INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING, *SERIES B*
Volume 5, Number 1-2, Pages 113–122

DEVELOPMENT OF A WEIGHTED FUZZY C-MEANS CLUSTERING ALGORITHM BASED ON JADE

KANGSHUN LI, CHUHU ZHANG, ZHANGXIN CHEN, AND YAN CHEN

Abstract. To overcome the shortcomings of falling into local optimal solutions and being too sensitive to initial values of the traditional fuzzy C-mean clustering algorithm, a weighted fuzzy C-means (FCM) clustering algorithm based on adaptive differential evolution (JADE) is proposed in this paper. To consider the particular contributions of different features, a ReliefF algorithm is used to assign the weight for each feature. A weighted morphology-similarity distance (WMSD) based on ReliefF instead of the Euclidean distance is used to improve the objective function of the FCM clustering algorithm. Experimental results on the international standard Iris data and the contrast experimental results with other evolution algorithms show that the proposed algorithm has higher clustering accuracy and greater searching capability.

Key words. Fuzzy C-means, adaptive differential evolution, weighted morphology-similarity distance, clustering precision

1. Introduction

Clustering is to divide the given unknown-type sample data into some meaningful or useful clusters according to a particular standard in a way that the objects in the same cluster are very similar and the objects in different clusters are very different. The purpose of a clustering analysis that now covers the fields of statistics, biology and machine learning such as data mining and pattern recognition is to reveal the internal structure of the underlying data.

Unlike the traditional enforced division, the boundary of a fuzzy clustering analysis is not clear and it does not have the "either-or" property [1], which indicates the fact that data in real life is of intermediary nature. A fuzzy clustering method considers an affiliation relation between sample data and cluster centers, and this relation is extended from two values $\{0, 1\}$ to $[0, 1]$ in order to represent the fuzziness of data and reflect the uncertainty of real world matters better. Fuzzy C-means (FCM) clustering is one of the most widely applied fuzzy clustering algorithms. It was proposed by Dumn (1973) when he was promoting the hard C-means (HCM) [2], and was introduced to a clustering analysis by Bezdek (1981) [3]. The principles of the least squares and iterative gradient descent methods are applied to the classic FCM, which result in the traditional FCM that runs effectively only with spherical or ellipsoidal clustering and is extremely sensitive to noise and outliers [4]. In addition, it has the following defects: (1) Its results are strongly influenced by the initial cluster center and (2) because of using the gradient descent method, it is easily falling into local optimal solutions and cannot gain the global optimal solution. To overcome these defects, in recent years many researchers have focused on the direction of combining global intelligent algorithms with the FCM clustering. Paper [5] combined a global Genetic Algorithm(GA) with a local climbing technique, and an improved GA based on a weighted FCM clustering algorithm

Received by the editors February 12, 2014 and, in revised form, March 20, 2014.
This work is supported by the Guangdong Province Science and Technology Research Project with the Grant (No.2012A020602037) and the Research Project of Science and Technology of Education Department of Jiangxi Province of China with the Grant (No.GJJ12348).

was proposed in paper [6] by using a Gaussian mutation operator and multiple correlation coefficients based on a weighted Euclidean distance instead of the standard Euclidean distance. Both algorithms achieved good clustering results. An improved algorithm based on Particle Swarm Optimization (PSO) using field operation was proposed [7]; it also effectively obtained results on an Iris data set. Other researchers improved FCM using a distance correction factor [8] and a kernel-based learning approach [9]. FCM was analyzed based on different similarity estimation methods [10].

Differential Evolution (DE) was first proposed by Storn and Price (1996) [11, 12]. Compared with the general evolutionary algorithm, DE was proved to converge to the global optimal solution more quickly and be more stable. Compared with other intelligent algorithms, DE's main characteristic is its differential mutation. It is easy to implement, runs with high operating efficiency, and has only three parameters: population size NP, mutation factor F and crossover probability CR. However, the classical DE is sensitive to the mutation factor F and the crossover probability CR; the user must assign different F and CR to different questions. In order to overcome this disadvantage, a weighted fuzzy clustering algorithm based on an adaptive differential evolutionary algorithm (JADE) [13] is proposed in this paper. This algorithm utilizes the JADE global searching capability, and takes into account different contribution of each dimensional feature of a vector to the pattern classification. A weighted morphology-similarity distance based on ReliefF instead of the standard Euclidean distance is used to improve the objective function of the FCM clustering algorithm. Experimental results for this new algorithm compared with general methods show its advantage in terms of better clustering precision of the algorithm.

