An Improved PSO Clustering Algorithm with Entropy-based Fuzzy Clustering

YUYAN ZHENG[1], YANG ZHOU[2], JIANHUA QU[1]
[1]School of Management Science and Engineering
Shandong Normal University
Jinan, Shandong
CHINA
[2]College of Information Engineering
Northwest A&F University
Yangling, Shannxi
CHINA
zyy0716_source@163.com, zhouyang_online@163.com, qujh1978@163.com

Abstract: - Particle swarm optimization is a based-population heuristic global optimization technology and is referred to as a swarm-intelligence technique. In general, each particle is initialized randomly which increases the iteration time and makes the result unstable. In this paper an improved clustering algorithm combined with entropy-based fuzzy clustering (EFC) is presented. Firstly EFC algorithm gets an initial cluster center. Then the cluster center is regarded as inputs of one of all particles instead of being initialized randomly. Finally we cluster with the improved clustering algorithm which guarantees unique clustering. The experimental results show that the improved clustering algorithm has not only high accuracy but also certain stability.

Key-Words: - Particle Swarm Optimization (PSO); Entropy-based Fuzzy Clustering; Cluster Center Initialization

1 Introduction

As an important method in the field of data mining, Clustering is the process of partitioning dataset with *n* data points into many sub-sets. Each sub-set represents one cluster and the data points in the same cluster have high similarity in comparison to one another, but are dissimilar to data points in other clusters [1]. It has been applied in many fields such as pattern recognition, knowledge discovery, machine learning, statistics and so on [2].

In current, there have been varieties of clustering algorithms. Partitional algorithm is one of most common methods. In addition, hierarchy-based, density-based, model-based and grid-based are also popular clustering methods. After that, many intelligence algorithms gradually developed. Particle swarm optimization (PSO) is a kind of classical swarm intelligence algorithm. And it is inspired by the natural phenomena like bird flocking or fish schooling [3], which has been an interesting area of study in artificial life. It has less parameters to adjust. By following the personal and global best value, each particle is constantly updated in the search space. After the fixed iteration number, the optimal result will be obtained.

There have been a number of improved PSO algorithms So far. Among them, the number of clusters is a study direction. Paper [4] presents a kind of dynamic clustering technique so as to find the best number of clusters automatically. And Yucheng and Szu-Yuan [5] propose a clustering approach with variable number of clusters, and they use K-means and CPSO [4]. Another direction of study is the research on cluster centers. Paper [6] obtains the initial cluster center by using K-means algorithm. The cluster center is regarded as input of one of all particles. This strategy improves the performance of PSO algorithm for clustering. In addition, there are also other directions of study. A comprehensive review of PSO algorithm and their applications in clustering can be found in paper [3].

Entropy-based fuzzy clustering algorithm (EFC) identifies the number of clusters and initial cluster centers by itself. It selects the data point with minimum entropy as cluster center. And it just requires two parameters which are easy to be specified.

In current, there have been many literatures about EFC algorithm. Paper [7] uses the average information entropy to give the cluster number. The corresponding cluster number is suitable when entropy is the least. Paper [8] gets cluster number and cluster center and then the result is regarded as initial value of K-means algorithm. The performance of algorithm has been improved on the terms of accuracy rate and running time. Paper [9] presents a kind of algorithm which gets cluster center with EFC algorithm and then improves them with FCM algorithm. It can get tight and different clustering result. Literature [10] introduces entropy into corporation network so as to identify particular network. Literature [11] introduces entropy into nervous network to propose an improved nervous network algorithm.

This paper proposes an improved PSO algorithm which combines two algorithms above. At first, a cluster center is obtained through EFC algorithm. Then during the process of initialization of the particles, it is regarded as input of one of the particles. In the end, the clustering results are showed with the improved PSO clustering algorithm, and at the same time experiment analysis is introduced.

The remainder is organized as follows. Section 2 describes the basic PSO algorithm and its application in clustering. Section 3 introduces the EFC algorithm briefly. In section 4 we show fitness function, particle encoding, inertia weight and the combination with EFC algorithm. Experimental results of two datasets and discussion are showed in section 5. The paper concludes in section 6 with discussion of future work.

