Neurocomputing 188 (2016) 233–238

Contents lists available at ScienceDirect

Neurocomputing

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

Kernel-based fuzzy c-means clustering algorithm based on genetic algorithm

Yi Ding, Xian Fu *

Department of Computer Science and Technology, Hubei Normal University, Huangshi 435002, China

ARTICLE INFO

Article history:
Received 23 October 2014
Received in revised form
31 December 2014
Accepted 4 January 2015
Available online 2 December 2015

Keywords:
Fuzzy clustering
Fuzzy c-means clustering
Kernel-based fuzzy c-means
Genetic algorithm

ABSTRACT

Fuzzy c-means clustering algorithm (FCM) is a method that is frequently used in pattern recognition. It has the advantage of giving good modeling results in many cases, although, it is not capable of specifying the number of clusters by itself. Aimed at the problems existed in the FCM clustering algorithm, a kernel-based fuzzy c-means (KFCM) is clustering algorithm is proposed to optimize fuzzy c-means clustering, based on the Genetic Algorithm (GA) optimization which is combined of the improved genetic algorithm and the kernel technique (GAKFCM). In this algorithm, the improved adaptive genetic algorithm is used to optimize the initial clustering center firstly, and then the KFCM algorithm is availed to guide the categorization, so as to improve the clustering performance of the FCM algorithm. In the paper, Matlab is used to realize the simulation, and the performance of FCM algorithm, KFCM algorithm and GAKFCM algorithm is testified by test datasets. The results proved that the GAKFCM algorithm proposed overcomes FCM's defects efficiently and improves the clustering performance greatly.

1. Introduction

Traditional pattern recognition involves two tasks: unsupervised clustering and supervised classification [1,2]. In unsupervised clustering, samples without class labels are grouped into meaningful clusters. These clusters can be utilized to describe the underlying structure in data, which is helpful for better understanding of data. In supervised classification, samples with class labels are used to build the classification mechanism, through which class labels can be provided for new samples.

When class information is available, most traditional classifiers are designed in a direct way by employing supervised information to determine their decision functions. Such classifiers usually provide only class labels for new samples, but rarely care about the revelation of data distribution. For example, multilayered perception (MLP) [3] and support vector machines (SVM) [4,5] successfully utilize the class information of samples to achieve high classification accuracies; however, they emphasize more the classification of the data than the revelation of the data distribution, thus fail to interpret the obtained classification results well.

In contrast to these classifiers, another type of classifiers is designed in an indirect way by incorporating structural information into their classification schemes. Since clustering analysis is appropriate for exploring the data distribution [1,2], these classifiers usually first perform clustering to uncover the underlying structure in data, and then design classification rules based on the obtained structural information. In this way, these classifiers fuse the advantages of both clustering learning and classification learning together to some extent. On the other hand, clustering methods can be roughly categorized into unsupervised ones and supervised ones, depending on whether using class labels or not.

1.1. Clustering algorithm background

Radial basis function neural network (RBFNN) [6] is a classical algorithm belonging to the first category, i.e., unsupervised-clustering plus classifier-design. To determine the parameters of the hidden layer in RBFNN, training samples are clustered in an unsupervised way by using c-means or FCM [7]. Then, the connection weights between the hidden and output layers are optimized by minimizing the mean squared error (MSE) criterion between the target and actual outputs. Here, clustering makes RBFNN yield good generalization [3], but its function is just to help determine the parameters of the neural network, rather than explore the underlying structure of the input space. In fact, RBFNN cannot really inherit the merits of both clustering learning and classification learning as shown below.

Recently, some fuzzy relation based methods are proposed to bridge clustering and classification [8,9], which also belong to the first category. Setnes et al. proposed relational classifier trained by fuzzy clustering (FRC) to represent a transparent alternative to

* Corresponding author.
E-mail addresses: teacher.dingyi@live.com (Y. Ding), teacher.fu@live.com (X. Fu).

http://dx.doi.org/10.1016/j.neucom.2015.01.106
0925-2312/

conventional black-box techniques such as neural networks. To enhance FRC's robustness by replacing FCM and hard class labels with KFCM [10,11] and soft labels, respectively. The training of both algorithms includes two steps. First, unsupervised clustering is performed on training samples to discover the natural structure in data. Then, a relation matrix R between the obtained clusters and given class labels is established to reflect the logical relationship between clusters and classes. Here this matrix R plays a role of a connection weight matrix as in RBFNN. However, such relationship in both FRC and RFRC is directly constructed by the logical composite operator rather than by optimizing some defined criterion function. As a result, the clustering and classification results cannot be simultaneously optimal. In addition, the entries in the relation matrix R lack the statistical characteristic, and thus fail to indicate the relative reliability of the obtained relationship. Moreover, it is difficult to optimize these entries by defining an objective function due to in the differentiability of the composite operators.