The organization of this paper is as follows. Section 2 gives a brief introduction to the standard FCM. The classical DE algorithm and the JADE algorithm will be depicted, respectively, in Sections 3 and 4. Section 5 outlines the weighted fuzzy clustering algorithm based on JADE. Section 6 gives the results of simulation and performance evaluation. Finally, conclusions are stated in Section 7.

2. Standard Fuzzy C-means Algorithm

FCM, in accordance with the principle of the least squares method, calculates an objective function mean squares error and iteratively optimizes the objective function to achieve the fuzzy classification of a data set. The basic idea is as follows: Dividing the data set $X = \{x_1, x_2, ..., x_n\} \in \mathbf{R}^n$ into C clusters, where $2 \leq C \leq n$, the clustering result is represented as an affiliation matrix $U = [u_{ik}]$ that satisfies: $u_{ik} \in [0, 1]$, $\sum_{i=1}^{C} u_{ik} = 1 \ \forall k = 1, 2, ..., n$ and $0 < \sum_{k=1}^{n} u_{ik} < n \ \forall i = 1, 2, 3, ..., C$. u_{ik} represents the x_k affiliation value to cluster i. Let $V = \{v_1, v_2, ..., v_C\}$ be the set of C cluster centers. FCM is implemented by minimizing the objective function $J_m(U, V)$ with the affiliation matrix U and the cluster center V:

$$J_m(U, V) = \sum_{k=1}^{n} \sum_{i=1}^{C} (u_{ik})^m d_{ik}^2(x_k, v_i). \tag{1}$$

In formula (1), $m \in [1, \infty)$ is a weighted fuzzy exponent that controls the fuzzy degree of the integer partition. For $m = 1$, the fuzzy clustering will be degraded to the hard C-means clustering. $d_{ik}(x_k, v_i)$ is the Euclidean distance of the sample

datum k to the cluster center i:

$$d_{ik}(x_k, v_i) = \left\| x_k - v_i \right\|. \tag{2}$$

FCM is implemented through iteratively optimizing the objective function $J_m(U, V)$. The concrete procedure is given as follows:

Step 1: Initiate the cluster number C, the weighted fuzzy exponent m and the cluster center $V = \{v_1, v_2, ..., v_C\}$.

Step 2: Calculate the affiliation matrix U:

$$u_{ik} = \frac{1}{\sum\limits_{j=1}^{C} \left(\frac{d_{ik}(x_k, v_i)}{d_{jk}(x_k, v_j)} \right)^{\frac{2}{m-1}}}, \quad k = 1, 2, ..., n. \tag{3}$$

Step 3: Update the cluster center V:

$$v_i = \frac{\sum\limits_{k=1}^{n} (u_{ik})^m x_k}{\sum\limits_{k=1}^{n} (u_{ik})^m}, \quad i = 1, 2, ..., C. \tag{4}$$

Continuously repeat the second and third steps until the change of the objective function value is smaller than a preset error value or the maximum number of iterations is reached.

3. ReliefF Algorithm

A weighted coefficient matrix employing the ReliefF algorithm [15] is developed in this paper, which weights each feature by using the method proposed in paper [14]. The basic Relief algorithm was proposed by Kira and Rendell in 1992 [16]. It can only be applied to the solution of a problem of two clusters classification. The Relief algorithm was extended to the ReliefF algorithm by Kononenko [15], which is applicable to classification with multiple clusters. It chooses k nearest samples from both its own cluster and other clusters, obtains the weight of each feature by calculating the mean, and provides a certain weight for each feature of the feature set. The detailed algorithm is given as follows.

