Clustering Algorithm Based on Artificial Bee Colony Optimization

Dandan Zhang[1, a], Ke Luo [1,b]

[1] Institute of Computer and Communication Engineering, Changsha University of Sciences and Technology, Changsha 410114, China

[a]daphnedwyyx@163.com, [b]luok@csust.edu.cn

Keywords: k-medoids clustering; granular computing; artificial bee colony; selection probability.

Abstract. After analyzing the disadvantages of sensitivity to the initial selection of the center, low clustering accuracy and the poor global search ability of k-medoids clustering algorithm, a clustering algorithm based on improved artificial bee colony (ABC) is proposed. By improving the initialization of bee colony, adjusting the search step dynamically with iteration increasing , and then introducing the selection probability based on sorting instead of depending on fitness directly, the ABC algorithm will quickly converge to global optimal. This paper will further optimize k-medoids to improve the performance of the clustering algorithm. The experimental results show that this algorithm can reduce the sensitive degree of the initial center selection and the noise, has high accuracy and strong stability.

Introduction

K-medoids algorithm is a classical algorithm to solve clustering problems [1]. Since it has the advantages in the convergence speed and local search ability, are widely used in data mining. However, it still have the problems of sensitive to initial centers, low clustering accuracy and poor global search ability. ABC [2, 3] algorithm is a novel swarm intelligence optimization algorithm by simulating the collecting nectar process of bee swarm which has few parameters, strong global search ability and robustness.

An improved algorithm based on initialization center fine-tuning and incremental center candidate set was proposed [4]. The computational time get slightly lower, but clustering accuracy is not high. Combining granular computing to select the highest density K granules center as the initial clustering centers, the algorithm (GCK) has been a certain degree of optimization, but the initial clustering centers may be located in the same cluster [5]. Gao and Liu introduced fine-tuning mechanism into ABC algorithm and discussed the perturbation factor range, improved the local search ability [6]. Wang introduced forgetting factor and neighborhood factor into onlooker bees' neighborhood search phase, enhanced the algorithm convergence [7].

In view of the above references is insufficient, an improved ABC algorithm which improves the colony initialization, dynamically adjusts neighborhood search step and introduces the probability based on sorting of onlooker bees is proposed. The improved ABC algorithm accelerate convergence speed and avoid premature convergence. Then we apply the improved ABC algorithm to optimize k-medoids. The experimental results show the proposed algorithm has faster convergence speed, higher accuracy and more stable performance.

Granular Computing

From the perspective of information granularity, the clustering analyze and solve problems under a uniform granularity [8].

Definition 1 Given the universe U , the partition of the knowledge P in U is $\{X_1, X_2, \cdots, X_n\}$, the particle density is defined in Eq. 1.

$$gd(X_i) = |X_i| / |U| . \tag{1}$$

Where $|X_i|$ is the number of objects in the ith divided block, and $|U|$ is the total object number of U. Suppose the number of particles is n, the average density of the particle is given in Eq. 2.

$$\overline{G} = \sum_{i=1}^{n} gd(X_i) \Big/ n \cdot \tag{2}$$

Definition 2 Suppose U is divided into $\{X_1, X_2, \cdots, X_n\}$, according to the attribute values (a_l), the attribute resolution of l (w_l) is defined in Eq. 3.

$$w_l = GD(a_l) = \sum_{i=1}^{n} |X_i|^2 \Big/ |U|^2 \cdot \tag{3}$$

Where U is universe, n is the number of divided blocks.

Definition 3 Given the cluster space $K = (U, A)$, U is universe. A is the attributes' set, the object similarity function is defined in Eq. 4.

$$S(x_i, x_j) = 1 \Big/ (1 + \sum_{l=1}^{|A|} w_l |x_{il} - x_{jl}|) \cdot \tag{4}$$

Where x_{il}, x_{jl} are the attribute l value of x_i, x_j respectively. If the number of objects is n, the object average similarity is defined in Eq. 5.

$$\overline{d} = \sum_{i,j=1}^{n} S(x_i - x_j) \Big/ n^2 \cdot \tag{5}$$

Definition 4 Suppose the object number of the ith particle is N, $\{x_{i1}, x_{i2}, \cdots, x_{iN}\}$, the particle's center is shown in Eq. 6.

$$v_i = \{x_{ij} \,|\, \min_{j=1}^{N} \left| x_{ij} - \sum_{k=1}^{N} x_{ik} \Big/ N \right| \} \cdot \tag{6}$$

Improved Artificial Bee Colony Algorithm

Improvement of Bee Colony Initialization. The initialization of the original ABC algorithm does not guarantee produced initial bee colony uniformly distribute within the solution space, and has great influence on the overall performance. A modified colony initialization combining granular computing and the maximum distances product method [9] is proposed, the process is as follows.