It is worth noting that all the above algorithms have a common point: sequentially optimizing the clustering and classification objective functions respectively. That is, the clustering learning obtains a description of the underlying data distribution, and then the classification learning uses the obtained information to train the classification rules. In these algorithms, although the clustering learning and classification learning separately optimize their own criteria, such kind of sequential learning manner cannot always guarantee simultaneous optimality for both clustering and classification learning. In fact, the clustering learning here just aids the classification learning and does not benefit from the classification learning.

1.2. Genetic algorithm background

Over the last decade, GA have been extensively used as search and optimization tools in various problem domains, including sciences, commerce, and engineering. The primary reasons for their success are their broad applicability, ease of use, and global perspective [12].

The concept of a genetic algorithm was first conceived by John Holland of the University of Michigan, Ann Arbor. Thereafter, he and his students have contributed much to the development of the field. Most of the initial research works can be found in several conference proceedings. However, now there exist several text books on GAs [12–16]. A more comprehensive description of GAs along with other evolutionary algorithms can be found in the recently compiled "Handbook on Evolutionary Computation" published by Oxford University Press [17]. Two journals entitled "Evolutionary Computation" published by MIT Press and IEEE are now dedicated to promote the research in the field. Besides, most GAs applications can also be found in domain-specific journals.

Meng et al. [18] studied the encoding techniques of GA since GA encoding has significant influence on GA systems performance when solving problems with high complexity. A sufficient convergence condition on genetic encoding in Genetic Algorithms has been presented, such as Bias Code, Uniform Code, Trisector Code and Symmetric Code. Angelov [19] proposed a new approach for on-line design of fuzzy controllers of Takagi–Sugeno type (TS type); fuzzy rules are generated based on data collected during the process of control using newly introduced technique for on-line identification of TS type fuzzy models. Output of the plant under control and the respective control signal has been memorized and stored in on-line mode, and used to train in a noniterative, recursive way the fuzzy controller. Gacognne [20] has used the GA to find a set of nondominated solutions in the sense of Pareto instead of a unique solution with a unique fitness function. Gaccognne first began with a small random population of points in the space of research and setting a maximal size, then he used a family of genetic operators in relation with each specific problem, and he made a control on that family to give reinforcement for the best of them. Magdalena and Monasterio [21] proposed a new way to apply GAs to fuzzy logic controllers (FLC), and applies it to a FLC designed to control the Synthesis of biped walk of a simulated. A new approach adapted to systems with a larger number of variables has been proposed and tested over a FLC controlling a complex problem the locomotion of a simulated six links biped robot. Lee and Takagi [22] proposed a method for automatically designing complete triangular fuzzy systems using a genetic algorithm and a penalty strategy to determine membership function shape and position, number of fuzzy rules, and consequent parameters. Experimental results demonstrated the practicality of the method comparably to a system produced by another method, in Lee and Takagi work they have used triangular and trapezoidal membership functions for the fuzzy controller, and experimental score function. Several papers have proposed automatic design methods. Much of the work has focused on tuning membership functions [23–25] Takagi and Hayashi [26] used neural networks as a membership values generator and in [27] they treated fuzzy systems as networks and used back propagation techniques to adjust membership functions. Alata and Demirli [28] investigated the influence of the shape, the distribution of the membership functions and the order of the functional consequent of Takagi–Sugeno controller on the interpolation function of the fuzzy system. Number of inputs, conjunction operator, the order of consequent, and complementary or non-complementary triangular membership functions will determine the shape of the output.

2. Clustering algorithm

FCM algorithm is a clustering algorithm based on partitioning, which makes the idea to be divided into the biggest similarity between objects on the same cluster, while the minimum similarity between different clusters. Since the introduction of the fuzzy set theory in 1965 by Zadeh, it has been applied in a variety of fields. FCM is an improvement of common c-means algorithm for data classification that is rigid, while the FCM is a flexible fuzzy partition.