Assume that $X = \{x_1, x_2, ..., x_n\}$ is the set of a clustering analysis, where $x_i = [x_{i1}, x_{i2}, ..., x_{iN}]^T$ represents N features of x_i. m samples are extracted from X. For any sample x_i, first find k nearest samples $H = \{h_1, h_2, ..., h_k\}$ that are in the same cluster with x_i and k nearest samples $M(C) = \{m_{C1}, m_{C2}, ..., m_{Ck}\}$ that are in different clusters from x_i. The difference between h_j and x_i is calculated by formula (5) below, where $diff\_hit$ is an $N \times 1$ matrix:

$$diff\_hit = \sum_{j=1}^{k} \frac{|x_i - h_j|}{max(X) - min(X)}. \tag{5}$$

Formula (6) below shows the difference of properties between m_{Cj} and x_i, where $diff\_miss$ is an $N \times 1$ matrix and $p(C)$ means the probability of cluster C:

$$diff\_miss = \sum_{x \neq class(x_i)} \frac{p(C)}{1 - p(class(x_i))} \sum_{j=1}^{k} \frac{|x_i - m_{cj}|}{max(X) - min(X)}. \tag{6}$$

The weighted matrix W is updated:

$$W = W - diff\_hit/(k \cdot m) + diff\_miss/(k \cdot m). \tag{7}$$

Initially, W is the zero N-order matrix. m is the iterative number. The weights of N features are obtained by continuously repeating equation (7) m times.

ReliefF was designed to classification where the type of each sample has been tagged but the types of samples are unknown before the clustering analysis. This paper, following [14], achieves the tags of samples by clustering the samples first and then finds the weight matrix of each feature from the feature set by using the method described above.

4. Basic Differential Evolution

Differential Evolution (DE) adopts real encoding. It contains three operations: selection, crossover and mutation. Compared to GA, the mutation operation first selects mutation vectors based on the current population to do the differential operation and gains a new mutant vector by adding them to the target vector. Then it generates a new vector between the new mutant vector and its parent vector according to a mutation strategy. Comparing the new vector with the parent vector, the vector with better fitness will participate in the next generation. It continuously iterates until a termination condition is reached.

The basic strategy of DE can be described as follows:

4.1. Mutation. At each generation, the operation selects a vector as the target vector and two other vectors as differential vectors; a mutant vector is generated according to:

$$v_{i,G+1} = x_{r_1,G} + F \cdot (x_{r_2,G} - x_{r_3,G}), \tag{8}$$

where the indices r_1, r_2, and r_3 are three distinct integers uniformly chosen from the set $\{1, 2, ..., NP\}$, they are also different from the running index i, NP is the population size, F is the mutation factor, G is the current generation, $G + 1$ represents the next generation, $x_{i,G}$ means vector i at generation G, and $v_{i,G+1}$ is the mutant vector.

4.2. Crossover. After mutation, a binomial crossover operation forms a trial vector:

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if}(randb(j) \le CR \quad \text{or} \quad j = rnbr(j)), \\ x_{ji,G} & \text{if}(randb(j) > CR \quad \text{and} \quad j \ne rnbr(j)), \end{cases} \tag{9}$$

where $randb(j)$ is the j^{th} evaluation of a uniform random number generator with an outcome belonging to [0,1], CR is the crossover constant belonging to [0,1] which is determined by the user, $rnbr(j)$ is a randomly chosen index belonging to $\{1, 2, ..., D\}$ which ensures that $u_{i,G+1}$ gets at least one parameter from $v_{i,G+1}$, and $x_{ji,G}$ is the j^{th} component of $x_{i,G}$ at generation G, which means $x_{i,G} = (x_{1i,G}, x_{2i,G}, ..., x_{Di,G})$.

4.3. Selection. The operation selects the better one from the parent vector $x_{i,G}$ and the trial vector $u_{i,G+1}$ according to a fitness function $f(\cdot)$. For example, if we have a minimization problem, the selected vector is given by

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & \text{if } f(u_{i,G+1}) < f(x_{i,G}), \\ x_{i,G} & \text{otherwise}, \end{cases} \tag{10}$$

and is used as a parent vector in the next generation.

