-t/100}$	3 (1.0000)	5 (0.9960)	13 (1.0000)	6 (0.9667)
	$e^{-t/1000}$	3 (1.0000)	2	13 (1.0000)	3

Table 6
Validity index values of PC, PE, MPC, MPE, FHV and XB for the Iris data.

c	PC	PE	MPC	MPE	FHV	XB
2	**0.8920**	**0.1961**	**0.7841**	**0.7172**	0.0216	**0.0542**
3	0.7632	0.3959	0.6748	0.6396	0.0205	0.1371
4	0.7065	0.5617	0.6087	0.5948	0.0209	0.1958
5	0.6654	0.6759	0.5818	0.5800	**0.0204**	0.2283
6	0.6065	0.8178	0.5278	0.5436	0.0217	0.3273

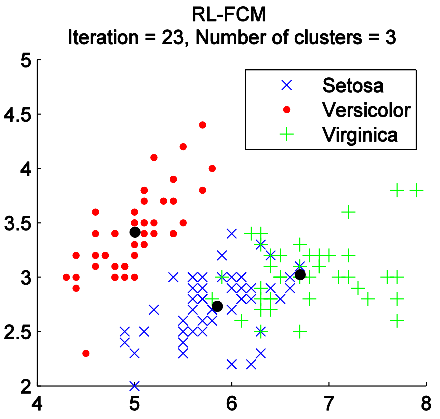

Fig. 13. Final result from RL-FCM for the Iris data set.

of 0.8933 (or 16 error counts). The validity index values of PC, PE, MPC, MPE, FHV and XB for the Iris data set are shown in Table 6, where all validity indices, except the index FHV with $c^* = 5$, give the best number of clusters with $c^* = 2$. However, no validity index can give three clusters with $c^* = 3$. For comparison, with parameter selections, RCA and OB give correct cluster number with average AR = 0.9667 and 0.8975, respectively, while C-FS gives two clusters.

Example 8 (Breast data set)**.** Breast data set consists of 699 instances and 9 attributes, divided into two clusters (benign and malignant) [35]. One attribute with missing value is discarded in this experiment. RL-FCM obtains two clusters with AR = 0.9528, while R-EM obtains seven clusters. Using proper parameter selections, RCA, OB, and C-FS give two clusters with average AR = 0.6552, 0.9471, and 0.7761, respectively. Using FCM with $c = 2$ gives AR = 0.9356.

Table 7
Cluster numbers obtained by RL-FCM, R-EM, RCA, OB, and C-FS using different parameter selections.

Dataset	True c	RL-FCM	R-EM	RCA	OB	C-FS
Ex. 1	3	3	3	3, 4	3, 4	3
Ex. 2	5	5	5	3, 4, 5	5, 6, 7, 8	5
Ex. 3	13	13	15	9, 13, 19, 27, 31, 32	2, 3, 4	2, 3, 4, 5, 13, 14
Ex. 4	6	6	5	4, 5, 6, 7, 8, 9, 10	4, 6	2, 3, 5, 6
Iris	3	3	3	2, 3	2, 3	2
Breast	2	2	7	2	2, 3	2
Seeds	3	3	3	2, 3, 4	2	2, 3, 4

Table 8
Percentages of RCA, OB, and C-FS that obtain the correct cluster number c using 60 different parameter selections.

Dataset	RCA	OB	C-FS
Ex. 1	96.67%	81.67%	100%
Ex. 2	16.67%	73.33%	100%
Ex. 3	1.67%	0%	35%
Ex. 4	15%	25%	40%
Iris	41.67%	13.33%	0%
Breast	1.67%	16.67%	50%
Seeds	68.33%	0%	65%

Table 9
Average AR and RI from RL-FCM, R-EM, RCA, OB, C-FS and FCM using the true number c of clusters.