2.1. Fuzzy clustering algorithm

The fuzzy clusters are generated by the partition of training samples in accordance with the membership functions matrix $\mathbf{U} = [\mu_{ki}]$. ν_i is the degree of membership of x_k in the cluster i, x_k is the kth of d-dimensional measured data. The standard FCM uses the Euclidean distance as a cost function to be minimized and expressed as the following equation:

$$J_{FCM}(U,V) = \sum_{k=1}^{c} \sum_{i=1}^{n} \mu_{ki}^{m} |x_k - \nu_i|^2 \tag{1}$$

where $|\cdot|$ is any norm expressing the similarity between any measured data and the center.

As the FCM objective function is minimized, each pixel is assigned a high membership in a class whose center is close to the intensity of the pixel. A low membership is given when the pixel intensity is far from the class centroid. The FCM is minimized when the first derivatives of Eq. (1) with respect to μ_{ki} and ν_i are zero. The final classes and their centers are computed iteratively through these two necessary conditions. In the end, a hard classification is reached by assigning each pixel solely to the class with the highest membership value.

The FCM algorithm can be summarized as follows.

Algorithm 1. Fuzzy c-means algorithm.

1. Begin
2. Randomly initialize class centers ν_i $(2 \le i \le n)$ and fuzzy c-partition $\mathbf{U}^{(0)}$. Give fuzzification parameter $m(1 \le m \le \infty)$, constant ζ, and the value $\varepsilon > 0$;
3. Calculate the membership matrix $\mathbf{U} = [\mu_{ki}]$ using Eq. (2):

$$\mu_{ki} = \frac{\sum_{i=1}^{n} \left(\frac{1}{d_{ki}^2} \right)^{1/(m-1)}}{d_{ki}^2} \tag{2}$$

where d_{ki} is the Euclidean distance between the training pattern x_k and the class center ν_i;

4. Update the class centers use Eq. (3):

$$\nu_i = \sum_{k=1}^{c} \frac{x_k \sum_{i=1}^{n} \mu_{ki}^m}{\mu_{ki}^m} \tag{3}$$

5. Compute $\vartriangle = \max\left(|\mathbf{U}^{(t+)} - \mathbf{U}^{(t)}| \right)$. If $\vartriangle > \varepsilon$, then go to Step 2; otherwise go to Step 5;
6. Find the results for the final class centers;
7. The end.

2.2. Kernel-based fuzzy clustering algorithm

Define a nonlinear map as $\Phi : x \to \Phi(x) \in F$, where $x \in X$. X denotes the data space, and F the transformed feature space with higher or even infinite dimension. KFCM minimizes the following objective function:

$$J_{KFCM}(U, V) = \sum_{k=1}^{c} \sum_{i=1}^{n} \mu_{ki}^m \left| \Phi(x_k) - \Phi(\nu_i) \right|^2 \tag{4}$$

where

$$\left| \Phi(x_k) - \Phi(\nu_i) \right|^2 = K(x_k, x_k) + K(\nu_i, \nu_i) - 2K(x_k, \nu_i) \tag{5}$$

where $k(x, \nu) = \Phi(x)^T \Phi(x)$ and is an inner product kernel function. If we adopt the Gaussian function as a kernel function, i.e., $K(x, \nu) = \exp\left(\frac{-|x-\nu|^2}{\tau^2} \right)$, then $k(x, x) = 1$, according to Eq. (5), Eq. (4) can be rewritten as

$$J_{KFCM}(U, V) = 2 \sum_{k=1}^{c} \sum_{i=1}^{n} \mu_{ki}^m (1 - K(x_k, \nu_i)) \tag{6}$$

Minimizing Eq. (6) under the constraint of U, we have

$$\mu_{ki} = \frac{\left(\frac{1}{1 - K(x_k, \nu_i)} \right)^{1/(m-1)}}{\sum_{j=1}^{c} \left(\frac{1}{1 - K(x_k, \nu_i)} \right)^{1/(m-1)}} \tag{7}$$

$$\nu_i = \frac{\sum_{k=1}^{n} \mu_{ki}^m K(x_k, \nu_i) x_k}{\sum_{k=1}^{n} \mu_{ki}^m K(x_k, \nu_i)} \tag{8}$$

It is worth to note that although Eqs. (7), (8) are derived using the Gaussian kernel function, we can use other functions satisfying $K(x, x) = 1$ in Eqs. (7), (8).