The DE flowchart can be described as follows:

Step 1: Initiate the parameters: population size NP, mutation factor F, crossover probability CR, max iterative number G_{max}, $G = 1$, and NP D-dimensional vectors according to each dimension's searching range.

Step 2: For each vector in population, calculate its fitness value by the fitness function $f(\cdot)$.

Step 3: Perform the operations of mutation, crossover, and selection by using formulas (8)-(10), respectively.

Step 4: Determine whether the termination condition is reached; if not, go back to Step 3.

5. Adaptive Differential Evolution (JADE)

Though DE is easy to implement, runs with high operating efficiency and has only three parameters, its performance is still quite dependent on the setting of control parameters such as the mutation factor and the crossover probability. A new adaptive differential evolution (JADE) was proposed in [10]. At each generation, the crossover probability of each individual is independently generated according to a normal distribution of mean CR and the mutation factor F of each individual is independently generated according to a Cauchy distribution. One records CR and F of the individual that successfully participates in the differential variation, and then calculates their means and generates new CR and F according to a specialized equation. JADE consists of two kinds of modes: One contains an external population while the other does not. This paper adopts the former one.

The flowchart of JADE is given as follows:

Step 1: Initiate the crossover probability μCR and the mutation factor μF, randomly generate the vector population P with size NP, set the max iteration number G_{max} and the iteration number $G = 0$, and the set of all successful crossover probabilities S_{CR}, the set of all successful mutation factors S_F and the external population A are all initially taken to be the empty set.

Step 2: For each individual i at generation G, generate CR_i according to a normal distribution of mean μCR and standard deviation 0.1 while F_i is generated according to a Cauchy distribution with location parameter μF and scale parameter 0.1. $x^p_{best,G}$ is randomly chosen as one of the top $100p\%$ individuals in the current population; for example, setting $p = 0.1$, $100p\%$ means the top 10% individuals in the current population. Randomly select $x_{r_1,G} \neq x_{i,G}$ from the current population and choose $x_{r_2,G} \neq x_{r_1,G} \neq x_{i,G}$ from the union of the current population P and the external population A. Perform the three operations: selection, crossover and mutation according to formulas (11)-(13), respectively:

$$v_{i,G+1} = x_{i,G} + F_i \cdot (x^p_{best,G} - x_{i,G}) + F_i \cdot (x_{r_1,G} - x_{r_2,G}), \tag{11}$$

$$u_{ij,G+1} = \begin{cases} v_{ij,G+1} & \text{if}(randb(j) \leq CR_i \quad \text{or} \quad j = rnbr(j)), \\ x_{ij,G} & \text{otherwise}, \end{cases} \tag{12}$$

(13)

$$x_{i,G+1} = \begin{cases} x_{i,G} & \text{if}(f(x_{i,G}) \leq f(u_{i,G+1})), \\ u_{i,G+1}; x_{i,G} \to A; CR_i \to S_{CR}; F_i \to S_F & \text{otherwise}. \end{cases}$$

Step 3: After all individuals are traversed, truncate A to make sure that its size is not larger than NP, and update μCR and μF according to formulas (14) and (15), respectively:

$$\mu CR = (1 - c) \cdot \mu CR + c \cdot mean_A(S_{CR}), \tag{14}$$

$$\mu F = (1 - c) \cdot \mu F + c \cdot mean_L(S_F), \tag{15}$$

where c is a positive constant between 0 and 1, $mean_A(S_{CR})$ is the usual arithmetic mean, and $mean_L(S_F)$ is the Lehmer mean calculated by

$$mean_L(S_F) = \frac{\sum_{F \in S_F} F^2}{\sum_{F \in S_F} F}. \tag{16}$$

Step 4: Add unity to the iterator G, and check whether the termination condition is reached; if not, go back to Step 2.

