

An Improved PSO Clustering Algorithm Based on Affinity Propagation

YUYAN ZHENG¹, JIANHUA QU¹, YANG ZHOU²

¹School of Management Science and Engineering
Shandong Normal University
Jinan, Shandong
CHINA

²College of Information Engineering
Northwest A&F University
Yangling, Shannxi
CHINA

zyy0716_source@163.com, qujh1978@163.com, zhouyang_online@163.com

Abstract: -Particle swarm optimization (PSO) is undoubtedly one of the most widely used swarm intelligence algorithm. Generally, each particle is assigned an initial value randomly. In this paper an improved PSO clustering algorithm based on affinity propagation (APPSO) is proposed which provides new ideas and methods for cluster analysis. Firstly the proposed algorithm get initial cluster centers by affinity propagation. Secondly obtained initial cluster centers are regarded as inputs of one of all particles instead of being assigned randomly. Finally we cluster with the improved PSO clustering algorithm. Through experiment test, we demonstrate that the improved PSO clustering algorithm has not only high accuracy but also certain stability.

Key-Words: -Particle Swarm Optimization (PSO); Affinity Propagation Clustering; Clustering Algorithm

1 Introduction

Clustering is one of most important methods in data analysis. It means the act of partitioning dataset with n data points into k clusters such that the data points in the same cluster are more similar than those in different cluster. It has been applied in different fields such as data mining, knowledge discovery, machine learning, statistics and so on [1].

There has been a variety of clustering algorithms currently. Among them, partitional algorithm and hierarchical algorithm are basic ones. In addition, density-based, model-based and grid-based are also popular clustering methods. With the development of technique at the aspect of clustering analysis, relevant intelligence algorithm is gradually developed. Particle swarm optimization (PSO) is one of which and it is shaped by investigating the behavior of fishes and birds[2]. It has less

parameters to adjust. Each particle is updated by following the best value in the search space. We will get optimal clustering result when the fitness value is minimum.

So far, varieties of improved PSO algorithms have been proposed. Jarboui et al.[3] presents a clustering approach based on the Combinatorial Particle Swarm Optimization (CPSO) with fixed number of clusters. It is an algorithm to solve problems of combinatorial optimization. But it needs to decide the number of clusters in advance. However, it is usually difficult to decide the correct number of clusters in many clustering problems. In order to estimate the number of clusters correctly, paper [4] proposes a dynamic clustering technique which can automatically find the best number of clusters and divide data points into corresponding clusters effectively. This method also improves the ideas proposed by Jarboui et al. [3]. In another study, Yucheng and Szu-Yuan [5] present a clustering

approach with variable number of clusters, and they use CPSO and K-means algorithms [4]. Paper [6] is the combination of PSO and Fuzzy C-Means and the utilization for the wetland extraction. There is also an integration of PSO and K-means algorithm (KPSO). Paper[7] further improves KPSO algorithm by proposing an enhanced cluster matching. Prior to the PSO updating process, the sequence of cluster centroids encoded in a particle is matched with the corresponding ones in the global best particle with the closest distance[7]. A comprehensive review of PSO algorithm and their applications to data clustering can be found in paper [2].

Affinity propagation (AP) takes as input a collection of real-valued similarities between data points to cluster. It regards all data points as cluster centers at the beginning. By adjusting the value of the preference gradually, we can obtain different number of clusters. The median of the similarities is often adopted as the value of the preference.

AP clustering algorithm is a new approach in data mining. The improved algorithm of AP is less than that of the other algorithms. Xia et al. [8] proposes a local and global approach of AP which can be used to cluster large scale data with a dense similarity matrix. And these two variants of AP speed up clustering. Literature [9] presents an adaptive affinity propagation clustering algorithm, which adaptively introduces ways of producing the P value and *lam* (*lam* is damping coefficient in AP) according to dataset. Paper [10] extends AP to a fuzzy variant and proposes a Fuzzy-AP algorithm, which returns fuzzy assignments to the cluster exemplars. Paper [11] gives the application of AP in community identification of financial market. In addition, it can also be used as Iris image analysis [12] and the other aspects.