In real applications such as the following RBF functions and hyper tangent functions:

1. RBF functions:

$$K(x, \nu) = \exp\left(\frac{-\sum_{i=1}^{c} |x_i^a - \nu_i^a|^b}{\tau^2} \right) \tag{9}$$

2. Hyper tangent functions:

$$K(x, \nu) = 1 - \tanh\left(\frac{-|x-\nu|^2}{\tau^2} \right) \tag{10}$$

Note that RBF function with $a = 1, b = 2$ reduces into the common used Gaussian function. In fact, Eq. (5) can be viewed as kernel-induced new metric is in the data space, which is defined as the following:

$$d(x, \nu) \triangleq \left| \Phi(x) - \Phi(\nu) \right| = \sqrt{2(1 - K(x, \nu))} \tag{11}$$

According to Eq. (8), the data point is endowed with an additional weight $K(x_k, \nu_i)$, which measures the similarity between x_k and ν_i. When x_k is an outlier, i.e., x_k is far from the other data points, $K(x_k, \nu_i)$ will be very small, so the weighted sum of data points shall be more robust. Since in incomplete dataset, a data point with missing components is likely to turn into an outlier, the algorithm based on KFCM to cluster incomplete data is of great potential.

The full description of KFCM algorithm is as follows:

Algorithm 2. Kernel-based fuzzy c-means algorithm.

1. Begin
2. Fix c, t_{max}, $m > 1$, $\varepsilon > 0$, for some positive constant;
3. Initialize the memberships $\mu_{ki}{}^0$;
4. For $t = 1, 2, \ldots, t_{max}$ do: Update all prototypes $\nu_{ki}{}^t$ with Eq. (6); Update all memberships $\mu_{ki}{}^t$ with Eq. (5); compute $E^t = \max_{ki} |\mu_{ki}^t - \mu_{ki}^{t-1}|$, if $E^t \le \varepsilon$ stop; else $t = t+1$;
5. The end.

3. Genetic algorithm

GAs are global, parallel, stochastic search methods, founded on Darwinian evolutionary principles. During the last decade GAs have been applied in a variety of areas, with varying degrees of success within each. A significant contribution has been made in control systems engineering. GAs exhibit considerable robustness in problem domains that is not conducive to formal, rigorous, classical analysis. They are not limited by typical control problem attributes such as ill-behaved objective functions, the existence of constraints, and variations in the nature of control variables. The computational complexity of the GAs has proved to be the chief impediment to real-time application of the technique. Hence, the majority of applications that use GAs are, by nature, off-line.

Commonly GAs have been used to optimize both structure and parameter values for both controllers and plant models. They have also been applied to fault diagnosis, stability analysis, robot path-planning and combinatorial problems.

3.1. Genetic algorithm

The GA begins, like any other optimization algorithm, by defining the optimization variables, the cost function, and the cost. It ends like other optimization algorithms too, by testing for convergence. In between, however, this algorithm is quite different. A path through the components of the GA is shown as a flowchart in Fig. 1. Each block in this "big picture" overview is discussed in detail in this section.

This individual is just the plant in the under solved problem, it is also a point in the searching space. While a population consists of several individuals. Generally, it is also a subset of the whole searching space. Chromosome is the expression of the individual which is coded. The character in the string is called gene. There are three kinds of operation in GA. Selection reproduction, crossover,

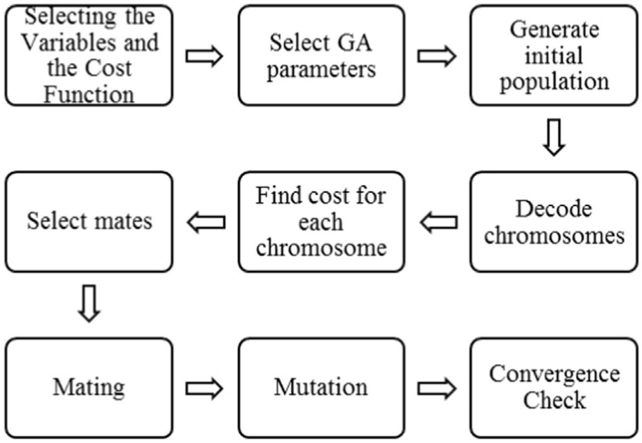


Fig. 1. Flowchart of GA.

as well as mutation. We usually do as follows: choose N chromosomes from population S in N separate times. The probability of one individual being chosen is $P(x_i)$. The computational formula of $P(x_i)=\frac{f(x_i)}{\sum_{j=1}^{n} f(x_j)}$. There is a chance that the chromosomes of the two parents are copied unmodified as offspring, or randomly recombined (crossover) to form offspring. There is a chance that a gene of a child is changed randomly. Generally the chance of mutation is low.

