A Modified Clustering Algorithm Based on Swarm Intelligence[1]

Lei Zhang[1], Qixin Cao[1], and Jay Lee[2]

[1] State Key Laboratory of Vibration, Shock & Noise,
Shanghai Jiao Tong Univ., 200030, Shanghai, China
{Lei Zhang, zhanglei75}@sina.com,
{Qixin Cao, qxcao}@sjtu.edu.cn
[2] NSF I/UCR Center for Intelligent Maintenance Systems,
Univ. of Cincinnati, OH 45221, USA
{Jay Lee, jay, lee}@uc.edu

Abstract. A modified clustering algorithm based on swarm intelligence (MSIC) is proposed in this paper. To improve the running efficiency of the SIC algorithm, the random projection of the patterns into the plane is modified. The patterns are firstly analyzed by principal component analysis (PCA) and the first two principal components (PCs) are retained. The patterns are projected into the plane according to their corresponding PCs, which are processed as the projection coordinates. This modification ensures that the pattern will be similar to the ones in its local surroundings and the rough clustering has been formed at the beginning time of the algorithm. Moreover, to reduce the influence of the parameters on the algorithm, a simple way to calculate the swarm similarity of the pattern is presented. The adjusting formula of the similarity threshold is also proposed. Finally, the modified algorithm is compared with the original one and the results prove the efficiency has been improved significantly.

1 Introduction

Swarm Intelligence is one kind of intelligent behavior shown by the cooperation of collective insects, such as ants and bees. Swarm Intelligence Clustering (SIC) is a clustering algorithm imitating the behavior of ants. Researchers have found that some ants can pick up the dead bodies randomly distributed in the nests and group them with different sizes. The large group of bodies attracts the ant workers to deposit more dead bodies and becomes larger and larger. The essence of this phenomenon is a positive feedback [1]. Deneubourg etc [2] gave the basic model to explain it, which was called BM model. Lumer and Faieta [3] extended the model and applied it in data analysis. In their analysis, the data object with n attributes can be looked as a point in n dimensional space. The point in the R^n space is projected into a low dimensional space (often a two dimensional plane). The similarity of the data object with other

[1] Supported by the National Natural Science Foundations of China under grant No. 50128504 and No.60443007.

L. Wang, K. Chen, and Y.S. Ong (Eds.): ICNC 2005, LNCS 3612, pp. 535 – 542, 2005.

ones in local surroundings is calculated to decide whether the object should be picked up or dropped. Wu[4] studied the SIC algorithm systematically. He defined some important concepts such as swarm similarity, similarity coefficient and probability conversion function. He also suggested a more simple probability conversion function to reduce the dependence of the algorithm on the parameters.

Compared with other clustering algorithms, such as k-means clustering, the SIC algorithm can find the number of clustering centers self-organizationally. The visualization and robustness of the algorithm are also very distinct. The parallel property built in the algorithm makes it very suitable to the clustering of big data sets. But the SIC algorithm also has some disadvantages. For example, its running efficiency is not high and there are no theories to guide the selections of the parameters [5]. To solve these problems, a modified SCI algorithm is proposed in this paper. First, Principal Component Analysis (PCA) is suggested to reduce the randomicity when the patterns are projected into the plane. Namely, the patterns are processed by PCA, then the first two principal components (PCs) are retained and processed as the pattern's projection coordinates. This pro-processing ensures that the patterns close in the R^n space are also close in the projection plane. So the pattern is of high similarity with others in its local surroundings at the beginning of the algorithm. Moreover, the similarity of the pattern is calculated by a more simple way, which reduces the influence of the parameters on the algorithm.

The paper is organized as follows. The SIC algorithm is introduced in section 2. Then the modified SIC algorithm is proposed in section 3. The simulation and the comparison of SIC and MSIC algorithm are shown in section 4. Finally the conclusion is given in section 5.

2 The Swarm Intelligence Clustering Algorithm (SIC)

Some important concepts in the SIC algorithm are firstly introduced as follows [4,5].

Local surroundings: it is a neighboring region of one pattern, which is often a circle region. The center of the circle is the point of the pattern's coordinates and the radius is *r*.

Swarm similarity: the integrated similarity of the pattern with other patterns in its local surroundings. The similarity is usually measured by the distance between the patterns.

Probability conversion function: it is a function that converts the swarm similarity into the probability of picking up or dropping the pattern by the ant.

In the SIC algorithm, the patterns which are going to be clustered are projected into a two-dimensional plane randomly. Then the ant calculates the swarm similarity of the pattern with others in its local surroundings. And the swarm similarity will be turned into the probability to pick up or drop the pattern through the probability conversion function. The patterns can be clustered after many cycles via the actions of the ant swarm.