This paper combines two algorithms above to present improved PSO algorithm. Firstly, we get a cluster center through AP clustering algorithm. Then we take it as the position value of one of the particles during the initialization of the particles. Finally, we cluster with the improved PSO clustering algorithm and at the same time introduce relevant experiment analysis.

The remainder of this paper is organized as follows. In section 2 PSO algorithm is described briefly. Section 3 shows AP clustering algorithm. Proposed PSO clustering algorithm is introduced in section 4. Experimental results and discussion follows in section 5. Finally we report the conclusion in section 6.

2 An Overview of Particle Swarm

Optimization

2.1 Basic Particle Swarm Optimization

Algorithm

The original PSO algorithm was developed by Kennedy and Eberhart in 1995[13][14]. It is inspired by bird flocking and fish schooling behavior. It is a kind of population-based optimization algorithm, each individual in the population is called a particle. The basic idea is to randomly initialize a group of particles which are massless and have no volume. Each particle is regarded as a candidate solution of the optimization problem. The performance of each particle is measured according to a fitness function, which is pre-defined and associated with the problem to be solved.

Each particle moves in the search space, updating its velocity and position according to the formulas of velocity and position. Generally, each particle attempts to follow the best particle that has been found by now, and finally obtains the optimal solution through iterative search. In basic PSO algorithm, each particle updates the velocity using Eq.(1)[15] given below.

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

Where:

1. v_{id} is the velocity of the i th particle in the d th dimension;
2. w is an inertia weight which usually linearly decreases during the iteration. The inertia weight w plays an important part in balancing the local and global search. The bigger w is, the quicker the velocity of particle is, the particle will carry out global search by the larger step; On the contrary, the particle will carry out search in local space carefully;
3. c_1, c_2 are regulatory factors which regulate global and local search and in the rang [0,2];
4. $r_1, r_2 \sim U(0,1)$, two random numbers generated by uniform distribution in the range [0,1];

5. p_{id} is the best positions found thus far by the i th particle while p_{gd} is that found by all the particles, that is, the personal best position and global best position. The computational formula is Eq.(2):

$$p_{id}(t+1) = \begin{cases} p_{id}(t) & \text{if } f(x_{id}(t+1)) \geq f(p_{id}(t)) \\ x_{id}(t+1) & \text{if } f(x_{id}(t+1)) < f(p_{id}(t)) \end{cases} \quad (2)$$

6. t is the iteration count.

The updating equation [15] of position of each particle is as follow:

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (3)$$

Where, x_{id} is the position of the i th particle in the d th dimension.

The general procedure of the basic PSO algorithm is as follows [16]:

1. Initialize the velocity and position of each particle randomly. Decide the population size.
2. Calculate the fitness value of each particle according to a fitness function.
3. Compare the fitness value of each particle with the previous individual best fitness value of this particle, personal best position of particle is replaced if better.
4. Compare the fitness value of each particle with the previous global best fitness value, global best position is replaced if better.
5. Update the velocity of particle according to the Eq. (1).
6. Update the position of particle according to the Eq. (3).
7. Repeat steps 2 to 6 until the termination condition is met (usually fixed number of iteration or good fitness value).

2.2 Particle Swarm Optimization for Data Clustering

Over the past few years, PSO has been proved to be both effective and fast for solving optimization problems. PSO shows a promising performance on

nonlinear function optimization and has thus received much attention [17]. It has less parameters to adjust, the speed of convergence is fast when the search space is low. In addition, it is easy to operate. Because of above advantages, it has been used to solve the clustering problems by many researchers.

Merwe is the first one of all researchers. He has presented two kinds of PSO clustering algorithms based on K-means. After that, relevant scholars improve the way of encoding of particle and aim function of the Merwe's algorithm and get good results.