2 The Description of Particle Swarm Optimization

2.1 The Standard Particle Swarm Optimization Algorithm

The original PSO algorithm was developed by Kennedy and Eberhart in 1995[12][13] which is inspired by the social behavior of bird flocking and fish schooling. Each particle presents a candidate solution to the problem, all particles are constantly updated following the personal and global optimal positions in the space until the maximum iteration number is satisfied. The performance of each particle is measured with a fitness function.

In basic PSO algorithm, each particle updates its velocity and position according to the formulas of velocity and position. In general, each particle tries to follow the personal and global best particle found by now and obtains the optimal solution after the iteration. The formulas of velocity and position are as followed.

$$v_{id}(t+1)=w*v_{id}(t)+c_1*r_1*(p_{id}(t)-x_{id}(t))+c_2*r_2*(p_{gd}(t)-x_{id}(t)) \quad (1)$$

$$x_{id}(t+1)=x_{id}(t)+v_{id}(t+1) \quad (2)$$

Where:

1. w is the inertia weight which usually linearly decreases during the iteration. It plays an important role in balancing the local and global search. The particle will carry out global search when it is larger; otherwise, the particle local space carefully;
2. c_1, c_2 are cognition factors and in the rang [0,2];
3. $r_1, r_2 \in U(0,1)$;
4. v_{id}, x_{id} are the velocity and position of the ith particle in the dth dimension, respectively;
5. p_{id}, p_{gd} are the best positions found thus far by the ith particle and all the particles, respectively; that is, the personal best position and global best position. The computational formula is Eq.(3):

$$p_{id}(t+1)=\begin{cases} p_{id}(t) & if\ f(x_{id}(t+1)) \geq f(p_{id}(t)) \\ x_{id}(t+1) & if\ f(x_{id}(t+1)) < f(p_{id}(t)) \end{cases} \quad (3)$$

6. t is the iteration number.

The steps of the basic PSO algorithm are as follows:

1. Parameter: decide the values of all parameters.
2. Initialization: the velocity and position of each particle are initialized randomly.
3. Fitness function: compute the fitness value according to the fitness function.
4. Comparation1: obtain personal best position by comparing with the previous personal best fitness value.
5. Comparation2: obtain global best position by comparing with the previous global best fitness value.
6. Update v: update the velocity of particle according to the Eq. (1).
7. Update p: update the position of particle according to the Eq. (2).
8. Repeat steps 2 to 7 until the termination condition (the maximum number of iteration) is satisfied.

2.2 Particle Swarm Optimization for Data Clustering

For the optimization problems, PSO has been proved to be both effective and fast since it was developed in 1995. Because of the promising performance on nonlinear function optimization, it has received much attention [14]. The clustering problem can be regarded as a special optimal problem so that it can also be solved by PSO algorithm. Its parameters are few and the convergence speed is fast when the dimension is low. What's more, it is easy to operate.

The first one of all researchers is Merwe. He has proposed two kinds of improved PSO clustering algorithms based on K-means, which used K-means at the beginning and in the end, respectively. After that, many experts carry on the study about the fitness function, particle encoding, initialization and so on.

Like many other clustering algorithms, the PSO clustering algorithm is aimed to minimize intra-cluster distances as well as maximize inter-cluster distances. Each particle represents a cluster center. Each data point is classified into different cluster from which the distance is the minimum. According to the updating formulas, the velocity, the position and fitness function are updated iteratively. Then data point is classified again until the maximum number of iteration is satisfied.

The way of particle encoding has several kinds of different styles in PSO clustering algorithm. The way which is frequently used is based on cluster centers. A particle represents *k* cluster centers. Assuming that dataset D is partitioned into *k* different clusters, then each particle C_i represents *k* cluster centers and is encoded as follows:

$$C_i = (C_{i1}, C_{i2}, ..., C_{ij}, ..., C_{ik}) \quad (4)$$

Where, C_{ij} is the *j*th clustering center of the *i*th particle.