Dataset		RL-FCM	R-EM	RCA	OB	C-FS	FCM
Ex. 1	AR	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	RI	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Ex. 2	AR	0.9960	1.0000	0.9980	0.9925	0.9845	0.9960
	RI	0.9967	1.0000	0.9984	0.9940	0.9930	0.9967
Ex. 3	AR	1.0000	–	1.0000	–	0.9136	0.8736
	RI	1.0000	–	1.0000	–	0.9486	0.9795
Ex. 4	AR	0.9667	–	0.8633	0.9664	0.9192	0.8910
	RI	0.9794	–	0.9400	0.9705	0.9558	0.9551
Iris	AR	0.9067	0.5200	0.9667	0.8975	–	0.8933
	RI	0.8923	0.7212	0.9575	0.8836	–	0.8797
Breast	AR	0.9528	–	0.6552	0.9471	0.7761	0.9356
	RI	0.9099	–	0.5475	0.8996	0.7070	0.8794
Seeds	AR	0.8952	0.8857	0.9034	–	0.7593	0.8952
	RI	0.8744	0.8677	0.8843	–	0.7683	0.8744

Example 9 (Seeds data set)**.** Seeds data set consists of 210 instances and 7 attributes, divided into three clusters [35]. RL-FCM and R-EM obtain correct cluster number with $\text{AR} = 0.8952$ and 0.8857, respectively. Using proper parameter selections, RCA and C-FS gives three clusters with average $\text{AR} = 0.9034$ and 0.7593, respectively, while OB detects this dataset as two clusters. Using FCM with $c = 3$ gives $\text{AR} = 0.8952$.

We next make more comparisons of the proposed RL-FCM with R-EM, RCA, OB, and C-FS. We implement RL-FCM, R-EM, RCA, OB, and C-FS on the data sets of Examples 1–4 and 7–9 using different parameter selections that are required by RCA, OB, and C-FS. We summarize these obtained numbers of clusters from the algorithms as shown in Table 7. Because RCA, OB, and C-FS need some parameter selections, they obtain several possible numbers of clusters that depend on parameter selections. However, for RL-FCM and R-EM, because they are algorithms with no initialization and no parameter selection, they obtain only one number of clusters. Note that, for RCA, required parameter selections are time constant, discarding threshold, and an initial value. For OB, we need to assign initial focal points, an increasing value, and also initial cluster centers. While for C-FS, the selection for cutoff distance and a decision window are required. Furthermore, we run RCA, OB, and C-FS by using 60 different parameter selections on the data sets of Examples 1–4 and 7–9. We then calculate the percentages that obtain the correct number of clusters under these 60 different parameter selections by RCA, OB, and C-FS. These percentages with the correct number of clusters are shown in Table 8. From Tables 7 and 8, we find that the proposed RL-FCM obtain the correct number of clusters that always presents better results than R-EM, RCA, OB, and C-FS.

Furthermore, we make comparisons of average AR when the true number c of clusters is assigned for RL-FCM, FCM, RCA, OB, C-FS, and R-EM. Besides AR, we also consider the Rand Index (RI). In 1971, Rand [36] proposed objective criteria for the evaluation of clustering methods, known as RI. Up to now, RI had popularly used for measuring similarity between two clustering partitions. Let C be the set of original clusters in a data set and C^* be the set of clusters obtained by the clustering algorithm. For a pair of points (x_i, x_j), a is the number of pairs if both points belong to the same cluster in C and C^*, b is the number of pairs if both points belong to the same cluster in C and different clusters in C^*, c is the number of pairs if both points belong to two different clusters in C and the same cluster in C^*, and d is the number of pairs if both points belong to the two different clusters in C and C^*. The RI is defined by $\text{RI} = (a + d)/(a + b + c + d)$, and so the larger RI is, the better clustering performance is. These average AR and RI with the true number c of clusters assigned to RL-FCM, FCM, RCA, OB, C-FS, and R-EM are shown in Table 9. From Tables 8 and 9, we find that the proposed RL-FCM always presents better accuracy than these existing clustering algorithms.

In next example, we compare the capability of finding the number of clusters from RL-FCM, R-EM and validity indexes for real data sets that have higher number of clusters. These are libra, soybean (large) and letter from UCI Machine Learning Repository [35].

Example 10. For a concern about the capability of finding the number of clusters using RL-FCM for real data with more number of clusters, in this example, we use three real data sets, libra, soybean (large), and letter [35]. Note that, for soybean, since the data set has missing values, we discard the points with missing values. The original n and true c are 307 and 19, respectively. But, after discarding these points with missing values, it has 266 data points with 15 clusters. The data number n, feature component d and true cluster number c are described in Table 10. These finding results for the numbers of clusters from RL-FCM, R-EM and validity indexes are also shown in Table 10. We see that it is difficult to find the exactly true number of clusters by these methods. However, the RL-FCM algorithm can find the numbers of clusters that are always the most closed to the true numbers of clusters.