The PSO clustering algorithm, like many other clustering algorithms, is aimed to minimize intra-cluster distances as well as maximize inter-cluster distances. Each particle is defined as a centroid. The velocity, the position and fitness value are updated according to formulas iteratively. Data points are partitioned into different cluster according to the nearest neighboring rule. After fixed number of iteration or good fitness value, the clustering result can be obtained. The fitness value is used to evaluate the performance of each solution.

In PSO clustering algorithm, the way of encoding of particle is based on cluster centers. A particle represents k cluster centers. Assuming that dataset D is divided into k clusters, then each particle C_i represents k cluster centers and is encoded as follows:

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

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

3 Affinity Propagation Clustering Algorithm

Affinity propagation clustering algorithm was developed by Brendan J.Frey and Delbert Dueck in 2007[18]. It is an unsupervised clustering algorithm and doesn't need prespecify cluster number. It takes as input a collection of real-valued similarities between data points and clusters by passing messages between data points. In this paper, we take Euclidean distance as measure index. The similarity between two data points is a negative squared

Euclidean distance. In addition, it devises a value for each data. These values are referred to as “preferences” (P).

Affinity propagation clustering algorithm exchanges two kinds of messages between data points. It is defined as responsibility (R) and availability (A), respectively. The responsibility $R(i, k)$, sent from data point i to candidate exemplar point k , reflects the accumulated evidence for how well-suited point k is to serve as the exemplar for point i , taking into account other potential exemplars for point i . The availability $A(i, k)$, sent from candidate exemplar point k to point i , reflects the accumulated evidence for how appropriate it would be for point i to choose point k as its exemplar, taking into account the support from other points that point k should be an exemplar [18]. The procedure is that of updating two kinds of messages iteratively.

The formula of responsibility and availability is as under:

$$R(i, k) \leftarrow S(i, k) - \max_{j \neq k} \{A(i, j) + S(i, j)\} \quad (5)$$

$$A(i, k) \leftarrow \min \{0, R(k, k) + \max_{j \neq i, k} \{0, R(j, k)\}\} \quad (6)$$

$$A(k, k) \leftarrow \max_{j \neq k} \{0, R(j, k)\} \quad (7)$$

Where:

1. $j \in \{1, 2, \dots, N, j \neq k\}$;
2. $S(i, k) = -\|X_i - X_k\|^2$ $i, k \in \{1, 2, \dots, N\}$
 $S(i, k)$ represents similarity and indicates how well the data point with index k is suited to be the class center for data point i ;
3. $R(k, k)$ is the self-responsibility while $A(k, k)$ is the self-availability.

In order to void numerical oscillations during the procedure of exchanging, it is essential to set a damped coefficient lam . The effect of lam is set as follows:

$$R_i = (1 - lam) \times R_i + lam \times R_{i-1} \quad (8)$$

$$A_i = (1 - lam) \times A_i + lam \times A_{i-1} \quad (9)$$

Where, lam is damping coefficient and $lam \in [0.5, 1)$, it adjusts the speed of convergence and the stability of iteration, iteration number will

reduce when lam becomes small while iteration number will increase when lam becomes big.

The steps of affinity propagation clustering algorithm are as follows [8]:

1. Input is the matrix of real-valued similarities between data points $S(i, k)$ and the preference of each data point. Initialize availabilities $A(i, k)$ to zero.
2. Update responsibility $R(i, k)$ using Eq.(5).
3. Update availability $A(i, k)$ and self-availability $A(k, k)$ using Eqs.(6),(7).
4. The message-passing procedure may be terminated after a fixed number of iteration, after changes in the messages stay stable or fall below a threshold for some number of iteration.

4 Proposed PSO Clustering Algorithm

4.1 Fitness Function

Kennedy and Eberhart (1995) suggested a fitness value associate with each particle. There has been many fitness functions described in the literature so far. Krovi [19] proposes a fitness function to assess partitions formed by only two clusters. Bandyopadhyay [20] intends to use a fitness function based on the Davis-Bouldin (DB) index which commonly is used as relative validity criteria for clustering.