The swarm similarity is calculated by the following formula,

$$f(O_i) = \sum_{O_j \in Neigh(r)} [1 - \frac{d(O_i, O_j)}{\alpha}] \tag{1}$$

Where, *Neigh*(*r*) represents the local surroundings of the pattern O_i, which is a circle region with the radius *r*. $d(O_i, O_j)$ is the distance between the pattern O_i and O_j, and usually Euclidean distance is preferred. α is the swarm similarity coefficient, which has an important influence on the number of the clustering centers and the speed of the algorithm.

The probability conversion function converts the swarm similarity into the probability of picking up or dropping the patterns by the ant. In the reference [4], a more simple probability conversion function than that in BM was applied, which is shown in the formula (2) and (3).

$$P_p = \begin{cases} 1-\varepsilon & f(O_i) \le 0 \\ 1-k\times f(O_i) & 0 < f(O_i) \le 1/k \\ 0+\varepsilon & f(O_i) \ge 1/k \end{cases} \tag{2}$$

$$P_d = 1 - P_p \tag{3}$$

Where, P_p, P_d are the probabilities of picking up and dropping the pattern respectively. ε is a little real number. P_p, P_d are compared with the threshold P_r to decide whether the patter should be picked up or dropped. Only the parameter *k* is needed to choose properly after the simplicity rather than k_1 and k_2 in the BM algorithm[2]. So the parameters are simplified. But there is still no theoretical guidance to determine the value of *k* in the practical application.

The process of SIC algorithm can be referred the document [5] and [6]. The terminating condition of the algorithm has two cases: one is up to the maximum cycle times, the other is no pattern is moved again. The former one is usually applied because the latter is complex in computation.

3 The Modified Swarm Intelligence Clustering Algorithm (MSIC)

To improve the efficiency and simplify the parameters of the SIC algorithm, a modified algorithm is proposed as follows.

3.1 Modifying the Random Projection of the Patterns Based on PCA

At the beginning time of SIC algorithm, the patterns are projected into the plane randomly and one pattern is corresponded with a pair of coordinates. Because the coordinates is randomly selected, the similarity of the pattern with the ones in its local surroundings is very low. This will induce that the pattern is easily picked up but not easily dropped by the ant. Therefore, it will take a long time from the beginning to the time when the pattern is similar to the ones near it.

How to keep the patterns close after the projection if they are close in the R^n space? We suggest that the patterns should be pre-processed by principal component analysis (PCA). Then the first two principal components (PCs) are retained and processed. According to the principles of PCA [7], the first two PCs can remain the most information of original patterns. If the patterns are projected corresponding with the coordinates composed by the processed two PCs, it will be ensured that the patterns near in the R^n space will be near in the projection plane. As a result, the rough clustering of the patterns has been formed at the beginning time of the modified algorithm. This result is the similar as that of the SIC algorithm after many cycles, so the running time is reduced significantly.

How to process the PCs will be introduced detailed in sections 4.

3.2 Modifying the Formula of Swarm Similarity

The similarity of the pattern is computed as the formula (1) in the SIC algorithm, where the similarity coefficient will influence on the number of the clustering centers as well as the speed of the algorithm. If α is too large, the patterns which are not similar will be clustered together. If α is too small, the patterns which are similar will be clustered into different groups. In the reference [4], Wu suggested that α should be changed with the cycles increasing. But there is no theory to guide how to change it. The change of α is various at different applications, so it is difficult to determine it properly. In addition, the probability conversion function is calculated as the formula (2) and (3), where the parameter *k* has an important influence on the probability. But how to select *k* is also a problem.

To avoid the influence of α and *k* on the clustering results, a more simple similarity computing method is presented. From the formula (1), it can be seen that the essence to measuring the similarity is the distance between the patterns. Therefore, the similarity is represented directly by the distance between the patterns in this paper. The similarity of the pattern O_i with others is.

$$f(O_i) = \frac{1}{n} \sum_{O_j \in Neigh(r)} d(O_i, O_j) \tag{4}$$

Where, *n* is the number of the patterns in the local surroundings of the pattern O_i. The means of other signs are the same as those in formula (1). The larger $f(O_i)$ is, the smaller the pattern O_i ‘s similarity is.

Dissimilar to the SIC algorithm, the MSIC algorithm doesn’t apply the probability conversion function. The threshold of the similarity *F* is set, and $f(O_i)$ is compared with *F* to determine whether the pattern is picked up or dropped. This simple way is easily computed and avoiding the influence of *k* on the algorithm. Because the distances of the patterns are large at the beginning time of the clustering, *F* should be set a large value. With the increasing of the cycles, $f(O_i)$ will be decreased, so *F* should be reduced correspondingly. The adjusting formula of *F* in the MSIC algorithm is

$$F(t)=\begin{cases}F(t) & \mathit{if}\ \mathrm{mod}(t,500)\neq 0\\ kF(t) & \mathit{otherwise}\end{cases} \tag{5}$$

Where, k is a real number smaller than 1. t is the number of cycles The formula (5) means $F(t)$ will be reduced per 500 cycles. “500 cycles” is a relative concept, which influences the reducing speed of $F(t)$ as well as k. These two parameters can be adjusted according to the changing speed of the similarity.