3 Entropy-based Fuzzy Clustering Algorithm

Entropy-based fuzzy clustering algorithm (EFC) was developed by J.Yao and the other experts in 2000 [15]. It identifies the number of clusters and initial cluster centers by itself. The entropy of each data point is based on similarity and is related to the Euclidean distance. The similarity between two data points is normalized to [0.0-1.0]. The data point with the least entropy value is selected as cluster center. The total entropy value of a data point is calculated as under:

$$E_i = -\sum_{j \in x}^{j \neq i} (S_{ij} \log_2 S_{ij} + (1 - S_{ij}) \log_2 (1 - S_{ij})) \quad (5)$$

Where $S_{ij} = e^{-\alpha d_{ij}}$ is the similarity between two data points *i,j* and normalized to [0.0-1.0], d_{ij} is the Euclidean distance between points i and j. The constant $\alpha = \ln(0.5 / \overline{D})$ and $\overline{D}$ is the mean distance among the pairs of data points in a hyper-space and is usually set to 0.5.

We evaluate entropy of every data point and select the data point with minimum value as the first cluster center. Then the data points whose similarity with cluster center is greater than the threshold value β are removed from dataset. Meanwhile cluster center is also removed from dataset. The threshold value β is viewed as a threshold of similarity among the data points in the same cluster. It takes a value in the range [0.0-1.0] and has an important effect on the performance of EFC algorithm. The β with 0.7 has good result and is quite robust [15].The data points whose similarity with center is greater than β can not be selected as cluster center. In the next iteration, we select the least entropy value from the remaining data points as the next cluster center. Similarly, relevant data points are removed from dataset. The process is repeated until no data point is left.

In addition, the outliers may have the least entropy value and be selected as cluster center. In order to tackle the problem, we introduce a parameter γ to distinguish potential cluster centers and the outliers. If the number of data points that have similarity with the selected data point greater than β is less than γ , then this selected data point can not be viewed as center and should be regarded as outlier. γ is usually 5% of the total number of data points[15].

The steps of entropy-based fuzzy clustering algorithm are as follows [16]:

1. Compute entropy E_i for each data point *i* in dataset D, *i=1,2,…,n*.
2. Choose the minimum value of all entropy values and identify it as the cluster center.
3. Remove the data point with minimum value and those data points whose similarity is greater than β from dataset D.
4. If D is not empty then go to step 2.

4 The Presented PSO Clustering Algorithm

4.1 Inertia weight

As a useful parameter of PSO algorithm, inertia weight plays an important role in the performance of PSO. In general, a bigger inertia weight is beneficial to global search while the smaller one can improve the capability of local search. Varieties of inertia weights have been proposed by now. The selection of inertia weight has been studied by Shi and Eberhart(1998), and they come to a conclusion that the convergence speed of PSO is higher when w is in the range of [0.8,1.2]. Literature [17] gives a kind of self-adaptive inertia weight which is the way of index. Shi and Eberhart(2001) present a random inertia weight. In addition, there are other various types of inertia weights such as constant, trigonometric functions, logarithmic functions and so on.

This paper adopts the following linear differential decline inertia weight:

$$w(t) = w_{star} - \frac{(w_{star} - w_{end})}{t_{\max}^2} \times t^2 \qquad (6)$$

Where $w(t)$, w_{star} and w_{end} are the current inertia weight, the initial inertia weight and the final inertia weight, respectively. $t_{\max}$ and t are the current iteration number and the maximum iteration number, respectively.

4.2 Particle Encoding

There have been three kinds of primary encoding schemes by now. They are the binary encoding, integer encoding, and real encoding [18]. For binary encoding, the value of particle is the binary string of length N which is the number of data points. For integer encoding, each partition is an integer vector of length N which is also the number of data points, these integers represent the cluster labels of different data points. For real encoding, the position of particle is comprised of real numbers that represent cluster centers.

This paper adopts the real encoding in which a particle represents k cluster centers. Assumed that dataset D is partitioned into k clusters, the dimension of dataset is n, then the position of particle C_i represents k cluster centers and is encoded according to the Eq.(4) and its length is $n \times k$. The first n values denote the first cluster center. The next n values denote the second cluster center, and so forth.

For an example, all data points are divided into 3 clusters in the 2 dimension space, then the encoding of particle is $C_1 = (2,4,-5,3,7.5,15)$ and its length is $2 \times 3 = 6$. The first 2 values represent the first cluster center (2, 4), the next 2 values represent the second cluster center (-5, 3) and the final 2 values represent the third cluster center (7.5, 15).