3.2. GAKFCM algorithm

The FCM based on the objective function is widely applied because of its strong ability of local search and its fast convergence speed. However, FCM algorithm has two defects. First, to one sample, the sum of the membership degree for all categorizations, which makes it sensitive to the noise and isolated data. Second, FCM is essentially a kind of local hill-climbing algorithm, which makes it sensitive to the initial clustering center and easy to converge to a local extremum. Aimed at the problems existed in the FCM clustering algorithm, a kernel-based FCM is proposed to optimize FCM clustering, based on the GAKFCM which is combined of the improved genetic algorithm and the kernel technique. Firstly, an improved adaptive genetic algorithm is designed by the real coding mode, nonlinear ranking select measurement, adaptive crossover and mutation strategy. The criteria are the maximum evolution iteration and the average fitness convergence. Then, KFCM is presented by change the clustering distance of the FCM to define the objective function, and then to improve the constraint conditions of probability in FCM. Finally, an algorithm called as GAKFCM is proposed, which is combined the improved adaptive genetic algorithms presented in this thesis with the KFCM clustering algorithm. In this algorithm, the improved adaptive genetic algorithm is used to optimize the initial clustering center firstly, and then the KFCM algorithm is availed to guide the categorization, so as to improve the clustering performance of the FCM algorithm.

Fitness is the standard to judge the individual. We can use it to evaluate the individuals in order to estimate them. Fitness function is the relationship between an individual and its fitness. It is an evaluation function in GA:

$$f(U,V)=\frac{1}{1+J_{KFCM}(U,V)}=\frac{1}{1+\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ki}^{m}\left|\Phi(x_k)-\Phi(\nu_i)\right|^2} =\frac{1}{1+\sum_{k=1}^{c}\sum_{i=1}^{n}\mu_{ki}^{m}(2-2K(x_k,\nu_i))} \tag{12}$$

where $K(x_k,\nu_i)=\exp\left(\frac{-|x_k-\nu_i|^2}{\tau^2}\right)$.

The full description of GAKFCM algorithm is as follows:

Algorithm 3. GAKFCM algorithm.

1. Begin
2. Initialize: Set the parameters of KFCM algorithm, c,t_{max}, $m>1$, $\varepsilon>0$, for some positive constant;
3. Set the parameters of GAKFCM algorithm, crossover rate p_{c0}, mutation rate p_{m0}, iterations T, initial population $p(t)=\{V_k^{(t)}\,|\,k=0,1,\dots,n\}$;
4. Compute $J_{KFCM}(U,V)$ of each individual in the population, and the fitness of each individual $f(U,V)$, $f(j)$, $f_c(t)$, $f_m(t)$;
5. Compute the t population $p(t)$, selection-reproduction, crossover p_{ct}, as well as mutation p_{mt} of each individual in the $t+1$ population $p(t+1)$;
6. Compute the fitness $\overline{f(U,V)}=\frac{1}{n}\sum_{k=1}^{n} f(k)$, if $|\overline{f(t)}-\overline{f(t-1)}|>\delta$, else $t=t+1$, goto step 3;
7. Select the best individual of the last generation as the algorithm's final results;
8. The end.

4. Experimental results

In this section, we show several examples to illustrate the ideas presented in the previous sections. A complete program using MATLAB programming language was developed to find the optimal value of the result about FCM algorithm, KFCM algorithm and GAKFCM algorithm. It starts by performing three clustering algorithms for input–output data, build the fuzzy model using three clustering algorithms and optimize the parameters by optimizing the least square error between the output of the fuzzy model and the output from the original function by entering a tested data. The optimizing is carried out by iteration. After that, the genetic algorithms optimized the weighting exponent of FCM, KFCM and GAKFCM. The same way build the fuzzy model using FCM, KFCM and GAKFCM then optimize the weighting exponent m by optimizing the least square error between the output of the fuzzy model and the output from the original function by entering the same tested data.