In this paper, the fitness function is as follows:

$$F = \sum_{i=1}^k \sum_{j=1}^{n_i} \|X_{ij} - C_i\|^2 \quad (10)$$

Where, X_{ij} is the j th data in the i th class, C_i is center of the i th class, n_i is number of data in the i th class. k is number of cluster.

4.2 Inertia weight

Inertia weight is an important parameter of PSO algorithm and it can affect the performance of algorithm. Generally, the bigger w will enhance the capability of global search while the smaller w will enhance the capability of local search. There has been a variety of inertia weights by now. Shi and Eberhart(1998) study the selection of inertia weight and show that the speed of convergence of algorithm is higher when w is in $[0.8,1.2]$. Paper [21] provides a self-adaptive inertia weight which is the way of index. Shi and Eberhart(2001) provide a random inertia weight.

In this paper, w is the following linear differential decline inertia weight:

$$w(t) = w_{star} - \frac{(w_{star} - w_{end})}{t_{max}^2} \times t^2 \quad (11)$$

Where, $w(t)$ is the current inertia weight, w_{star} is initial inertia weight, w_{end} is final inertia weight, t is the current iteration count, t_{max} is the maximum iteration count.

4.3 Particle Encoding

Several encoding schemes have been presented in the literature so far. We can divide them into three types: binary, integer, and real encoding [22]. And each type has several ways of representing a partition. In binary encoding, each clustering partition is usually represented as a binary string of length N , where N is the number of data points. In integer encoding, each partition is an integer vector of length N , where N is the number of data points, these integers define the cluster labels that denote the partition of data points. In real encoding, the genotypes are made up of real numbers that represent cluster centers.

In this paper, we adopt the real encoding and a particle represents k cluster centers. Provided that dataset D is divided into k clusters, then each particle C_i represents k cluster centers and is encoded as the Eq.(4). If dataset D is in an n dimensional space and is divided into k clusters, then its length is $n \times k$. Thus, the first n values are the first cluster center. The next n values are the second cluster center, and so forth.

In order to illustrate the encoding of the particle, we take a particle for a two-dimensional problem with three clusters for an example. It is encoded

in $C_1 = (-4, 3, 15, 32, 2.7, -6.5)$. Three cluster centers represented by this particle are $(-4, 3)$, $(15, 32)$ and $(2.7, -6.5)$, respectively.

4.4 Combine with Affinity Propagation Clustering Algorithm

In this paper, we present an improved PSO clustering algorithm based on affinity propagation (APPSO). The procedure of APPSO algorithm is then described in detail. Firstly, we adjust the value of P in AP algorithm and run AP to get a proper cluster center with proper number of clusters. The number of clusters needs to be pre-defined. Then the proper cluster center is taken as one of particles during the initialization of particles, the rest of the particles is initialized randomly. Cluster centers are obtained. Data points are divided into corresponding cluster according to the following rule of the nearest neighbor. The distance between data point $X_m (m = 1, 2, \dots, n)$ and cluster center $C_{ij} (j = 1, 2, \dots, k)$ satisfies Eq.(12), then data point X_m is classified into cluster C_{ij} . After that, we calculate the fitness value using Eq.(10). By comparing fitness values, we find personal and global best positions. Finally, we update the particle and classify data points and calculate the fitness value iteratively.

$$\|X_m - C_{ij}\| = \min \{\|X_m - C_{i1}\|, \dots, \|X_m - C_{ik}\|\} \quad (12)$$