4.3Fitness Function

So far, there have been many kinds of fitness functions described in the literature. The fitness function is related with the problem to be solved. Kennedy and Eberhart (1995) suggested a fitness value associated with each particle. For the assessment of partitions formed by only two clusters, Krovi [19] presents a kind of fitness function. Combined with the Davis-Bouldin (DB) index which commonly is the relative validity criteria for clustering, Bandyopadhyay introduces a fitness function in paper [20].

This paper adopts the following fitness function based on the sum of distance.

$$F = \sum_{i=1}^{k} \sum_{j=1}^{n_i} \| X_{ij} - C_i \|^2 \qquad (7)$$

Where, n_i is the number of data points which belong to the ith class, k is the number of all clusters. X_{ij} denotes the jth data in the ith class, C_i is the cluster center of the ith class.

4.4 The Combination with Entropy-based Fuzzy Clustering Algorithm

Combine with the entropy-based fuzzy clustering, we propose an improved PSO clustering algorithm (EFCPSO).As we all know, the entropy-based fuzzy clustering algorithm can identify the number of clusters and the cluster centers. The PSO algorithm needs the initial value of all particles. So we take advantage of the merits of two algorithms, the EFCPSO algorithm is presented.

The basic procedure is described in detail. After the constant α, the threshold value β and the parameter γ which are introduced in section 3 are identified, We first obtain a cluster center with the entropy-based fuzzy clustering. Then during the process of initialization, the obtained cluster center is regarded as the position value of one particle and the other particles are initialized randomly in the range of [0,1].Finally, according to the rule of the nearest neighbor, data points are classified into the corresponding cluster. The rule of the nearest neighbor means that the data point $X_m (m = 1,2,...,n)$ is divided into the cluster $C_{ij} (j = 1,2,...,k)$ from which the distance of the data point is shorter than from the other cluster center. The distance between data point and cluster center satisfies Eq.(8). After that, the fitness value is calculated using Eq.(7) and the personal and global best positions is found which are used to update the velocity and position of the particle in Eqs.(1) and (2). The above steps are repeated until the final number of iteration is satisfied.

$$\| X_m - C_{ij} \| = \min\{\| X_m - C_{i1} \|,...,\| X_m - C_{ik} \|\} \quad (8)$$

Where, C_{ij} is the jth clustering center of the ith particle, || || means the distance between one point and another one.

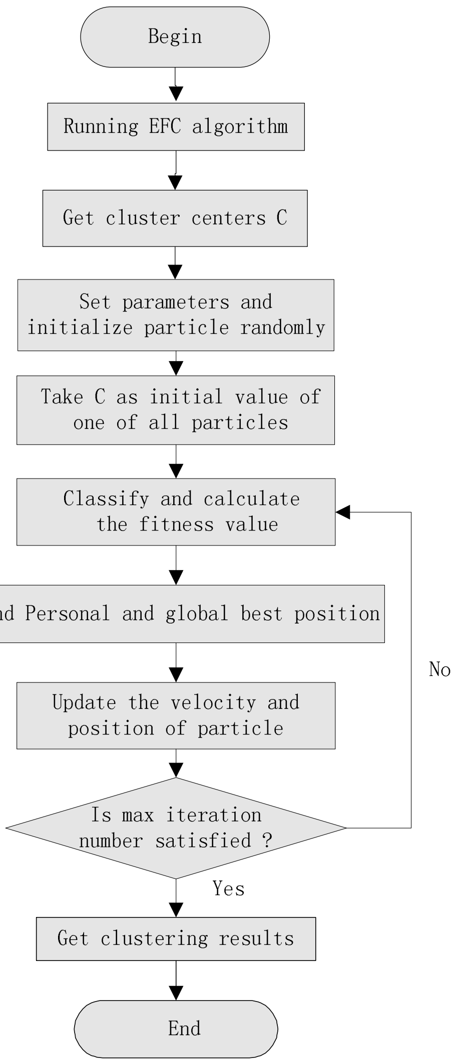

Fig.1 Flow chart showing basic steps of EFCPSO algorithm

The steps of EFCPSO clustering algorithm are as follows:

1. Get a cluster center C with EFC algorithm.
2. Set all parameters of PSO algorithm and initialize the velocity and position randomly.
3. Take C as the position value of one of the particles.
4. Classify and calculate the fitness value of each particle using Eq. (7).
5. Compare the fitness value of each particle and get personal best position.
6. Compare the fitness value of all particles and get global best position.
7. Update the velocity of particle using Eq. (1).
8. Update the position of particle using Eq. (3).
9. Repeat steps 4 to 8 until the maximum number of iteration is satisfied.