In the first experiment, we use the test data from the IRIS database and WINE database, all experiments are averaged over 20 runs with $n=50, T=100, p_{c0}=0.6, p_{m0}=0.001$ and $\sigma=150, k=20$, $c=3, m=2, \varepsilon=0.0001$. Results are shown in Figs. 2 and 3.

The next experience, we use the ANC database which downloads from http://www.anc.org. The documents are organized into broader categories: World, Politics, Tech, Science, It and Health. Among these documents, 3000 of them are training dataset and the others are testing dataset. Just as Table 1, because of the large

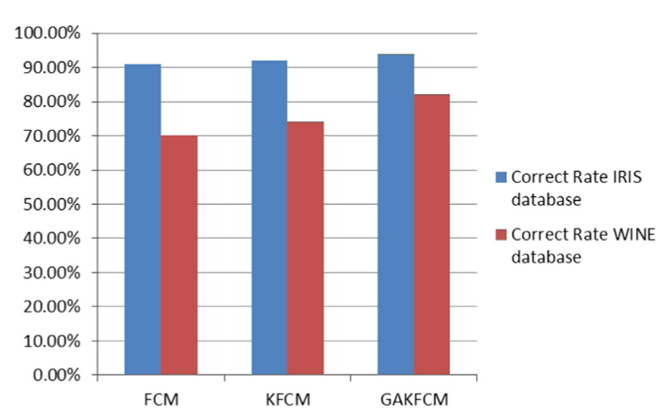


Fig. 2. Correct rate of three algorithms use the IRIS database and WINE database.

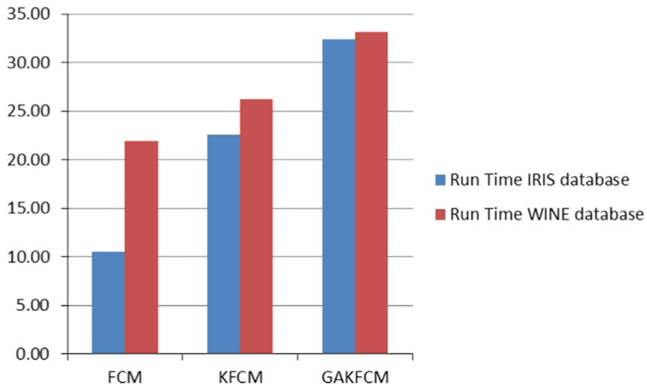


Fig. 3. Run time of three algorithms use the IRIS database and WINE database.

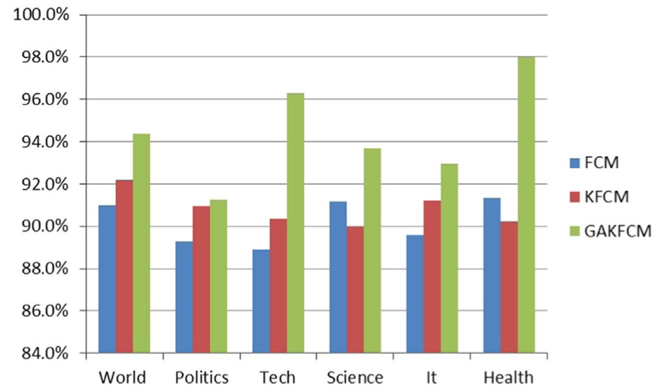


Fig. 4. The precision of three algorithms.

Table 1
The distribution of training and the testing dataset.

Sort	World	Politics	Tech	Science	It	Health
Training numbers	500	500	500	500	500	500
Testing numbers	400	400	400	400	400	400

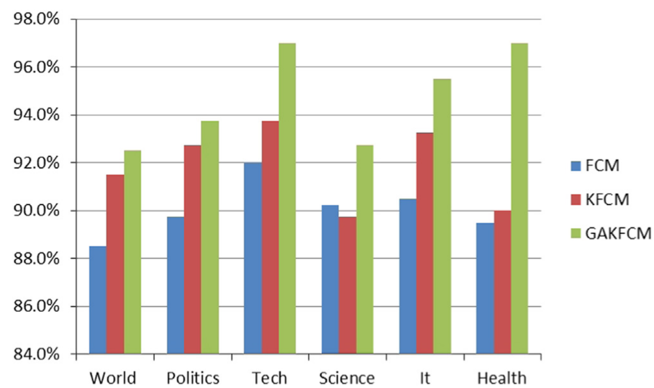


Fig. 5. The recall of three algorithms.

Table 2
The result based on three algorithms.