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

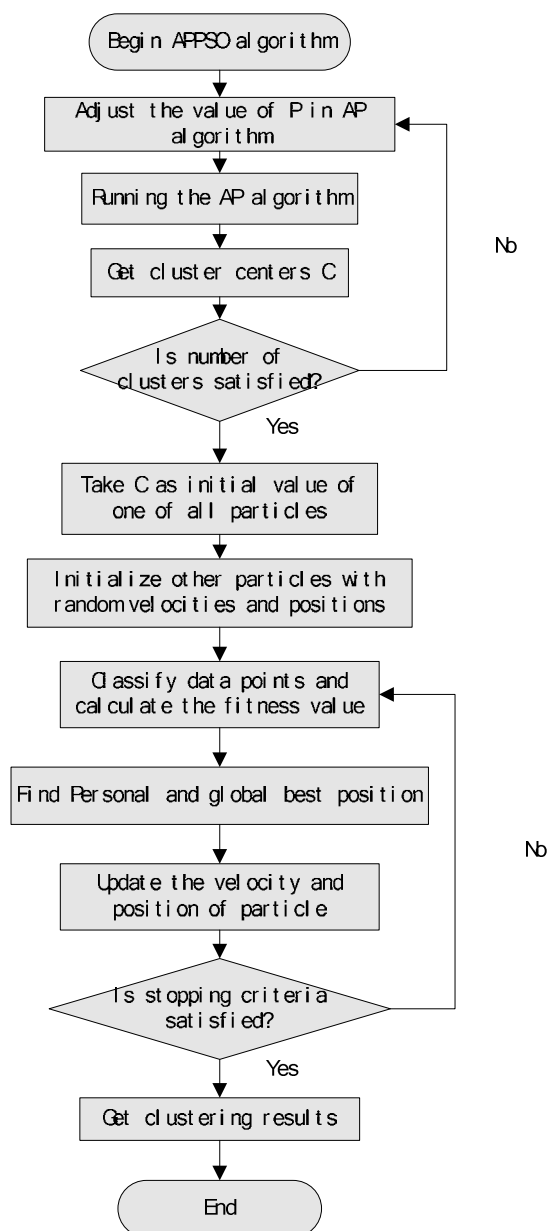


Fig.1 Flow chart showing basic steps in the APPSO clustering algorithm

The steps of the APPSO clustering algorithm are as follows:

1. Run AP algorithm and obtain a cluster center C which satisfies the number of clusters.
2. Initialization the velocity and position: C is used as the position value of one of the particles while the rest of particles is generated by selecting random numbers in the range $[0,1]$.
3. Classify data points and calculate the fitness value of each particle using Eq. (10).
4. Compare the fitness value of each particle with

the previous individual best fitness value of this particle, personal best position of particle is replaced if better.

5. Compare the fitness value of each particle with the previous global best fitness value, global best position is replaced if better.
6. Update the velocity of particle using Eq. (1) from the PSO algorithm.
7. Update the position of particle using Eq. (3) from the PSO algorithm.
8. Repeat steps 3 to 7 until the termination condition is met (usually fixed number of iteration or good fitness value).

Fig.1 is the flow chart which shows basic steps of the APPSO clustering algorithm.

5 Experimental Results and Discussion

5.1 Description of the dataset

In this paper, we conduct experiment using Iris and Wine datasets. It can be obtained from UC Irvine machine learning repository [23]. These datasets have often been used as a standard for testing clustering algorithms.

Iris dataset has three classes that represent three different varieties of Iris flowers and they are named Iris setosa(I), Iris versicolor(II) and Iris virginica(III), respectively. Each of the three classes has fifty objects, thus a total of 150 objects is available. Every object is described by four attributes, viz sepal length, sepal width, petal length and petal width [1].

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. There are overall 178 samples. Every sample is described by a set of thirteen attributes. There are 59, 71 and 48 samples in class I, class II and class III respectively. The classes are separable.