3.3 The Process of MSIC Algorithm

The detailed process of the MSIC algorithm is as follows.

Algorithm: the MSIC algorithm
Inputs: The patterns waiting to be clustered
Outputs: The clustered patterns or the centers of the clustered groups

```
Process: 1: The initialization of all parameters: cycle_number (the maximum cycle times); ant_number (the number of the ants), the radius $r$; the initial threshold of the similarity $F(1)$; the adjusting parameter of the similarity threshold $k$.
         2: The patterns are processed by PCA, and the first two PCs are retained. Then the PCs are processed as the coordinates and the patterns are projected into the plane according to its processed PCs.
         3: Set the initial patterns and set the coordinates of the patterns to the ants. The initial load states of the ants are without any load.
         4: for $i$=1:cycle_number
             4.1 for $j$=1:ant_number
                 4.1.1 take the coordinates of the ant as the center, $r$ as the radius, calculate the similarity $f$ of the ant's pattern in its local surroundings by the formula (4)
                 4.1.2 if load_ant($j$)=0, compare $f$ with the threshold $F(i)$. If $f \leq F(i)$, the ant picks up the pattern and the ant's load is set to 1, namely load_ant($j$)=1. Otherwise, the ant doesn't pick up the pattern and new pattern and corresponding coordinates are set to the ant.
                 4.1.3 If load_ant($j$)=1, compare $f$ with the threshold $F(i)$. If $f>F(i)$, the ant drops the pattern and the current coordinates of the ant are set to the pattern. load_ant($j$)=0. Then a new pattern and its coordinates are set to the ant randomly. Otherwise, the ant doesn't drop the pattern and new coordinates are set to the ant again.
             End $j$ ( up to the maximum number of the ants)
             4.2 Calculate the threshold F($i$) by the formula (5)
           End $i$ ( up to the maximum number of the cycles)
         5: Calculate the clustering center of all groups and output the clustered
```

4 The Comparison of MSIC and SIC Algorithms

To compare the performance of SIC and MSIC algorithms, the data in the reference [4] is analyzed by the MSIC algorithm. The experiment data is from the machine learning database of the website http://www.ics.uci.edu/~mlearn/MLRepository.html. It is the data of the iris. The number of the items in the data set is 150 and each item has 4 attributes. The number of classes is 3.

The parameters in two algorithms are listed in the table 1. It can be seen that the MSIC algorithm avoids the influence of the parameters α and k, which may be selected improperly.

Table 1. The comparison of two algorithms

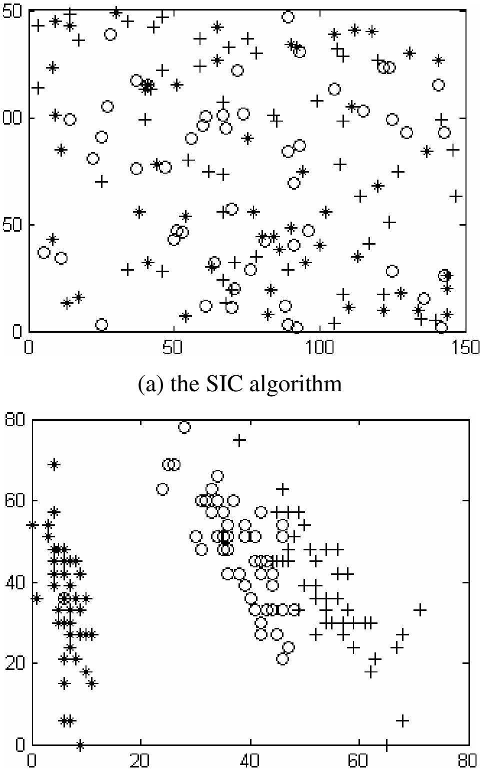
Parameters	SIC algorithm	MSIC algorithm
The maximum cycle times	60000	10000
The number of ants:	6	6
The size of the projection plane	480×450	80×80
The radius	r=20	r=5
The similarity coefficient	α =0.4-0.3	\
Others	k=0.1 (in the formula 2)	k= 0.95 (in the formula 5)
Others	P_r （not mentioned)	F(1)=1.8

In the MSIC algorithm, the patterns are firstly analyzed by PCA and the first two PCs are retained. The two PCs remain about 97.76 percent of the information of original data. Then the PCs are processed as follows:

(1) Enlarging: Because the PCs are very small, they are multiplied by 10 to be distinguished easily.
(2) Rounding
(3) Shifting: Finding the minimums of the fist PC and the second PC respectively. Subtracting the minimums from the pair of PCs and the last processed values are obtained.