Sort	World	Politics	Tech	Science	It	Health
FCM retrieved	389	402	414	396	404	392
FCM relevant	354	359	368	361	362	358
KFCM retrieved	397	408	415	399	409	399
KFCM relevant	366	371	375	359	373	360
GAKFCM retrieved	392	411	403	396	411	396
GAKFCM relevant	370	375	388	371	382	388

number of classes and also uneven number of samples in each class, it is a difficult clustering problem.

And the following Table 2 and Figs. 4, 5 are based on FCM, KFCM and GAKFCM algorithm respectively. We wanted to compare our algorithm against existing implementation of clustering algorithms, especially GAKFCM algorithm. Let the sort and the numbers of every time be the same. Because the same distribution is used for cluster centers throughout, the expected distances between cluster centers will be easily to understand. Thus the expected value of the cluster isolation parameter varies inversely with the standard deviation. The initial centers were chosen by taking a random sample of data points.

For each standard deviation we run each of Algorithms 3 times and then we got the average value. For all runs the same data points were used. For each of the three runs a new set of initial center points was generated, and both the algorithms were run using the same dataset and initial center points. The algorithms run for a maximum of 30 stages or until convergence. The reason that we felt it safe to limit the number of stages is that the running times tend to remain constant after the first few stages.

In the experiment, each document was assigned to one of two categories, "relevant" and "not relevant". In this case, the "relevant" is the list of all documents on the test database that are relevant for a certain topic. The "retrieved" is the list of documents produced by search for a query. Precision is the probability that a retrieved document is relevant. Precision is defined as the number of relevant documents retrieved by a search divided by the total number of documents retrieved by that search in Eq. (13). Recall is the probability that a relevant document is retrieved in a search, also called sensitivity. Recall is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents in Eq. (14). View from the result of the experiments shows that the GAKFCM can get significantly precise categorization than the FCM and KFCM algorithm. By using of GAKFCM algorithm, the precision and the recall are both significantly improved:

Table 3
The four outcomes contingency table by classification.

Total	Condition positive	Condition negative
Test outcome positive	True positive	False positive
Test outcome negative	False negative	True negative

$$precision = \frac{|relevant \cap retrieved|}{|retrieved|} \tag{13}$$

$$recall = \frac{|relevant \cap retrieved|}{|Testing|} \tag{14}$$

For classification tasks, the terms true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) compare the results of the classifier under test with trusted external judgments. The terms positive and negative refer to the classifier's prediction (sometimes known as the expectation), and the terms true and false refer to whether that prediction corresponds to the external judgment. The four outcomes can be formulated in a 22 contingency table or confusion matrix, as follows Table 3: The precision and recall are defined in the following equation:

$$precision = \frac{TP}{TP + FP} \tag{15}$$

$$recall = \frac{TP}{TP+FN} \tag{16}$$

View the experiments; we can see that the GAKFCM algorithm is super than algorithm of tradition FCM and KFCM whatever the speed of convergence or stability of the clustering. When it is applied to a bit more datasets, the superiority of GAKFCM algorithm is more obvious, and we can obtain the useful results more quickly.

5. Conclusions

In this paper, presented a GAKFCM clustering algorithm based on genetic algorithm. In this algorithm, the improved adaptive genetic algorithm is used to optimize the initial clustering center firstly, and then the KFCM algorithm is availed to guide the categorization, so as to improve the clustering performance of the FCM algorithm. We realize the simulation, and the performance of FCM algorithm, KFCM algorithm and GAKFCM algorithm is testified by test datasets. The results proved that the GAKFCM algorithm proposed overcomes FCM's defects efficiently and improves the clustering performance greatly. The GAKFCM is more accurate, and can overcome premature for the GA algorithm, which can perform quite well when the user has some knowledge about the non-linearities in the problem. And study the GA algorithm from a different perspective.

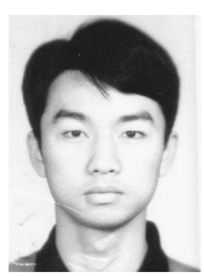
Acknowledgment

Yi Ding's work is supported by the S&T plan projects of Hubei Provincial Education Department of China (No.Q20152501).