Step1 Calculate w_l, $S(x_i, x_j)$ and $\overline{d}$ by Eq. 3, Eq. 4 and Eq. 5. Threshold $d \in [\overline{d} - 0.2, \overline{d} + 0.2]$, if $\boldsymbol{S}(x_i, x_j) \geq d$, then $\boldsymbol{M}(i,j)=1$, else $\boldsymbol{M}(i,j)=0$. Where $\boldsymbol{M}$ is a fuzzy similarity matrix between objects.

Step2 Classify the objects according to the matrix $\boldsymbol{M}$ .Get the coarse particles set of objects, $\{X_1, X_2, \cdots, X_h\}$, $1 \leq h \leq n$, which will record the similar object number for each object.

Step3 Calculate the density of each particle. Calculate the average density of particles $\overline{G}$. Produce the effective particle set R according the following formula, $gd(X_i) \geq \overline{G}$.

Step4 Calculate the center of each particle in the R, and calculate the euclidean distance between any two particles, recording it in the matrix $\boldsymbol{D}$.

Step5 Select the largest particle density of R, corresponding to the center as the first cluster center o_1. Choose the center of the largest density's particle that is the farthest from o_1 as the second cluster center o_2. For the remaining particles in R, calculate the distance ($d_{i1}, d_{i2}, ..., d_{im}$) between the center of each particle and $o_1, o_2, ..., o_m$, according to the matrix $\boldsymbol{D}$. Select the center of the particle that has the maximum ($d = \max(d_{i1} \times d_{i2} \times \cdots \times d_{im})$) as o_i. In the same way, getting o_k.

Step 6 Regard these k cluster centers as initial position encoder of first bee. Calculate its fitness value, get the first bee. Randomly select an object within the each particle of the cluster center where the first bee as the second bee position encoder, then calculate the fitness value. Repeated $SN-1$ times to generate the initial colony$\{Z_1,Z_2,...,Z_{SN}\}$, which has the population size of SN.

The Adjustable Search Step. At the beginning of iteration, when all employed bees select a new candidate food source, are strive to expand the algorithm search space to quickly lock the approximate location of the optimal solution. After determining general direction of the optimal solution, bees should be refined local search to find optimal solution quickly. So we adjust step size adaptively by Eq. 7. During neighborhood search, bees can dynamically adjust the neighborhood search with iteration increasing, has faster search ability and more effective.

$$fai_{ij} = \varphi_{ij}\left(d_{max} - \left(d_{max} - d_{min}\right)C / MCN\right). \tag{7}$$

Where φ_{ij} is a random number between$\left[-1,1\right]$. The value d_{max} and d_{min} represent the maximum and minimum percentage of the position adjustment for the employed and onlooker bee. MCN is maximum number of iteration. C is current number of iteration.

The Selection Probability of Onlooker Bees. ABC algorithm select the best food source based on fitness value ratio method, according to the information of employed bees provided. It may lead to super bee and prematurity, entrapped in a local optimum. Use the selection probability based on sorting, to avoid a negative impact of extraordinary individual on the selection process.

According to the merits of the fitness value of employed bees, arrange bees in descending order. Calculate the selection probability (p_i) of onlooker bees by Eq. 8 and Eq. 9.

$$p_i = 1/M + Q_C\,(\mathrm{M}+1-2\mathrm{i})/(M(\mathrm{M}+1)). \tag{8}$$

$$Q_C = 0.2 + 3C/4MCN\ . \tag{9}$$

Where $i = 1, 2, \cdots, M$, M is the half of SN, Q_C is an adaptive parameter, MCN is the maximum number of iteration, C is the current number of iteration.

K-medoids Algorithm Based on Improved Artificial Bee Colony

The Fitness Function of the Algorithm. Fitness function is defined in Eq. 10, when it reaches the maximum fitness value, the algorithm will get the optimal spatial clustering results.

$$fit_i = 1/J_m\ . \tag{10}$$

Where fit_i is fitness value of the ith bee , J_m is inner-class distance, as shown in Eq. 11.

$$J_m = \sum_{j=1}^{k}\sum_{p\in c_j}\left|p\text{-}z_{ij}\right|^2 . \tag{11}$$

Where z_{ij} is the jth clustering center of ith bee, p represents non-cluster centers of c_j.