We also consider real datasets of USPS handwriting [37], Olivetti face [30], and ovarian cancer [38] that have higher attribute dimensions.

Example 11. In this example, we consider three real datasets with high attribute dimensions. These are USPS handwriting (150×256) [37], Olivetti face (100×1024) [30], and ovarian can-

Table 10
Three real data sets with more number of clusters.

Data set	$n \times d$	True c	Obtained c by							
			RL-FCM	R-EM	PC	PE	MPC	MPE	FHV	XB
Libra	360×90	15	13	2	2	2	2	20	2	2
Soybean (large)	266×36	15	13	8	2	2	2	5	4	2
Letter	$20{,}000 \times 17$	26	22	1	2	2	3	3	2	2

Table 11
Cluster numbers obtained by RL-FCM, R-EM, PC, PE, MPC, MPE, FHV and XB.

Data set	$n \times d$	True c	Obtained c by							
			RL-FCM	R-EM	PC	PE	MPC	MPE	FHV	XB
USPS	150×256	10	8	1	2	2	2	8	2	2
Olivetti face	100×1024	10	9	1	2	2	2	15	2	2
Ovarian cancer	216×4000	2	3	1	2	2	2	2	2	2

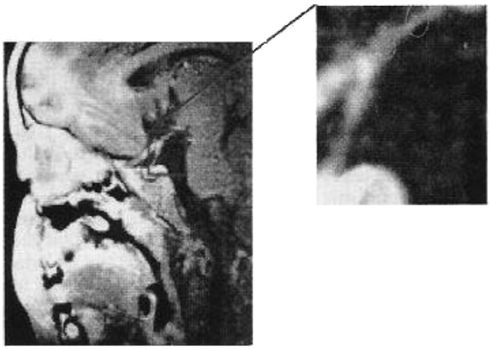

Fig. 14. The brain MR image with its window selection image.

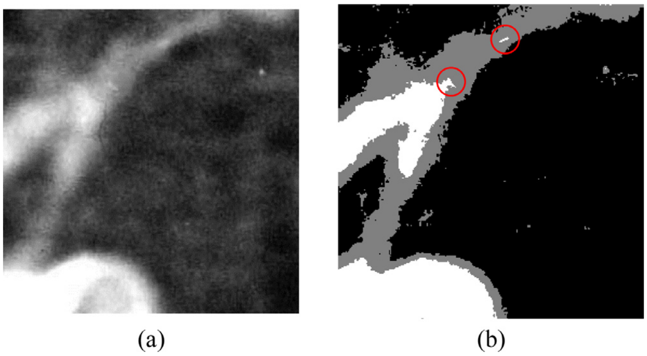

Fig. 15. (a) Window selected MR image; (b) Clustering with the RL-FCM.

cer (216 × 4000) [38]. We implement RL-FCM, R-EM, PC, PE, MPC, MPE, FHV and XB for the three datasets. These obtained number of clusters by RL-FCM, R-EM, PC, PE, MPC, MPE, FHV and XB are shown in Table 11. We find that it becomes more difficult to get the correct number of clusters. The R-EM cannot get the correct number of clusters. For ovarian cancer, all validity indices get the correct number of clusters, but these validity indices cannot get the correct number of clusters for USPS handwriting and Olivetti face. The RL-FCM cannot get the correct number of clusters, but the obtained numbers of clusters from RL-FCM are very closed to the true number c of clusters for the three datasets.

Beside synthetic and real datasets, some images are given for our following experiments.

Example 12 (MRI image)**.** In this example, we consider the Ophthalmology MRI image taken from Yang et al. [39]. This MRI image is from a 2-year-old patient. She was diagnosed with Retinoblastoma in her left eye, an inborn malignant neoplasm of the retina with frequent metastasis beyond the lacrimal cribrosa. The CT scan image showed a large tumor with calcification occupying the vitreous cavity of her left eye. The first MR image was acquired with its grayscale image $400 \times 286 = 114{,}400$ pixels. This MR image showed that an intra-muscle cone tumor mass with high T1-weight image signals and low T2-weight image signals noted in the left eyeball. The tumor was measured 20 mm in diameter and occupied nearly the whole vitreous cavity. Since a shady signal abnormality along the optic nerve to the level of the optic chiasm toward the brain is suspected, the second MR image of the brain MRI image $283 \times 292 = 82{,}636$ pixels with its window selection image was acquired and analyzed, as shown in Fig. 14. From the picture, one lesion was clearly seen by MR image. However, some vague shadows of lesions were suspected with tumor invasion. These suspected abnormalities are not easily detectable. For the purpose of detecting these abnormal tissues, a window of the area around chiasma is selected from the brain MR image as shown in Fig. 15(a). According to [39], Ophthalmologist recommended this MRI to be categorized into three groups: connective tissue, nervous tissue, and tumor tissue. Using the RL-FCM algorithm, we also get three clusters with $c^* = 3$ in 44 iterations, as shown in Fig. 15(b). We can see occult lesions clearly enhanced with the proposed RL-FCM algorithm (marked by circle).