References

[1] J.M.A.K. Jain, R.P.W. Duin, Statistical pattern recognition: a review, IEEE Trans. Pattern Anal. 22 (2000) 4–37.
[2] P.J.F.A.K. Jain, M.N. Murty, Data clustering: a review, ACM Comput. Surv. 31 (1999) 264–323.
[3] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, New Jersey, 1999.
[4] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1999.
[5] C. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discov. 2 (1998) 121–167.
[6] K.C.M.T. Musavi, W. Ahmed, On the training of radial basis function classifiers, Neural Netw. 5 (1992) 595–603.
[7] J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum, New York, 1981.
[8] R.B.M. Setnes, Fuzzy relational classifier trained by fuzzy clustering, IEEE Trans. Syst. Man Cy. B 29 (1999) 619–625.
[9] D.Z.W.L. Cai, S.C. Chen, Robust fuzzy relational classifier incorporating the soft class labels, Pattern Recognit. Lett. 28 (2007) 2250–2263.
[10] S.C.D.Q. Zhang, A novel kernelized fuzzy c-means algorithm with application in medical image segmentation, Artif. Intell. Med. 32 (2004) 37–50.
[11] D.Z.S.C. Chen, Robust image segmentation using fcm with spatial constraints based on new kernel-induced distance measure, IEEE Trans. Syst. Man Cy. B 34 (2004) 1907–1916.
[12] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, New York, 2012.
[13] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.
[14] Z. Michalewicz, Genetic Algorithms Data Structures Evolution Programs, Springer-Verlag, Berlin, 2012.
[15] M. Mitchell, Introduction to Genetic Algorithms, MIT Press, Ann Arbor, 1996.
[16] R.C.M. Gen, Genetic Algorithms and Engineering Design, Wiley, New York, 1997.
[17] Z.M.T. Black, D. Fogel, Handbook of Evolutionary Computation, Institute of Physics Publishing, Oxford University Press, New York, 1997.
[18] Z.C.Q.C. Meng, T.J. Feng, Genetic algorithms encoding study and a sufficient convergence condition of gas systems, in: IEEE SMC'99 Conference Proceedings, vol. 1, 1999, pp. 649–652.
[19] A. Plamen, A fuzzy controller with evolving structure, Inf. Sci. 161 (2013) 21–35.
[20] L. Gacognne, Research of Pareto Set by Genetic Algorithm, Application To Multicriteria Optimization of Fuzzy Controller, Institut Informatique Entreprise, Evry France, 2011.
[21] F.L. Magdalena, Evolutionary-based learning applied to fuzzy controllers, in: International Joint Conference of the 14th IEEE International Conference, 1995, vol. 3, pp. 1111–1118.
[22] M.A. Lee, Integrating design stage of fuzzy systems using genetic algorithms, in: 12th IEEE International Conference, 1993, vol. 1, pp. 612–617.
[23] R.P.S.K. Sinha, R.N. Patel, Application of ga and pso tuned fuzzy controller for agc of three area thermal-thermal-hydro power system, Int. J. Comput. Theory Eng. 2 (2010) 238–244.
[24] R.Jang, Fuzzy controller design without domain experts, in: IEEE International Conference on Fuzzy Systems, 1992, vol. 1, pp. 289–296.
[25] R. Jang, Self-learning fuzzy controllers based on temporal back propagation, IEEE Trans. Neural Netw. 5 (1992) 714–723.
[26] H. Takagi, I. Hayashi, NN-driven fuzzy reasoning, Int. J. Approx. Reason. 1991, 3 (5) 191–212.
[27] N.H. Nomura, I. Hayashi, A self-tuning method of fuzzy control by descent method, in: 21th IFSA Congress, 1991, pp. 155–158.
[28] A.M. Alata, K. Demirli, Interpolation behavior of ts fuzzy controllers, in: International Symposium on Intelligent Control/Intelligent Systems and Semiotics Cambridge, 1999, pp. 359–364.

Yi Ding received the Ph.D. degree in School of Computer Science and Technology from Huazhong University of Science and Technology, Wuhan, China, in 2013. She is currently an Associate Professor with the College of Computer Science and Technology, Hubei Normal University, Huangshi, China. Her current research interests and experience include fuzzy clustering, neural networks, pattern recognition.



Xian Fu received the bachelor degree in the College of Computer Science and Technology, Hubei Normal University, Huangshi, China, in 2005. He is currently a Lecturer with the college of Computer Science and Technology, Hubei Normal University, Huangshi, China. His current research interests and experience include neural networks, pattern recognition, network, text mining, automatic ontology acquisition.