Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005

DATA CLUSTERING BY ANT COLONY ON A DIGRAPH

LING CHEN[12], LI TU[1], HONG-JIAN CHEN[1]

[1] Department of Computer Science, Yangzhou University, Yangzhou 225009, China
[2] National Key Lab of Novel Software Tech, Nanjing University, Nanjing 210093, China
E-MAIL: lchen@yzcn.net

Abstract:

An adaptive data clustering algorithm based on ant colony (Ant-Cluster) is presented. Enlightened by the self-organizing behavior of ant society, we assign acceptance rates on the directed edges of a pheromone digraph in Ant-Cluster system. The pheromone on the edges of the digraph is adaptively updated by the ants passing it. Some edges with less pheromone are progressively removed under a list of certain thresholds in the process. Strong connected components of the final digraph are extracted as clusters. The performance of Ant-Cluster is compared with classical K-means clustering algorithm and ACO clustering algorithm LF in terms of clustering quality and efficiency on several real datasets and clustering benchmarks. Experimental results indicate that the Ant-Cluster is able to find clusters faster with better clustering quality and is easier to implement than K-means and LF.

Keywords:

Digraph; clustering; ant colony; K-means; LF

1. Introduction

Clustering aims to discover sensible organization of objects in a given dataset by identifying similarities as well as dissimilarities between objects. It classifies a mass of data, without any prior knowledge, into clusters which are clear in space partition outside and highly similar inside. Cluster analysis has found many extensive applications including classification of coals[1], toxicity testing[2], discovering of clusters in DNA nucleotides [3], etc.

Recently, inspired by the swarm intelligence [4-5] shown through the social insects' (e.g. birds, bee, fish, ants etc.) self-organizing behavior, researchers create a new type of artificial ants to imitate the nature ants' behaviors, and named it artificial ant colony system. By simulating the ants' swarm intelligence, M.Dorigo et al. first advanced the ant colony optimization algorithm (ACO)[6-8] to solve several discrete optimization problems. In ACO, artificial ants are created to emulate the real ants in the process of seeking food and information exchanging. The successful simulation has been applied to traveling salesman problem (TSP) [9], system fault detecting [10], sequential ordering[11], and other combinational optimization problems [12-13].

Researchers also have applied artificial ant colony to data clustering by simulating the behavior of ants' corpses piling. Deneubourg et al advanced a basic model (BM)[14] to explain the ants' corpses piling phenomenon and presented a clustering algorithm based on this model. By modifying BM algorithm, Lumer and Faieta [15] presented a formula to measure the similarity between two data objects and designed the LF algorithm for data clustering. In BM and LF, the ants consume large amount of computation time and memory space since the data items cannot move directly and efficiently.

In ants' process of seeking their fellows, they will also volatilize a kind of chemical odor. Based on this kind of odor, ant *i* may exclude or attract ant *j*. Enlightened by this fact, we propose a novel adaptive data clustering algorithm by ant colony with a digraph (Ant-Cluster). In this digraph, the vertexes represent the data to be clustered and the weighted edges between vertexes represent the acceptance rate between the two data it connected. The artificial ants travel on the digraph and deposit pheromone on the edges they passed. The pheromone on each edge of the digraph will be updated according to the artificial ants' adaptive movements. The more similar data objects are, the higher the quantity of pheromone may be deposited on the edge between them. In the process of the algorithm, we progressively omit some invalid connections whose pheromone value is less than a certain threshold. The strong connected components of the final digraph form the results of data clusters. Some adaptive strategies are also presented to speed up the clustering greatly. Experimental results indicate that the Ant-Cluster is able to find clusters faster with better clustering quality and is easier to implement than K-means and LF.

The rest of this paper is organized as follows: section 2 describes the main ideas of ACO. Details of Ant-Cluster are illustrated in section.3. Experimental results are depicted

0-7803-9091-1/05/$20.00 ©2005 IEEE

in section 4, while section 5 concludes the paper.