6. Weighted Fuzzy C-means Algorithm Based on JADE

To overcome the shortcomings of falling into the local optimal solutions and being sensitive to the initial values of the traditional fuzzy C-means clustering algorithm, JADE that has global searching ability is used to optimize FCM. Using an Euclidean distance to measure the difference between patterns results in the preference for the super sphere data structure, and is difficult to identify the data structures of other shapes. This paper proposes a weighted morphology-similarity distance (WMSD) instead of the Euclidean distance, which is based on the morphology-similarity distance (MSD) proposed in paper [17].

Set $X = \{x_1, x_2, ..., x_n\}$ as the set for a clustering analysis, where $x_i = [x_{i1}, x_{i2}, ..., x_{iN}]^T$ represents N features of x_i. The distance between x_i and x_j is defined as follows:

$$d_{wmsd}(x_i, x_j) = WL_2 \times (2 - ASD/WL_1), \tag{17}$$

$$WL_1 = \left(\sum_{k=1}^{N} |x_{ik} - x_{jk}| W_k \right), \tag{18}$$

$$WL_2 = \sqrt{\sum_{k=1}^{N} (x_{ik} - x_{jk})^2 W_k}, \tag{19}$$

$$ASD = \left| (\sum_{k=1}^{N} (x_{ik} - x_{jk}) W_k) \right| \tag{20}$$

where $W = (W_k)$ is the weight matrix of the features of the feature set calculating by the RelifF algorithm, WL_1 is the Manhattan distance, WL_2 is the Euclidean distance, and ASD is a weighted absolute value of difference. On one hand, multiple kinds of distances are calculated by WMSD, which makes it easy to identify different shapes of data and becomes more versatile. On the other hand, WMSD considers the particular contributions of different features, which means that the more important the feature is the heavier weight it has. Focusing on the main feature helps WMSD enhance the recognition accuracy.

The purpose of using JADE to optimize FCM is to make sure that the algorithm is not sensitive to the initial value and the key points are how to encode and design the fitness function. The following steps explain these issues:

Step 1: Individual encoding and initiation. Choose a cluster center to encode since the key point of clustering is to recognize the cluster center. Assume that the sample dimension is N and the cluster center number is C so the length of each individual is $N \times C + 1$ and its form is $[v_{11}, v_{12}, ..., v_{1N}, ..., v_{C1}, v_{C2}, ..., v_{CN}, J]$, where J is the individual fitness value.

Step 2: Fitness function. We use WMSD instead of an Euclidean distance; the smaller the fitness value, the better the individual. The fitness function can be describe as follows:

$$J_m(U,V) = \sum_{k=1}^{n}\sum_{i=1}^{C}(u_{ik})^m d^2_{wmsd}(x_k, v_i). \tag{21}$$

In summary, the flowchart of the weighted fuzzy C-means clustering algorithm based on JADE is given as follows:

Step 1: Set the parameters including the population size NP, the max iterative number G_{max}, the crossover probability μCR, the mutation factor μF, and the iterator $G = 0$; the set of all successful crossover probabilities SCR, the set of all successful mutation factors SF and the external population A are initially set to be the empty set.

Step 2: Calculate the weight matrix W of features of the feature set by using the ReliffF algorithm.

Step 3: Initiate population P according to each dimension's searching range and the principle of encoding mentioned above; each individual represents the cluster centers collection. Calculate the distance between a sample and the cluster center by formula (17) and calculate the fitness value by formula (21).

Step 4: For each individual i at generation G, generate CR_i that satisfies the normal distribution of mean and F_i that satisfies the Cauchy distribution according to the basic JADE. Perform the three operations: selection, crossover and mutation according to formulas (11)-(13) and calculate the fitness value by formula (21).

Step 5: After all the individuals are traversed, truncate A to make sure that its size is not larger than NP, and update μCR and μF according to formulas (14) and (15), respectively.

Step 6: Add unity to the iterator, and check whether the maximum number of iterations is reached; if not, go back to Step 4.