Example 13 (Lena image)**.** In this example, we use Lenna image with 256×256 pixels from Matlab database, as shown in Fig. 16(a). Using the RL-FCM algorithm, we get seven clusters with $c^* = 7$ in 117 iterations, as shown in Fig. 16(c). We can see the RL-FCM algorithm gives clear segmentation results. From the histogram, as shown in Fig. 16(b), it can be seen that the number of peaks is more than six. We also use cluster validity indices of PC, PE, MPC, MPE, FHV and XB to find the best number of clusters. The charts of PC, PE, MPC, MPE, FHV, and XB validity indexes are shown in Fig. 17, where the best numbers of clusters c^* of PC, PE, MPC, MPE, FHV and XB are 2, 2, 98, 100, 4, and 5, respectively.

Example 14 (Peppers image)**.** In this example, we use peppers image with 384×512 pixels from Matlab database, as shown in Fig. 18(a). Using the RL-FCM algorithm, we get six clusters with $c^* = 6$ in 132 iterations, as shown in Fig. 18(c). The results of cluster validity indices PC, PE, MPC, MPE, FHV and XB are 2, 2, 100, 100, 17, and 15, respectively. We can see the RL-FCM algorithm gives clear segmentation results. From the histogram, as shown in Fig. 18(b), it can be seen that the number of peaks is around five to six.

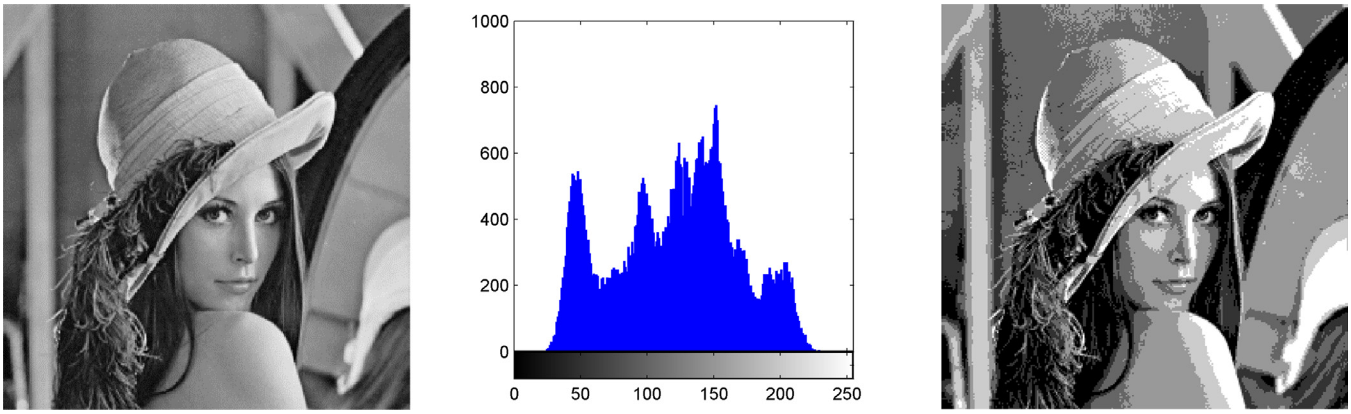

Fig. 16. (a) Original Lena image; (b) Histogram of (a); (c) Clustering using RL-FCM.