2. Ant colony optimization

In ACO, the chemical substance for communicating information by individual real ant is called pheromone laid by artificial ants. The pheromone trails are reinforced by other artificial ants and validated with time. In the sense, ACO has been applied successfully to a range of different combinatorial optimization problems recently.

The main ideas of ACO algorithm are as follows:

- In front of two or more paths, an artificial ant has to select one path. The larger amount of pheromone the path has, the more probability it has to be chosen.
- When an artificial ant follows a path, the amount of the pheromone deposited on the path is incremented by the quality proportion of the candidate solution.
- In every iteration, each path followed by an ant will be a candidate solution for a given problem.

According to these rules, the ants gradually converge to a shorter and shorter path. Eventually, the optimum or near-optimum solution for the target problem will be found.

In the details of ACO algorithm designation, several parameters of functions will be used:

- $\tau_{ij}(t)$: represents the amount of pheromone between stage i and j at t time,
- η_{ij} : a problem-dependent heuristic function that measures the quality of local path (i,j);
- $p_{ij}^{k}(t)$: the transition probability for ant k to select the path (i,j) at time t, which depends on the amount of pheromone on trail (i,j) ($\tau_{ij}(t)$) and the value of heuristic function (η_{ij}).
- α , β : two parameters which decide respectively the effect proportion of $\tau_{ij}(t)$ and η_{ij} on ants' selecting path.

Based on these parameters and rules above, the ACO algorithm could find the optimum solution corresponding to the given problem by artificial ants. The special formulas are different with the different target problems.

3. Data clustering by ant colony on a digraph

In the algorithm Ant-Cluster, a weighted digraph is built where the vertexes represent the data to be clustered and the weight of the edge between vertexes is the acceptance rate between the two data it connected. The ants travel in the digraph and update the pheromone on the paths it passed. The digraph is modified by gradually omitting some edges whose pheromone values are less than a threshold. At last, the strong connected components of the updated digraph are computed to form the data clusters.

3.1. The framework of the algorithm Ant-Cluster

The framework of the proposed algorithm Ant-Cluster is as follows.

Algorithm: Ant-Cluster

1. initialize parameters: $\{g_0,g_1...g_h\}$, h=1;
2. initialize the pheromone digraph (V,E);
3. place ants at randomly selected data sites;
4. while (*iter-num*<*maxnum*) do // *maxnum*=500
5. for each ant k do
6. while not ($\forall v \in V$ have been visited) do
7. select the next edge in E to visit according to probability function p ;
8. end do
9. end for
10. update the pheromone on edges in E;
11. if (*iter-num*%10==0) then
12. for each edge(i,j) in E do
13. if $\tau_{ij} < g_h$ remove (i,j) from E;
14. end for
15. adaptively update the value of α, β ;
16. h=h+1;
17. end if
18. *iter-num* = *iter-num* +1;
19. end do
20. compute the strong connected components of the final digraph to form the clusters.

3.2. The acceptance rate between data objects

Line 2 in the algorithm constructs a weighted digraph where the vertexes represent the data to be clustered and the weighted edges between vertexes represent the acceptance rate between the two data it connected. The acceptance rate can be computed from the similarity between the data

objects.

Definition 1: The set of data items

We use $S = (O, A)$ to denote a set of n data items here

- $O = \{O_1, O_2, ...O_n\}$ represents the set data objects,
- $A = \{A_1, A_2, ...A_r\}$ represents the attributes of data objects, where
- $\forall i, i \in (1,2,...n)$, $\exists a_{ik}, k \in (1,2,...r)$ denotes the attribute A_k of O_i , therefore A_k could be denoted as a r-dimensional vector （a_{i1}, a_{i2}, ...a_{ir}） , $i \in \{1,2,...n\}$.

Definition 2: The difference between data items

The difference between two data items O_i and O_j is defined as:

$$dif\ (i, j) = \sum_{k=1}^{r} \left| a_{ik} - a_{jk} \right|, i, j = 1,2,...n \qquad (3.1)$$

Definition 3: The similarity between data items

For two data items O_i and O_j , we use $Sim(i, j)$ to denote their similarity:

$$Sim(i, j) = 1 - \frac{dif\,(i, j)}{\max dif} \qquad (3.2)$$

Here, $\max dif = \max_{1 \le i, j \le n} dif\,(i, j)$ denotes the largest difference among the data items.