5.2 Results and discussion

This section shows the performance of PSO for data

clustering. It can further be improved by applying the result of AP algorithm. We have conducted relevant experiments of K-means, the integration of K-means and PSO (KPSO), PSO and APPSO algorithms on Iris and Wine datasets 10 times. The experiment measures the performance of the algorithm with accuracy rate (AR) in Eq.(13).

$$AR = \frac{n}{N} \times 100\% \quad (13)$$

Where, n is the number of correctly classified data points, N is the number of total data point.

The experimental results have been summarized in Tables 1,2,3,4 and 5.

Table 1 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 2 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 3 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%

Table 4 APPSO clustering on Iris dataset

Clusters found	Points in cluster	Coming from			Accuracy rate
		I	II	III	
C1	50	50	0	0	100%
C2	49	0	44	5	88%
C3	51	0	6	45	90%
Total	150				92.67%

Table 5 APPSO clustering on Wine dataset

Clusters found	Points in cluster	Coming from			Accuracy rate
		I	II	III	
C1	47	46	1	0	77.97%
C2	68	0	50	18	70.42%
C3	63	13	20	30	62.50%
Total	150				70.79%

Table 1 shows 10 different clustering results of PSO algorithm on Iris dataset. The highest accuracy rate is 84.00% while the lowest accuracy rate is 74.00%, the average is 77.80%. It is obvious that the result of clustering is not stable. The clustering result of PSO algorithm on Wine dataset is very bad. In most cases, we get one cluster and the correct result should be three clusters.

Table 2 and Table 3 show clustering results of KPSO algorithm on Iris and Wine datasets. The accuracy rate of KPSO is 89.33% and raises 11.53% compared to that of PSO on Iris dataset. The clustering result of KPSO on Wine dataset is very good compared with only one cluster of PSO. It is obvious that the clustering result of KPSO improves

and is stable.

The clustering result of APPSO algorithm is described in Table 4 and Table 5. We can see the first cluster is completely correct, only 6 data points in the second cluster is misclassified and only 5 data points in the third cluster. Rate of classification error is 0%, 12% and 10%, respectively which is very low. The total accuracy rate arrives to 92.67% and is very high. Compared with PSO and KPSO, APPSO raises 14.87% and 3.34% on Iris dataset, 9.17% on Wine dataset. Most importantly, the result is very stable. Unlike PSO, APPSO algorithm which is run 10 times even more can obtain stable and good clustering result. Moreover, PSO algorithm may get 2 clusters on Iris dataset in the worst case.

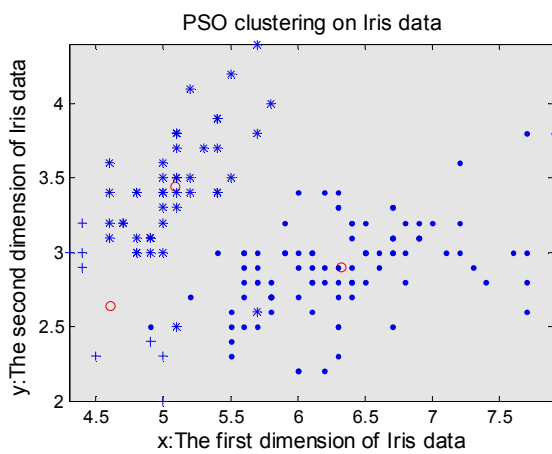


Fig.2 clustering result: PSO algorithm

Fig. 2 shows scatter diagram of PSO algorithm on Iris dataset while Fig. 3 shows that of APPSO. Fig.2 is one of many times of PSO clustering. It is obvious that the result of Fig.2 is not good and the number of the third cluster has only several points and many points that belong to the second cluster are wrongly divided into the third cluster. By comparison with Fig.2, the result of Fig.3 is better and the number of wrongly classified points is less.