7. Simulation Experiments and Performance Evaluation

In order to test the performance of the proposed algorithm we use the Iris data set as a sample test set. The Iris data set is the typical internationally recognized data to check the effect of unsupervised clustering. It contains 150 instances, where each instance consists of four attributes that represent the petal length, petal width, sepal length, and sepal width, respectively. These sample data can be classified into three clusters: Setosa, Versicolor and Virginica, and each cluster includes 50 instances. Setosa is quite different from other two clusters and is linearly separable. Versicolor and Virginica are quite similar and partially overlapping. Paper [14] provides the real cluster centers of the Iris data set: $v_1 = (5.00, 3.42, 1.46, 0.24)$, $v_2 = (5.93, 2.77, 4.26, 1.32)$ and $v_3 = (6.58, 2.97, 5.55, 2.02)$. Set the population size to 60, the max iterative number to 250, and the crossover probability CR and the mutation factor to 0.5. Randomly running 100 times using Matlab, the results are shown in Tables 1 and 2. Table 1 shows the results that compare the proposed algorithm with that in paper [5] that combines the GA and FCM and that in paper [7] that combines PSO and FCM on average accuracy. Table 2 illustrates the results that compare the proposed algorithm with the basic FCM and the improved weighted FCM based on GA that was proposed in paper [6] on the best clustering results.

Table 1 illustrates the rationality to use MSD instead of the Euclidean distance after comparison of the FCM algorithm and the FCM+MSD algorithm.

TABLE 1. The average accuracy of algorithms on real data set

Cluster algorithm	right number	error number	accuracy/%
FCM	134	16	89.33
FCM+MSD	141	9	94.00
Paper [5]	140	10	93.33
Paper [7]	143	7	95.33
The proposed algorithm	144.77	5.23	96.51

TABLE 2. Results of clustering of four algorithms on the Iris data set (l-e-n=least error number, b-c=best accuracy)

Cluster algorithm	l-e-n	b-c/%	cluster center of the best case
FCM	16	89.33	v_1 = (5.0036,3.4030,1.4850,0.2515) v_2 = (5.8892,2.7612,4.3643,1.3974) v_3 = (6.7751,3.0524,5.6469,2.0536)
FCM+MSD	9	94.00	v_1 = (5.0086,3.4170,1.4749,0.2480) v_2 = (6.6887,2.9972,5.5705,2.0070) v_3 = (5.8930,2.7816,4.3225,1.3828)
Paper [6]	4	97.33	v_1 = (5.0068,3.406,1.4755,0.25004) v_2 = (5.9035,2.7496,4.272, 1.336) v_3 = (6.6569,3.0168,5.4246,2.0429)
The proposed algorithm	4	97.33	v_1 = (5.0422,3.4264,1.4688,0.2498) v_2 = (5.9489,2.7635,4.3052,1.3465) v_3 = (6.6022,2.9584,5.5209,1.9972)

TABLE 2. Results of clustering of four algorithms on the Iris data set (continued) (e-n-w-c=error number of the worst case)

Cluster algorithm	sum of squared errors of the best case	e-n-w-c
FCM	0.0750	16
FCM+MSD	0.0229	9
Paper [6]	0.0265	6
The proposed algorithm	0.0071	8

FCM+MSD produces a much better result than FCM. For the average accuracy of the five algorithms, it is obvious that the proposed algorithm can overcome the shortcomings of falling into the local optimal solutions and being sensitive to the initial values of the traditional FCM. The proposed algorithm can yield better results when compared to the other known intelligent algorithms. Table 2 shows the acceptable global searching ability of the proposed algorithm. With running 100 times, the least error number is 4 while the largest number is 8. The difference between the best case and the worst case is 4. Analyzing the results of 100 times, we can see that the case of the error number 8 only exists one time and the case of the error number 7 only appears three times, which means that 96% of the error number is not larger than 6. Although the biggest cluster error number is larger than the result mentioned in paper [6], the cluster center is much better than that in paper [6] and is very close to the real cluster centers, which shows the stability

of the proposed algorithm and that it has higher clustering accuracy and greater searching capability.