The two PCs after the processing are taken as the projection coordinates of the pattern (The first PC as x-coordinate and the second PC as y-coordinate). The aim of these processing is making the coordinates of the patterns distributed in the first quadrant and identified easily.

The projections of the patterns at the beginning of the algorithms are shown in the Fig. 1. We apply three signs to identify different classes. It can be seen from (a) in Fig.1, the patterns are randomly distributed in the projection plane in the SIC algorithm. While in (b) of Fig. 1, some patterns have been divided from others after they are projected according to their processed PCs. Especially the patterns of the class one (signed as *) are distinct from other two classes. The projection way in the MSIC algorithm ensures that the rough clustering has been formed at the beginning of the algorithm, which is similar as the result of the SIC algorithm after several hundreds or thousands of cycles.



(a) the SIC algorithm

(b) the MSIC algorithm

Fig. 1. The projection of the patterns in two algorithms

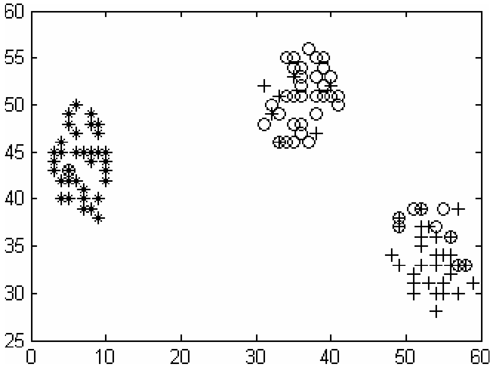


Fig. 2. The clustering result of the MSIC algorithm

In this sample, the MSIC algorithm will cluster the patterns into three classes after about 8000-10000 cycles while the SIC algorithm needs 60000 cycles. If the parameters in the two algorithms, such as the number of the ants, are the same, the running efficiency of the algorithm is dependent mainly on the number of cycles. So the running time of the MSIC algorithm is reduced. The clustering results of the MSIC algorithm are shown in the Fig 2. The average accuracy is 90.3 percent which has no big difference with 90 percent of the SIC algorithm in the reference [4].

5 Conclusions

This paper focuses on a modified clustering algorithm based on swarm intelligence (MSIC). To improve the efficiency of the SIC algorithm, PCA is suggested to reduce the randomicity when the patterns are projected into the plane. The first two PCs of the pattern are processed as the corresponding projection coordinates. This projection way ensures the running time of the algorithm can be reduced because the rough clustering has been formed. Moreover, a simple way to calculate the similarity based on the distance between the patterns is presented and the adjusting formula of the similarity threshold is given. The comparison results of the MSIC algorithm and the SIC algorithm prove the running efficiency of the MSIC algorithm is improved. In the MSIC algorithm, how to measure the swarm similarity more properly and how to adjust the similarity threshold need to be further studied.

References

1. Marco D., Eric B., Guy T. Ant Algorithms and Stigmercy. Future Generation Computer System, 16(2000)851-871
2. Deneubourg J. L., Goss S., Frank N., etc. The Dynamics of Collective Sorting: Robot-like Ants and Ant-like Robots. In: Proceedings of the 1st International Conference on Simulation of Adaptive Behavior: From Animals to Animats. MIT Press/Bradford Books, Cambridge, MA, (1991)356-363
3. Lumer E., Faieta B. Diversity and Adaptation in Populations of Clustering Ants. In: Processing of the 3rd International Conference on Simulation of Adaptive Behavior: From Animals to Animats. MIT Press/Bradford Books, Cambridge, MA, (1994)501-508
4. Bin W. Research on Swarm Intelligence and Its Application in Knowledge Discovery. Ph.D. Thesis, Institute of Computing Technology, Chinese Academy of Science, Beijing. (2002) 40-47
5. Bin W., Yi Z., Wei-peng F., etc. A Customer Behavior Analysis Algorithm Based on Swarm Intelligence. Chinese Journal of Computers, Vol.26 No.8 (2003)913□918
6. Bin W., Zhong-zhi S. A Clustering Algorithm Based on Swarm Intelligence. International Conference on Info-tech and Info-net (ICII 2001), Beijing, 29 Oct.-1 Nov. Vol.3 (2001)58-66
7. Partridge M., Rafael A.C. Fast Dimensionality Reduction and Simple PCA. Intelligent Data Analysis. 2(1998)203-210