5 Experiments and Discussion

5.1 The datasets

Iris and Wine datasets are used to conduct relevant experiments in this section. The UC Irvine machine learning repository [19] gives the detail description of datasets which have often been used to test the performance of different clustering algorithms.

Iris dataset has three kinds of Iris flowers whose names are Iris setosa(I), Iris versicolor(II) and Iris virginica(III),respectively. Each class has fifty objects, every object is described by four attributes, viz sepal length, sepal width, petal length and petal width [1], thus a total of 150 objects are available.

Wine recognition dataset is the result of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. Every sample contains thirteen attributes and the total number of samples is 178.The samples in class I, class II and class III are 59, 71 and 48, respectively. The classes are separable.

The description of two datasets is showed in table 1.

Table 1 Summary of dataset

Data	No. of class	No. of data	No. of I II III	No. of attribute
Iris	3	150	50/50/50	4
Wine	3	178	59/71/48	13

5.2 Results and discussion

This section gives relevant performance of different clustering algorithms. The performance of PSO algorithm is better by initializing cluster center using EFC algorithm. Relevant experiments of K-means, the integration of K-means and PSO (KPSO), PSO and EFCPSO algorithms have been conducted on Iris and Wine datasets 10 times. We have adopted the accuracy rate (AR) to measure the performance of several algorithms in Eq.(9).

$$AR = \frac{n}{N} \times 100\% \quad (9)$$

Where, n and N are the number of correctly classified data points and total data points, respectively.

The experimental results have been summarized in tables 2[20], 3[20], 4[20], 5 and 6.

Table 2 PSO clustering on Iris dataset

Order number	Points in cluster			Accuracy rate
	C1	C2	C3	
1	49	15	86	75.33%
2	54	46	50	77.33%
3	35	65	50	78.00%
4	49	34	67	84.00%
5	50	35	65	78.00%
6	50	21	79	74.00%
7	23	52	75	78.00%
8	13	87	50	75.33%
9	83	50	17	76.67%
10	50	78	22	81.33%
Correct number	50	50	50	
average				77.80%

Table 3 KPSO clustering on Iris dataset

Clusters found	Points in cluster	Coming from			Accuracy rate
		I	II	III	
C1	50	50	0	0	100%
C2	62	0	48	14	96%
C3	38	0	2	36	72%
Total	150				89.33%

Table 4 KPSO clustering on Wine dataset

Order number	Points in cluster			Accuracy rate
	C1	C2	C3	
1	47	69	62	70.22%
2	30	100	48	57.87%
3	28	100	50	56.74%
Correct number	59	71	48	
average				61.62%

Ten different clustering results of PSO algorithm are depicted in table 2 on Iris dataset. The lowest accuracy rate, the highest accuracy rate and the average are 74.00%, 84.00% and 77.80%, respectively. We find that the clustering result is not stable by conducting experiments. The number of different cluster and the corresponding accuracy rate are always changing. The clustering result of PSO algorithm on Wine dataset is very bad. The correct result should be three clusters, but we often get one cluster.

Table 5 EFCPSO clustering on Iris dataset

Clusters found	Points in cluster	Coming from			Accuracy rate
		I	II	III	
C1	50	50	0	0	100%
C2	54	0	45	9	90%
C3	46	0	5	41	82%
Total	150				90.67%

Table 6 EFCPSO clustering on Wine dataset

Clusters found	Points in cluster	Coming from			Accuracy rate
		I	II	III	
C1	63	50	6	7	84.75%
C2	58	1	52	5	73.24%
C3	57	8	13	36	75.00%
Total	178				77.53%

Clustering results of KPSO algorithm are showed in table 3 and table 4 on Iris and Wine datasets. On Iris dataset, the accuracy rate of KPSO is 89.33% and rises up 11.53% compared to that of PSO. On Wine dataset, the clustering result of KPSO is very good compared with only one cluster of PSO. We can see that the clustering result of KPSO improves and is stable.