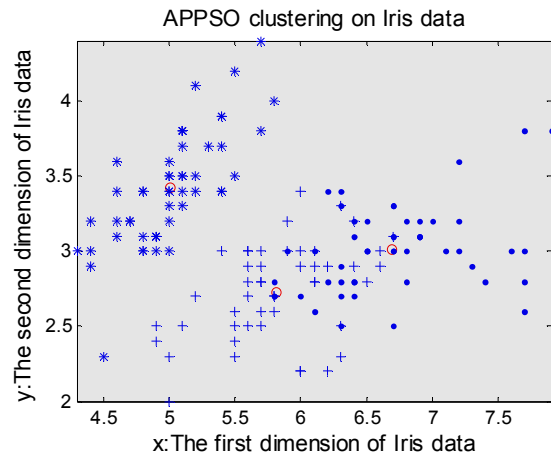


Fig.3 clustering result: APPSO algorithm

Optimal Fitness value of four algorithms is displayed in Fig.4. We run relevant experiment 10 times and get fitness value. The optimal fitness value of PSO is not only big but also instable. Although the optimal fitness value of K-means is large at the beginning, we can get good result in the end. The optimal fitness value of KPSO is smaller than that of K-means or PSO which is 152.4, 153.2230 in the worst case. KPSO is only 123.9695 which reduce 28.4305, 29.2535, respectively. APPSO is the most stable.

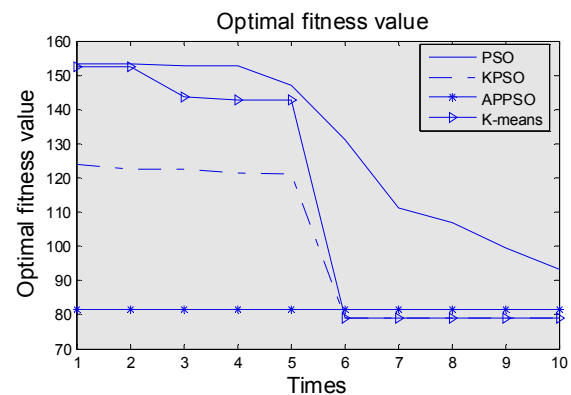


Fig.4 optimal fitness value of four algorithms
Table 6 clustering comparison of different algorithms on Iris dataset

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
APPSO	92.67%	92.67%	92.67%	2

Table 7 clustering comparison of different 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
APPSO	70.79%	70.79%	70.79%	2

Table 6 and Table 7 summarize the comparative result of three different clustering algorithms on Iris dataset. Compared with other clustering algorithms K-means, PSO and KPSO, APPSO algorithm has higher accuracy and better stability. For the traditional partitioned clustering algorithm K-means, the best accuracy rate is 89.33%, the worst is 52.67%, the average is 61.40%. APPSO improves 8.67% in the value of best accuracy rate, 18.67% in the worst, 14.87% in the average, compared to those of PSO. APPSO improves 3.34%, 3.34% and 3.34%, respectively, in three corresponding accuracy rates, compared to those of KPSO. Compared with KPSO, APPSO improves relatively less on Wine dataset than on Iris dataset. Percentage is 0.57%, 14.05% and 9.17% in three accuracy rates. But APPSO increases much than that of PSO itself.

The result explicitly demonstrates that APPSO clustering algorithm has higher accuracy and better stability than that of traditional PSO clustering algorithm and KPSO algorithm.

6 Conclusion

PSO is mainly a global optimization method and it can also be used to cluster data. Owing to the

inferior performance of PSO, an improved PSO algorithm which combines PSO with AP algorithms is present in this paper. We evaluate the proposed algorithm through experiment. The result demonstrates that APPSO clustering algorithm has higher accuracy and better stability than that of traditional PSO clustering algorithm. The performance of APPSO is also better than that of KPSO. The quality of solution of APPSO is very high.

In future work, we can improve PSO with intelligent algorithms and the other algorithms. Even we can combine PSO with two or more kinds of algorithms. Another research direction is determining the optimal number of clusters about PSO and cluster on higher dimensional problems and large number of patterns. Moreover, there is also much work to be done on proposing a way of improving PSO along with deciding clustering number.

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).