Update Clustering Center. The new center (a nearest object to cluster centers) is shown in Eq. 12.

$$v_{ij} = \{x_t \left| \min_{t=1}^{n} \left| x_t\text{-}z_{ij} \right| \right.\}\ . \tag{12}$$

Where i=1,2,...,SN, j=1,2,...,k, n is the total number of objects, x_t is a object, z_{ij} is bee position i dimension j (the cluster center).

Algorithm Description. The colony uses real coding which composed of the location and fitness value of bees. The location composed of k cluster centers, fitness value is calculated by Eq. 10 and Eq. 11. The bee encoding is shown in Eq. 13.

$Z_i=(z_{i1},z_{i2},...,z_{ik},fit_i)$. (13)

The detailed steps of the algorithm is given below.

Step1 Generate an initial population of SN individuals $\{Z_1,Z_2,...,Z_{SN}\}$.

Step2 Array bees in descending order according to the fitness value. The first half is employed bees, the later half is onlooker bees. Each employed bee which is placed at a food source that is different from others. Search in the neighborhood of its current position to find a better food source.

Step3 Apply greedy selection scheme to choose one of them (current position and new position). After employed bees complete neighborhood search, calculate probability p_i by Eq. 8 and Eq. 9.

Step4 Assign each onlooker bee to a employed bee, according to the roulette principle with probability proportional to p_i. Then produce new food positions for each onlooker bee.

Step5 Evaluate the fitness of each onlooker bee's current position and the new solution. Apply greedy selection scheme to keep the fitter one and discard other.

Step6 If a particular solution has not been improved over a number of cycles ($Limit$), then select it for abandonment. Replace the solution by placing a scout bee at a food source.

Step7 Carry out k-medoids clustering for each bee. Apply new cluster centers to update the colony.

Step8 If the best solution found is acceptable or the maximum iterations elapsed, stop and return the best solution found so far. Otherwise go back to Step 2 and repeat again.

Experimental Results Analysis

We implemented our algorithm in C++ and MATLAB based on Windows 7. The processor was the Intel Core 2 Duo CPU T5870 with a speed of 2 GHz and 1 GB of RAM.

This experiment selected datasets: one artificial dataset, Iris and Wine datasets. Parameter settings of the algorithm: $SN=100$, $MCN=100$, $Limit$=15. Parameters settings on artificial dataset: $d=0.4$, $d_{min}=0.25$, $d_{max}=1.1$. Parameter settings on Iris dataset: $d=0.8$, $d_{min}=0.2$, $d_{max}=1.0$. Parameter settings on Wine dataset: $d=0.018$, $d_{min}=0.3$, $d_{max}=1.0$.

With the increase of iteration, the change of fitness value in this algorithm is shown in Fig. 1.

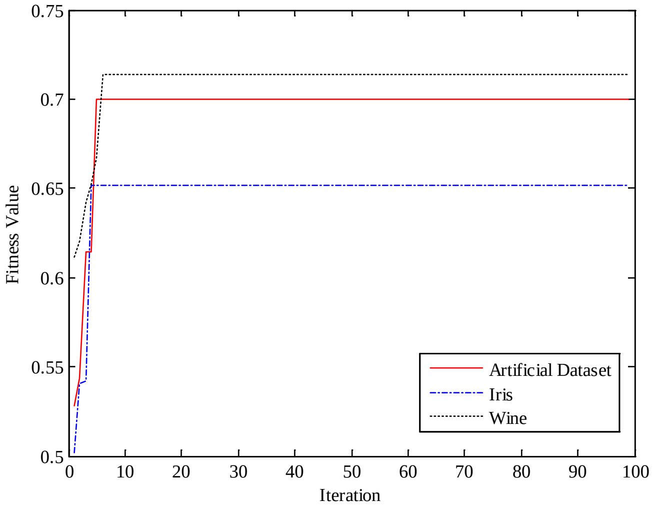

Fig. 1. The change of fitness by the algorithm runs on the above datasets.

Fig. 1 shows that the fitness value range of the k-medoids algorithm based on improved ABC is small. The algorithm reaches local extreme quickly by dynamically adjusting search step. Moreover, it escapes from local optima through the selection probability based on sorting and scout bees searching, avoiding premature, ultimately to the global optima. Obviously, the proposed algorithm can effectively avoid local optimal, converge more quickly, and has a strong stability.