The clustering result of EFCPSO algorithm is described in table 5 and table 6. The first cluster is completely correct, only 5 data points and 9 data points are misclassified in the second cluster and in the third one. Rate of accuracy rate is 100%, 90% and 82%, respectively. The total accuracy rate arrives to 90.67% and is very high. Compared with PSO and KPSO, EFCPSO rises up 12.87% and 1.34% on Iris dataset, 15.91% on Wine dataset. Furthermore, the result is very stable which is very importantly. By running 10 times even more, EFCPSO algorithm can all obtain stable and good clustering result. Moreover, PSO may get 2 clusters on Iris dataset in the worst case.

The scatter diagrams of PSO and EFCPSO algorithms are showed in Fig. 2[20] and Fig. 3 on Iris and Wine datasets. The PSO is run many times and we select one of them. The result of the first cluster is good, but the second and third clusters are not good. Only several data points are classified into the second cluster and many points that belong to the second cluster are wrongly divided into the third cluster. Compared with Fig.2, the result of Fig.3 is better and the number of wrongly classified points is less. Fig.4 is the result of EFCPSO on Wine dataset, it is obvious that dataset is divided into three cluster. Otherwise, the PSO may get two clusters even one cluster.

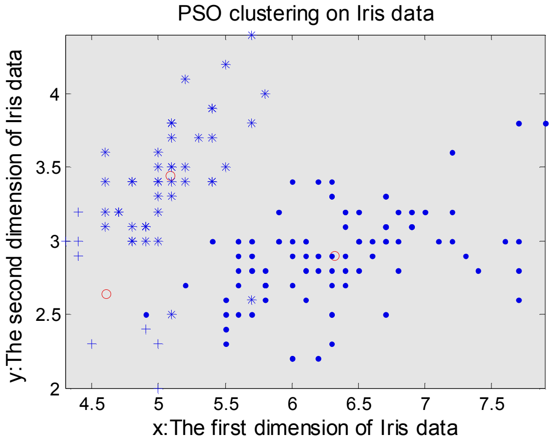

Fig.2 clustering result: PSO algorithm

Fig.5 shows the optimal fitness value. The relevant experiments are run 10 times The optimal fitness value of PSO is big and instable. It is 153.2230 in the worst case. The optimal fitness value of K-means is large which is 152.4 at the beginning and good in the end. The optimal fitness value of KPSO is smaller than that of above two algorithms. It is only 123.9695 which reduce 28.4305, 29.2535, respectively. EFCPSO is the most stable.

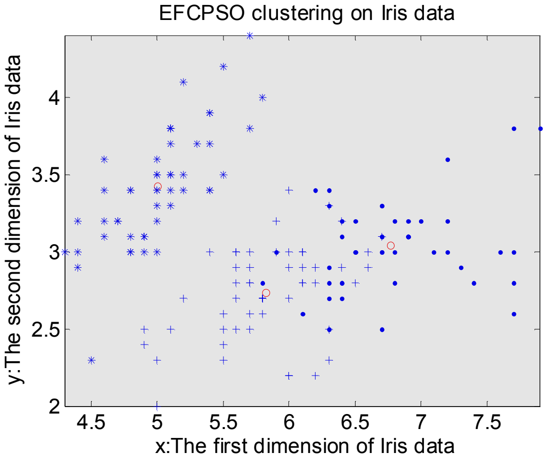

Fig.3 Scatter diagram of EFCPSO

From three aspects of best, worst and average accuracy rates, tables 7 and 8 summarize the comparative result of four different clustering algorithms on Iris and Wine datasets. Compared with K-means, PSO and KPSO, EFCPSO algorithm has higher accuracy and better stability. For K-means which is the traditional partitional algorithm, the best, the worst and the average accuracy rates are 89.33%, 52.67% and 61.40%, respectively. Compared to those of PSO, EFCPSO improves 6.67%, 16.67% and 12.87% in the values of the best, the worst and the average accuracy rate, respectively. Compared to those of KPSO, EFCPSO all improves 1.34% in three corresponding accuracy rates. Compared with Iris dataset, EFCPSO improves more obviously on Wine dataset. Percentage is 7.31%, 20.79% and 15.91% in three accuracy rates.