References:

- [1] 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.
- [2] 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.
- [3] B. Jarboui, M. Cheikh, P. Siarry, Combinatorial particle swarm optimization(CPSO) for partitioned clustering problem, *Appl Math Comput*, Vol.192, No.2,2007, pp.337-345.

- [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] H. Liu, T. Pei, C. Zhou, AX. Zhu, Multi-temporal MODIS-data-based PSO-FCM clustering applied to wetland extraction in the Sanjiang Plain, *International Conference on Earth Observation Data Processing and Analysis, Wuhan, China*, Vol.7285, 2008.
- [7] Yau-King Lam, P. W. M. Tsang, Chi-Sing Leung, PSO-based K-Means clustering with enhanced cluster matching for gene expression data, *Neural Comput & Applic*, Vol.22, No.7-8, 2013, pp.1349-1355.
- [8] D. Y. Xia, F. Wu, X. Q. Zhang, Y. T. Zhuang, Local and global approaches of affinity propagation clustering for large scale data, *Journal of Zhejiang University SCIENCE A*, Vol.9, No.10, 2008, pp. 1373-1381.
- [9] K. J. Wang, J. Y. Zhang, D. Li et.al, Adaptive affinity propagation clustering, *Acta Automatica Sinica*, Vol.33, No.12, 2007, pp. 1242-1246.
- [10] T. Geweniger, D. Zuhlke, B. Hammer, Thomas Villmann, Fuzzy Variant of Affinity Propagation in Comparison to Median Fuzzy c-Means, *J.C. Principe and R. Miikkulainen (Eds.): WSOM 2009*, LNCS 5629, 2009, pp. 72–79.
- [11] L. Hong, S. M. Cai, Z. Q. Fu, P. L. Zhou, Community Identification of Financial Market. F.L. Gao (Ed.): *Recent Progress in DEIT*, Vol.1, LNEE 156, pp. 121–127.
- [12] H. B. Xiao, P. Guo, Iris Image Analysis Based on Affinity Propagation Algorithm, *W. Yu, H. He, and N. Zhang (Eds.): ISSN 2009, Part II*, LNCS 5552, 2009, pp. 943–949.
- [13] J. Kennedy, R. C. Eberhart, Particle swarm optimization, *In Neural networks 1995 proceedings, IEEE international conference*, New Jersey, USA, 1995, pp. 1942–1948.
- [14] R. C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, *In: Proc. of the 6th International Symposium on Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39–43.
- [15] G. Xu, An adaptive parameter tuning of particle swarm optimization algorithm, *Applied Mathematics and Computation*, Vol.219, No.9, 2013, pp. 4560–4569.
- [16] Chi-Yang Tsai. I-Wei Kao, Particle swarm optimization with selective particle regeneration for data clustering, *Expert Systems with Applications*, Vol.38, No.6, 2011, pp. 6565–6576.
- [17] 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.
- [18] Brendan J.Frey, Delbert Dueck, Clustering by Passing Messages Between Data Points, *Science*, Vol.315, No.5814, 2007, pp. 972-976.
- [19] R. Krovi, Genetic Algorithms for Clustering: A Preliminary Investigation, *In Proc. of the 25th Hawaii Int. Conference on System Sciences*, Vol.4, 1992, pp. 540-544.
- [20] S. Bandyopadhyay, U. Maulik, Genetic Clustering for Automatic Evolution of Clusters and Application to Image Classification, *Pattern Recognition*, Vol.35, No.6, 2002, pp.1197-1208.
- [21] 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.
- [22] ER.Hruschka, RJGB. Campello, AA. Freitas, ACPLF. Carvalho, (2009) A survey of evolutionary algorithms for clustering. *IEEE Trans Syst Man Cybern, Part C, Appl Rev*, Vol.39, No.2, 2009, pp. 133–155.
- [23] UCI Machine Learning Repository, December 5, 2011, Available from: <http://archive.ics.uci.edu/ml/>