Experiment on Artificial Dataset. To prove the efficiency of the proposed algorithm, it is compared with Partitioning Around Medoids (PAM) [1] and GCK [5] on randomly generated artificial dataset. The dataset contains 3 classes, 2 attributes and 220 instances, as shown in Fig. 2. Each algorithm runs 20 times. The clustering results of these algorithms are shown in Fig. 3 ~ Fig. 5.

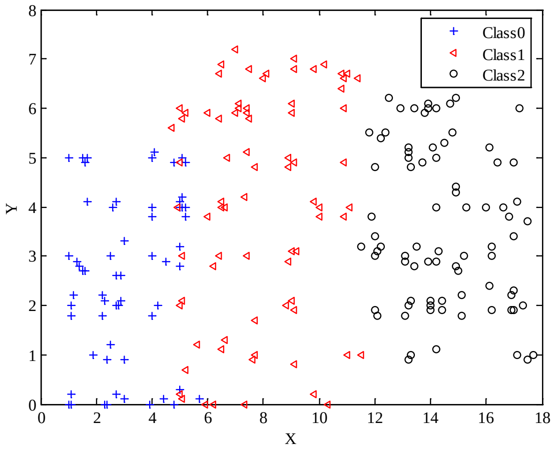

Fig. 2. The artificial dataset.

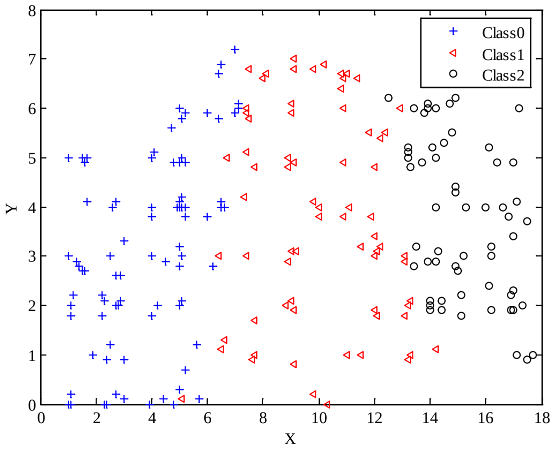

Fig. 3. The clustering results of PAM.

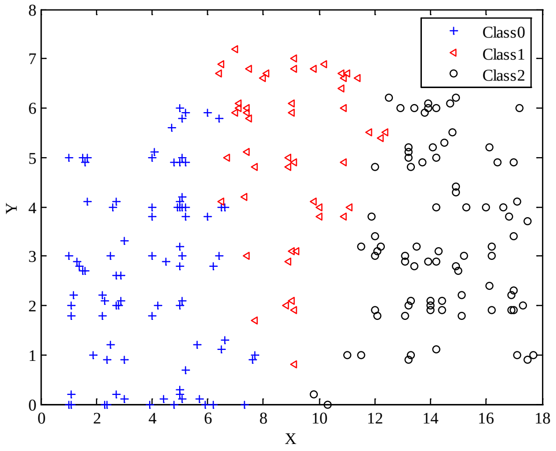

Fig. 4. The clustering results of GCK.

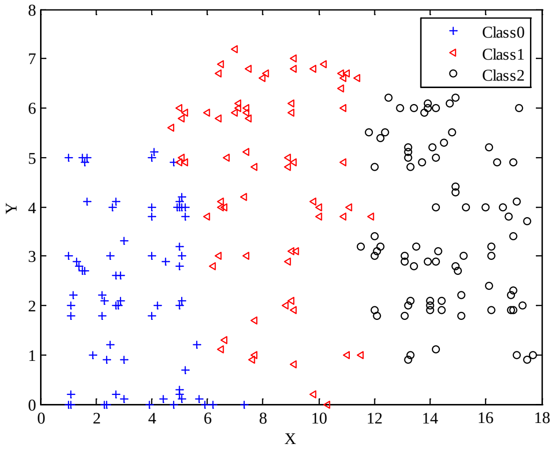

Fig. 5. The clustering results of the proposed algorithm.