8. Conclusion

In this paper, a weighted fuzzy C-means clustering algorithm based on JADE is proposed to overcome the disadvantages of falling into the local optimal solutions and being sensitive to the initial values of the traditional fuzzy C-means clustering algorithm. The proposed algorithm uses the weighted morphology-similarity distance instead of the Euclidean distance to improve the distance function of FCM. The overall performance of the algorithm is improved. The experimental results illustrate that the weighted fuzzy C-means clustering algorithm based on JADE has good robustness and clustering effects.

References

[1] Kurgan L A, Cios K J. CAIM discretization algorithm. IEEE transactions on Knowledge and Data Engineering, 16(2):145-153, 2004.
[2] Dunn J C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. Journal of Cybernetics, 3(3):32-57, 1973.
[3] Bezdek J C. Pattern recognition with fuzzy objective function algorithms. Kluwer Academic Publishers, 1981.
[4] Daoqiang Zhang. Kernel-Based Associative Memories, Clustering Algorithms and their Applications. Nanjing Aerospace University, Nanjing, 2004.
[5] Changjiang Zhu, Ying Zhang. Research of Improved Fuzzy C-means Clustering Algorithm. Journal of Henan University (Natural Science), 42(1):92-95, 2012.
[6] Tongqiang Li, Tianyi Zhou, Bin Wu. Weighted Fuzzy C-mean Clustering Algorithm Based on Iimproved Genetic Algorithm. Jounal of Computer Applications,29(12):260-262, 2009.
[7] Zhongwei Wen, Rongjun Li. Fuzzy C-means Clustering Algorithm Based on Improved PSO. Application Research of Computers, 27(7):2520-2522, 2010.
[8] Xiaojun Lou, Junying Li, Haitao Liu. Improved Fuzzy C-means Clustering Algorithm Based on Distance Correction. Jounal of Computer Applications, 32(3):646-648, 2012.
[9] Sen Zhang, Meiling Zhu, Guangkui Hou. Improved Fuzzy Kernel Clustering Algorithm. Journal of Beijing University of Technology, 38(9):1408-1411, 2012.
[10] Zhong Li, Jinsha Yuan. Cluster Analysis of Fuzzy C-means algorithm Based on Different Similarity Estimation Distances. Computer Engineering and Application, 47(18):17-18, 2011.
[11] Storn R, Price K. Differential evolutionCa simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4):341-359, 1997.
[12] Storn R, Price K. Minimizing the real functions of the ICEC'96 contest by differential evolution. Proceedings of IEEE International Conference on Evolutionary Computation, 1996.
[13] Zhang J, Sanderson A C. JADE: adaptive differential evolution with optional external archive., IEEE Transactions on Evolutionary Computation, 13(5):945-958, 2009.
[14] Jie Li, Xinbo Gao, Licheng Jiao. A New Feature Weighted Fuzzy Clustering Algorithm. Acta Electronica Sinica, 34(1):89-92, 2006.
[15] Kononenko I. Estimating attributes: analysis and extensions of RELIEF[C].ECML-94 Proceedings of the European conference on machine learning on Machine Learning,1994.New York : Springer-Verlag, 1994.
[16] Kira K R L. A practical approach to feature selection [C]. ML92 Proceedings of the ninth international workshop on Machine learning, 1992. San Francisco: Morgan Kaufmann Publishers, 1992.
[17] Li Z, Yuan J, Zhang W. Fuzzy C-mean algorithm with morphology similarity distance. Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009.

College of Information, South China Agricultural University, Guangzhou 510642, China.

School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China.
E-mail: likangshun@sina.com

College of Information, South China Agricultural University, Guangzhou 510642, China.
E-mail: 813777109@qq.com

Center for Computational Geosciences, College of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China. Department of Chemical and Petroleum Engineering, Schulich School of Engineering, University of Calgary, 2500 University Drive N.W., Calgary, Alberta, Canada
E-mail: zhachen@ucalgary.ca

College of Information, South China Agricultural University, Guangzhou 510642, China.
E-mail: 641141629@qq.com