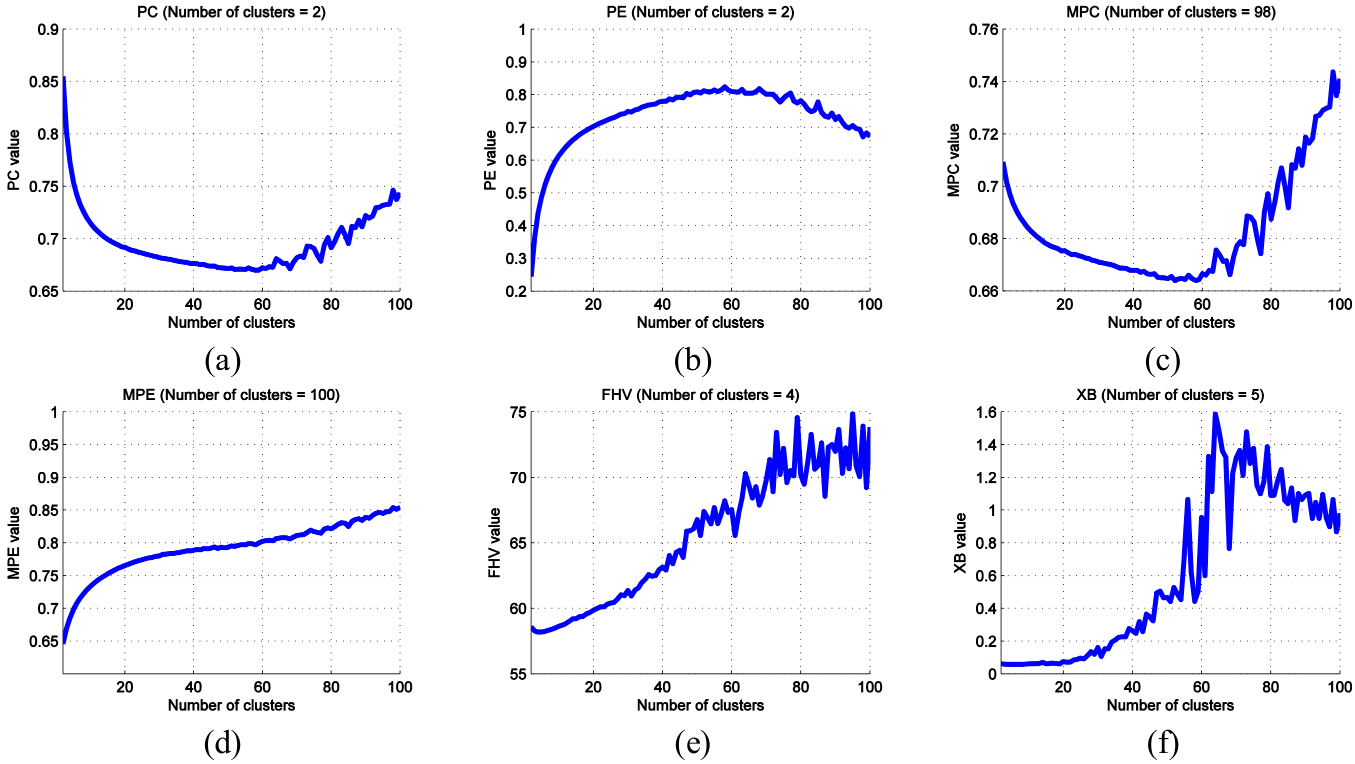

Fig. 17. Plot of validity indexes for Fig. 16(a): (a) PC; (b) PE; (c) MPC; (d) MPE; (e) FHV; (f) XB.