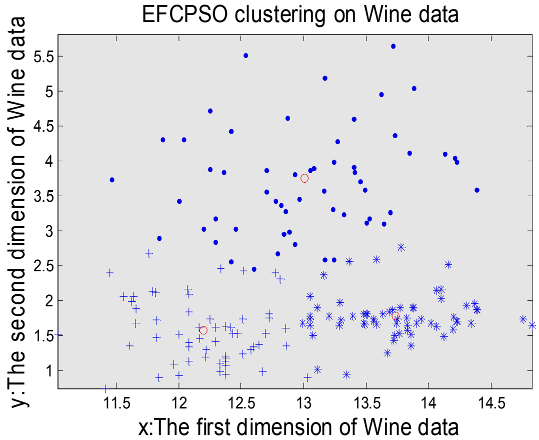

Fig.4 Scatter diagram of EFCPSO

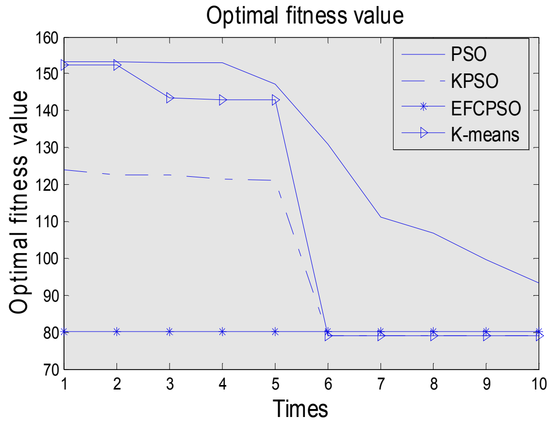

Fig.5 optimal fitness value of four algorithms

Table 7 clustering comparison of four algorithms on Iris dataset

algorithm	Best accuracy rate	Worst accuracy rate	Aver accuracy rate	Ave Iter num
K-means	89.33%	52.67%	61.40%	7
PSO	84.00%	74.00%	77.80%	40
KPSO	89.33%	89.33%	89.33%	2
EFCPSO	90.67%	90.67%	90.67%	2

From above experiments, we can come to a conclusion that the improved PSO clustering algorithm, that is EFCPSO clustering algorithm, has higher accuracy and better stability than those of PSO clustering algorithm, KPSO algorithm and traditional K-means algorithm.

Table 8 clustering comparison of four algorithms on Wine dataset

algorithm	Best accuracy rate	Worst accuracy rate	Aver accuracy rate	Ave Iter num
K-means	70.22%	53.37%	61.80%	10
PSO	39.89%	39.89%	39.89%	1
KPSO	70.22%	56.74%	61.62%	2
EFCPSO	77.53%	77.53%	77.53%	2

6 Conclusion

The clustering problem can be regarded as a kind of particular global optimization problem. Hence, the PSO algorithm which solves the optimization problem can also be used to cluster data. The performance of PSO algorithm depends on the initial cluster centers and might converge to local optimum. It can generate different clustering results when initialized with different clusters and can not guarantee unique clustering.

In order to solve this problem, an improved PSO algorithm by initializing cluster center using EFC algorithm is proposed in this paper. Through relevant experiments, the proposed algorithm is evaluated. The result demonstrates that EFCPSO algorithm has higher accuracy and better stability than that of traditional PSO clustering algorithm with random cluster centers. And the performance of EFCPSO is also better than that of KPSO.

In future work, it needs to be further studied whether it is suitable for larger amount of dataset. The following research work can also focus on the combination of PSO and two or more kinds of algorithms. In addition, we can combine entropy and discrete PSO algorithm.

Acknowledgment

This work is supported by the Science-Technology Program of the Higher Education Institutions of Shan-dong Province, China(No.J12LN22), Research Award Foundation for Outstanding Young scientists of Shandong Province, China (No.BS2012DX041) ， the National Natural Science Foundation of China (Grant No. 61472231).

References:

[1] Han, Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kauffmann Publishers, 2001, San Francisco.

[2] F. Y. Cao, J. Y. Liang, G. Jiang, An initialization method for the K-means algorithm using neighborhood model, Computers and Mathematics with Applications, Vol.58, No.3, 2009, pp. 474-483.