Through experiments, the clustering accuracy of PAM, GCK and the proposed algorithm are 77.27%, 84.55% and 94.55% respectively. Compared with PAM and GCK, the proposed algorithm has better and stable clustering effect, higher accuracy. It can be seen the proposed algorithm has better processing ability for boundary data, is more similar to the original data, while the clustering results of PAM and GCK have larger difference with the original dataset and have lower accuracy.

Experiments on Iris and Wine Datasets. To further prove the advancement of the proposed algorithm, it is compared with PAM, fast algorithm for k-medoids clustering (FK) [10] and GCK on Iris and Wine dataset. Each algorithm runs 20 times. The characteristics of the two datasets are given in Table I. The average accuracy and iteration of above algorithms are shown in Table II.

Table I. Characteristics of datasets considered

Name of Dataset	No. of Clusters	No. of Features	Size of Dataset (Size of Clusters in Parentheses)
Iris	3	4	150 (50, 50, 50)
Wine	3	13	178(59, 71, 48)

From the results in Table II, it can be seen that the clustering accuracy of the proposed algorithm on both datasets are higher than other algorithms, especially on Iris dataset. The improvement percentages of the clustering's average accuracy, obtained from our method is up to 24.69% and 8.26% higher respectively when comparing to PAM and FK. Moreover, the iteration is minimal when comparing to three other algorithms. Comprehensive experimental results, the proposed algorithm has higher accuracy, faster global optimal convergence speed and more stable performance.

Table II. Results obtained by the algorithms runs on Iris and Wine dataset

Method	Iris Dataset		Wine Dataset	
	Accuracy (%)	Average Iteration	Accuracy (%)	Average Iteration
PAM	77.56	5.03	52.87	9.13
FK	89.33	4.21	70.79	5.62
GCK	90.00	3.83	70.79	4.36
The Proposed Algorithm	96.71	3.27	75.03	3.92

Conclusions

This paper proposes an improved artificial bee colony algorithm which is used for k-medoids clustering optimization. By optimizing initial colony, dynamically adjust search step, using selection probability based on sorting, the proposed algorithm can further accelerate the convergence speed and avoid premature convergence. Through simulation experiment, the proposed algorithm is compared with PAM, FK and GCK, which outperforms to the considered algorithms in terms of initialization, convergence speed, accuracy and stability. However, how to reduce the computational complexity of the algorithm and adaptively produce related parameters will be the next focus of the study.

Acknowledgements

This work was supported by National Natural Science Foundation of China (11171095, 71371065), Funding Project for Hunan Science & Technology Program (2013SK3146) and Natural Science Hengyang Mutual Fund of Hunan Province (10JJ8008).

References

[1] J. Han, M. Kamber. J. Pei, Data Mining: concepts and techniques, Beijing: China Machine Press, 2012.

[2] K. Luo, L. Li, B. X. Zhou, “A Honey-Bee Mating Optimization Clustering Algorithm,” Acta Electronica Sinica, 2014, Vol. 42, No. 12, pp. 2435–2441. In Chinese.

[3] A. Banharnsakun, T. Achalakul, B. Sirinaovakul, “The best-so-far selection in Artificial Bee Colony algorithm,” Applied Soft Computing, Nov. 2011, pp. 2888–2901.

[4] N. X. Xia, Y. D. Su, X. Qin, “Efficient k-medoids clustering algorithm,” Application Research of Computers, 2010, Vol. 27, No. 12, pp. 4517–4519. In Chinese.

[5] Q. Ma, J. Y. Xie, “New k-medoids clustering algorithm based on granular computing,” Journal of Computer Applications, 2012, Vol. 32, No. 7, pp. 1973–1977. In Chinese.

[6] W. F. Gao, S. Y. Liu, “A modified artificial bee colony algorithm,” Computers & Operations Research, Mar. 2012, Vol. 39, pp. 687–697.

[7] H. Wang, “Improved artificial bee colony algorithm,” Computer Engineering and Design, 2011, Vol. 32, No. 11, pp. 3869–3872. In Chinese.

[8] L. Xu, S. F. Ding, “Research on Granularity Clustering Algorithms,” Computer Science, 2011, Vol. 38, No. 8, pp. 25–28.

[9] Z. W. Xiong, K. Luo, “Clustering algorithm based on improved simplified particle swarm optimization,” Application Research of Computers, 2014, Vol. 31, No. 12, pp. 3550–3552.

[10] H. S. Park, C. H. Jun, “A simple and fast algorithm for k-medoids clustering,” Expert Systems with Applications, 2008, Vol. 36, No. 2, pp. 3336–3341.