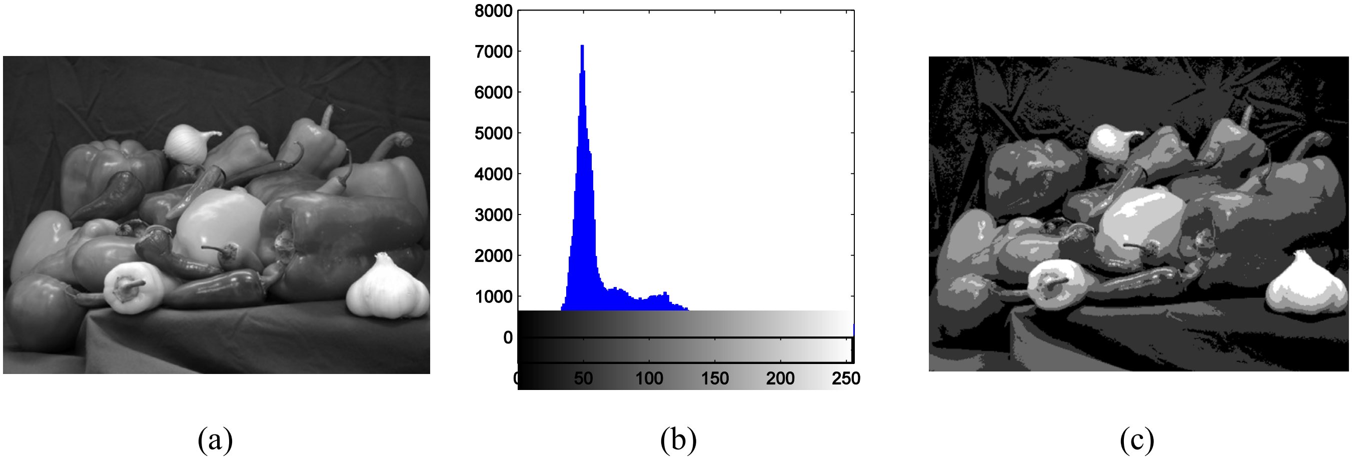

Fig. 18. (a) Original peppers image; (b) Histogram of (a); (c) Clustering using RL-FCM.

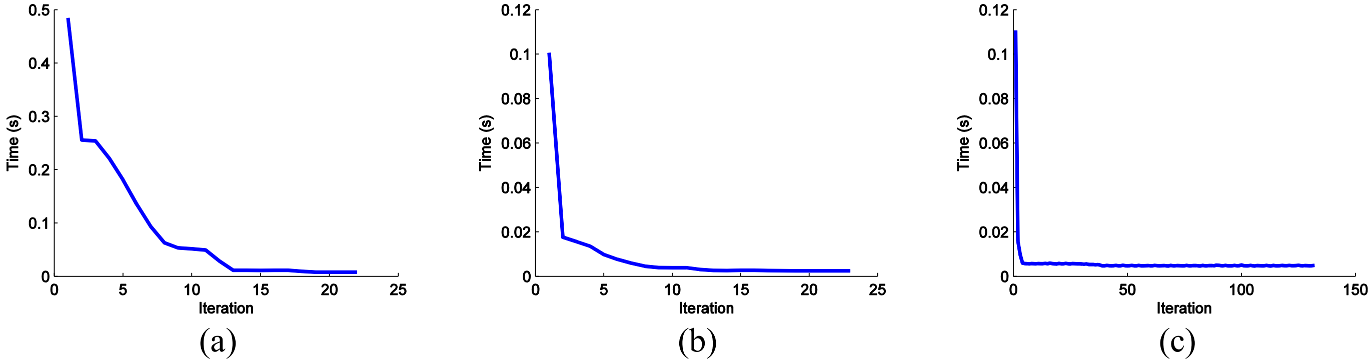

Fig. 19. Plots of per iteration time as implementing RL-FCM for different data sets: (a) Gaussian mixture model of Example 1; (b) Iris data set; (c) Peppers image.

Finally, we would like to demonstrate that, although the algorithm uses the number of data points as the number of clusters (i.e., $c = n$) in the beginning iteration, the iteration time will decrease rapidly after several iterations. This situation occurs because the clusters with $\alpha_k \leq 1/n$ will be discarded during iterations, so that the number of clusters c decreases rapidly. To demonstrate this phenomenon, we show the running time in seconds per iteration for the data sets of Example 1 (Gaussian mixture model with $c = 3$), Example 7 (Iris dataset), and Example 14 (Peppers image) as shown in Fig. 19(a)–(c), respectively. As we can see, the running time decreases rapidly after the 10th iteration.

4. Conclusions

In this paper we proposed a new schema with a learning framework for fuzzy clustering algorithms, especially for the fuzzy c-means (FCM) algorithm. We adopted the merit of entropy-type penalty terms for adjusting the bias and also free of the fuzziness index in the FCM. We then created a robust-learning FCM (RL-FCM) clustering algorithm. The proposed RL-FCM uses the number of data points as initial number of clusters for solving initialization problem. It then discards these clusters that have the mixing proportion values less than one over the number of data points, such that the best cluster number can be automatically found according to the structure of data. The advantages of RL-FCM are free of initializations and parameters that also robust to different cluster volumes and shapes, noisy points and outliers with automatically finding the best number of clusters. The computational complexity of RL-FCM was also analyzed. The main difference for computational time between RL-FCM and FCM is the beginning iteration for assigning the number of data points as initial number of clusters. However, in general, the iteration time for RL-FCM will decrease rapidly after several iterations. On the other hand, for very large data sets, we may consider by using a grid base to divide dimensions of feature space into grids and then only choosing one data point in each grid as the initial cluster center so that we can reduce the computational time for RL-FCM.