[3] Sandeep Rana, Sanjay Jasola, Rajesh Kumar, A review on particle swarm optimization algorithms and their applications to data clustering, *Artificial Intelligence Review*, Vol.35, No.3, 2011, pp. 211–222.

[4] Hamid Masoud, Saeed Jalili, Seyed Mohammad Hossein Hasheminejad, Dynamic clustering using combinatorial particle swarm optimization, *Applied Intelligence,*Vol.38,No.3,2013,pp. 289–314.

[5] Yucheng K, Szu-Yuan. L (2009) Combining K-means and particle swarm optimization for dynamic data clustering problems, *IEEE international conference on intelligent computing and intelligent systems (ICIS)*, Vol.1, 2009, pp. 757–761.

[6] MERWE D W vander, ENGELBRECHT A P. Data clustering using particle swarm optimization[C] // Proc of IEEE Congress on Evolutionary Computation, 2003:215-220.

[7] Shiming Zheng, Zhuang Miao,Zilin Song, etal. Clustering Algorithm Based on Fuzzy theory in the Circumstance of WEKA[J]． Journal of Liberation Army Technology University: Natural Science Edition,2012,13(1) : 22-26.

[8] Asha Gowda Karegowda，Vidya T，Shama，et al． Improving performance of K-means clustering by initializing cluster centers using genetic algorithm and entropy based fuzzy clustering for categorization of diabetic patients [J].Advances in Intelligent Systems and Computing，2012，174: 899-904.

[9] Vidyut Dey, Dilip Kumar Pratihar, Datta G L． Genetic algorithm-tuned entropy-based fuzzy C-means algorithm for obtaining distinct and compact clusters [J]. Fuzzy Optim Decis Making，2011,10: 153-166.

[10] Xiya Sun. Corporation Detection Algorithm Based on the Least Entropy Clustering [J]． Electronic Science and Technology, 2012, 25(3) : 13-16,20

[11] Cun Zhu, Yuanping Ni. The Research on EFC-RBF Nervous Network Algorithm and Breakdown Pattern Recognition [J]. Journal of Yunnan University: Natural Science Edition，

2009，31(S2): 182-186.H. B. Xiao, P. Guo, Iris Image Analysis Based on Affinity Propagation Algorithm, W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, 2009, pp. 943–949.

[12] J. Kennedy, R. C. Eberhart, Particle swarm optimization, In Neural networks 1995 proceedings, IEEE international conference, New Jersey, USA, 1995, pp. 1942–1948.

[13] R. C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, In:Proc. of the 6th International Symposium on Micro Machine and Human Scince, Nagoya, Japan, 1995, pp. 39–43.

[14] Li-Yeh Chuang, Chih-Jen Hsiao, Cheng-Hong Yang, Chaotic particle swarm optimization for data clustering, Expert Systems with Applications, Vol.38, No.12, 2011, pp. 14555–14563.

[15] Yao J., Dash M., Tan S.T., Liu H.: Entropy based fuzzy clustering and fuzzy modeling. Fuzzy Sets and Systems, 113(3), 2000, pp: 381–388.

[16] Aswatha Kumar M. et al. Improving Performance of K-Means Clustering by Initializing Cluster Centers Using Genetic Algorithm and Entropy Based Fuzzy Clustering for Categorization of Diabetic Patients. (Eds.): Proceedings of ICAdC, AISC 174, pp. 899–904.

[17] Jiang Chuanwen, Etorre Bompard, A self-adaptive chaotic particle swarm algorithm for short term hydroelectric system scheduling in deregulated environment, Energy Conversion and Management, Vol.46, No.17, 2005, pp. 2689–2696.

[18] R.Hruschka, RJGB. Campello, AA. Freitas, ACPLF. Carvalho, (2009) A survey of evolutionary algorithms for clustering. IEEETrans Syst Man Cybern, Part C, Appl Rev, Vol.39, No.2, 2009, pp. 133–155.

[19] UCI Machine Learning Repositpory, December 5, 2011, Available from: http://archive.ics.uci.edu/ml/

[20] Yuyan Zheng. Jianhua Qu. Yang Zhou, An Improved PSO Clustering Algorithm Based on Affinity Propagation, WSEAS TRANSACTIONS on SYSTEMS, 447-456, 12(9),2013.