Several numerical data and real data sets with MRI and image segmentation are used to show these good aspects of RL-FCM. Experimental results and comparisons actually demonstrated the effectiveness and superiority of the proposed RL-FCM algorithm. Except these experiments in the paper, the RL-FCM algorithm could be applied to text mining, face recognition, marketing segmentation and gene expression. As a whole, the proposed RL-FCM is actually an effective and useful robust learning-based clustering algorithm. Although the RL-FCM algorithm actually improves the performance of fuzzy clustering algorithms, especially capable of obtaining the number of clusters, it still has limitations in handling high-dimensional data sets. We think that, for high-dimensional data, a suitable schema with feature selection should be considered. For our future work, we will consider data with high dimensions by building a new feature selection procedure in the RL-FCM algorithm.

Acknowledgments

The authors would like to thank the anonymous referees for their helpful comments in improving the presentation of this paper. This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 105-2118-M-033-004-MY2.

References

[1] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data: an Introduction to Cluster Analysis, Wiley, New York, 1990.
[2] G.J. McLachlan, K.E. Basford, Mixture Models: Inference and Applications to Clustering, Marcel Dekker, New York, 1988.
[3] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm (with discussion), J. R. Stat. Soc. Series B 39 (1977) 1–38.
[4] S.E. Schaeffer, Graph clustering, Comput. Sci. Rev. I (2007) 27–64.
[5] C.C. Aggarwal, J.L. Wolf, P.S. Yu, Method for targeted advertising on the web based on accumulated self-learning data, clustering users and semantic node graph techniques, U.S. Patent No. 6714975 (2004).
[6] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, 1, University of California Press, 1967, pp. 281–297.
[7] D. Pelleg, A. Moore, X-Means: extending K-means with efficient estimation of the number of clusters, in: Proceedings of the 17th International Conference on Machine Learning, San Francisco, 2000, pp. 727–734.
[8] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.
[9] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York, 1981.
[10] M.S. Yang, A survey of fuzzy clustering, Math. Comput. Model. 18 (1993) 1–16.
[11] A. Baraldi, P. Blonda, A survey of fuzzy clustering algorithms for pattern recognition Part I and II, IEEE Trans. Syst. Man Cybern. Part B 29 (1999) 778–801.
[12] F. Hoppner, F. Klawonn, R. Kruse, T. Runkler, Fuzzy Cluster Analysis: Methods for Classification Data Analysis and Image Recognition, Wiley, New York, 1999.
[13] E. Ruspini, A new approach to clustering, Inf. Control 15 (1969) 22–32.
[14] J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact, well-separated clusters, J. Cybern. 3 (1974) 32–57.
[15] D.E. Gustafson, W.C. Kessel, Fuzzy clustering with a fuzzy covariance matrix, in: Proceedings of IEEE CDC, California, 1979, pp. 761–766.
[16] N.B. Karayiannis, MECA: maximum entropy clustering algorithm, in: Proceedings of IEEE International Conference on Fuzzy Systems, 1, Orlando, FL, 1994, pp. 630–635.
[17] S. Miyamoto, K. Umayahara, Fuzzy clustering by quadratic regularization, in: Proceedings of the 7th IEEE International Conference on Fuzzy Systems, 2, Piscataway, NJ, 1998, pp. 1394–1399.
[18] C. Wei, C. Fahn, The multisynapse neural network and its application to fuzzy clustering, IEEE Trans. Neural Netw. 13 (2002) 600–618.
[19] R.J. Hathaway, J.C. Bezdek, Y. Hu, Generalized fuzzy c-means clustering strategies using L_p norm distances, IEEE Trans. Fuzzy Syst. 8 (2000) 576–582.
[20] M.S. Yang, K.L. Wu, J.N. Hsieh, J. Yu, Alpha-cut implemented fuzzy clustering algorithms and switching regressions, IEEE Trans. Syst. Man Cybern. Part B 38 (2008) 588–603.
[21] T.C. Havens, J.C. Bezdek, C. Leckie, L.O. Hall, M. Palaniswami, Fuzzy c-means algorithms for very large data, IEEE Trans. Fuzzy Syst. 20 (2012) 1130–1146.
[22] H. Izakian, W. Pedrycz, I. Jamal, Clustering spatiotemporal data: an augmented fuzzy c-means, IEEE Trans. Fuzzy Syst. 21 (2013) 855–868.

[23] J.C. Bezdek, Numerical taxonomy with fuzzy sets, J. Math. Biol. 1 (1974) 57–71.

[24] J.C. Bezdek, Cluster validity with fuzzy sets, J. Cybern. 3 (1974) 58–73.

[25] M. Roubens, Pattern classification problems with fuzzy sets, Fuzzy Sets Syst. 1 (1978) 239–253.

[26] R.N. Dave, Validating fuzzy partition obtained through c-shells clustering, Pattern Recognit. Lett. 17 (1996) 613–623.

[27] I. Gath, A.B. Geva, Unsupervised optimal fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 11 (1989) 73–781.

[28] X.L. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 13 (1991) 841–847.

[29] H. Frigui, R. Krishnapuram, A robust competitive clustering algorithm with applications in computer vision, IEEE Trans. Pattern Anal. Mach. Intell. 21 (6) (1999) 450–465.

[30] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, Science 344 (6191) (2014) 1492–1496.

[31] P. Fazendeiro, J.V. de Oliveira, Observer-biased fuzzy clustering, IEEE Trans. Fuzzy Syst. 23 (2015) 85–97.

[32] D. Dembélé, P. Kastner, Fuzzy c-means method for clustering microarray data, Bioinformatics 19 (2003) 973–980.

[33] V. Schwämmle, O.N. Jensen, A simple and fast method to determine the parameters for fuzzy c-means cluster analysis, Bioinformatics 26 (2010) 2841–2848.

[34] M.S. Yang, C.Y. Lai, C.Y. Lin, A robust EM clustering algorithm for Gaussian mixture models, Pattern Recognit. 45 (2012) 3950–3961.

[35] C.L. Blake, C.J. Merz, UCI repository of machine learning databases, a huge collection of artificial and real-world data sets, 1998.

[36] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Amer. Stat. Assoc. 66 (1971) 846–850.

[37] J.J. Hull, A database for handwritten text recognition research, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1994) 550–554.

[38] T.P. Conrads, M. Zhou, E.F. Petricoin III, L. Liotta, T.D. Veenstra, Cancer diagnosis using proteomic patterns, Expert Rev. Mol. Diagn. 3 (2003) 411–420.

[39] M.S. Yang, Y.J. Hu, K.C.R. Lin, C.C.L. Lin, Segmentation techniques for tissue differentiation in MRI of ophthalmology using fuzzy clustering algorithms, Magn. Reson. Imaging 20 (2002) 173–179.

Miin-Shen Yang received the BS degree in mathematics from the Chung Yuan Christian University, Chung-Li, Taiwan, in 1977, the MS degree in applied mathematics from the National Chiao-Tung University, Hsinchu, Taiwan, in 1980, and the PhD degree in statistics from the University of South Carolina, Columbia, USA, in 1989.

In 1989, he joined the faculty of the Department of Mathematics in the Chung Yuan Christian University (CYCU) as an Associate Professor, where, since 1994, he has been a Professor. From 1997 to 1998, he was a Visiting Professor with the Department of Industrial Engineering, University of Washington, Seattle. During 2001–2005, he was the Chairman of the Department of Applied Mathematics in CYCU. His current research interests include clustering, pattern recognition, machine learning, and neural fuzzy systems.

Dr. Yang was an Associate Editor of the IEEE Transactions on Fuzzy Systems (2005–2011), and is an Associate Editor of the Applied Computational Intelligence & Soft Computing and Editor-in-Chief of Advances in Computational Research. He was awarded with 2008 Outstanding Associate Editor of IEEE Transactions on Fuzzy Systems, IEEE; 2009 Outstanding Research Award of CYCU; 2012–2018 Distinguished Professorship of CYCU; 2016 Outstanding Research Award of CYCU.

Yessica Nataliani received the BS degree in mathematics and the MS degree in computer science from the Gadjah Mada University, Yogyakarta, Indonesia, in 2004 and 2006, respectively.

She is currently a Ph.D. student at the Department of Applied Mathematics in the Chung Yuan Christian University, Taiwan. His research interests include cluster analysis and pattern recognition.