We use $avesim(i)$ and $\max sim(i)$ to denote the average and maximum similarity of O_i with all the other data items:

$$avesim(i) = \frac{1}{n} \sum_{j=1}^{n} Sim(i, j) \qquad (3.3)$$

$$\max sim(i) = \max_{1 \le j \le n} Sim(i, j) \qquad (3.4)$$

Definition 4: The acceptance rate

We use *accept*(*i*,*j*) to denote the acceptance rate of data item O_i to O_j :

$$accept\,(i, j) = Sim(i, j) - \frac{1}{2}(avesim\,(i) + \max\ sim\,(i)) \qquad (3.5)$$

We can see that acceptance rate between two data objects is not symmetric, namely, *accept*(*i*,*j*) is normally not equal to *accept*(*j*,*i*). The more similar two data objects are, the greater acceptance rates to each other will be.

Using *accept*(*i*,*j*) as the weight of the edge (*i*,*j*), we can form a weighed digraph as the initial pheromone digraph. Denote the weight of the directed edge from the vertexes representing data items O_i and O_j as $\tau_{ij}(0)$. This value will be updated in every step of clustering by the pheromone deposited by the ants passing this edge.

The value of $accept(i, j)$ may be below zero, that means O_i rejects O_j. In this case, we treat this edge as a invalid one and it will not be included in E. The set of valid edges E will be updated in the process of the algorithm since some of the edges with pheromone less than a certain threshold could be deleted from the graph gradually.

In Ant-Cluster, the proposed initial pheromone values are set on the edges of the digraph, while in the traditional ant colony algorithm initial values of the pheromone at all edges are set as zero. This initial pheromone value of Ant-Cluster is much important for ants' latter movements and the efficiency of algorithm's execution. It will help to update this pheromone digraph for the final clustering.

3.3. The probability function

In Ant-Cluster, artificial ants are used for visiting data items represented by the vertexes in the digraph. A certain probability function is computed for the ant_k to select the next vertex, namely the next data object, to move to. To select the most similar data item, the ant on data item i should select the next data item j according to the following formula:

$$j = \begin{cases} \arg\left\{ \max_{u \in allowed_k} [\tau_{iu}^{\alpha}(t)\eta_{iu}^{\beta}] \right\}, & when\ q \le q_0 \\ selected\ by\ probability\ p_{ij}^{k}(t) & otherwise \end{cases} \qquad (3.6)$$

where $allowed_k$ is the set of vertexes can be chosen by the k-th ant, constant q_0 is a threshold for the vertex connected by the edge with the largest amount of pheromone to be chosen. In each iteration, a random number $q \in [0,1]$ is generated and compared with q_0. When $q > q_0$, data item j is chosen by the probability function defined as (3.7), otherwise the ant selects the vertex connected by the edge with the largest amount of pheromone.

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{r \in allowed_k} \tau_{ir}^{\alpha}(t)\eta_{ir}^{\beta}(t)} & j \in allowed_k \\ 0 & otherwise \end{cases} \qquad (3.7)$$

Here, we adopt the common probability formula form in ACO, but an adaptive strategy is applied on α and β which will be introduced in section 3.5.

3.4. Heuristic function and pheromone updating

In (3.7), η_{ij} is the heuristic function which reflects the preference of the edge (i,j) to be selected by the ants. Obviously, the more similarity between the two data objects connected, the more preference the edge should have. Therefore, η_{ij} is defined as:

$$\eta_{ij} = Sim(i, j) \tag{3.8}$$

Different from pheromone τ_{ij}, η_{ij} is a static and unidirectional heuristic information.

After each iteration, line 10 in Ant-Cluster updates the pheromone of edges on the paths the ants just passed through according to the following formula:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \sum_{k=1}^{m}\Delta\tau_{ij}^{k} \tag{3.9}$$

In (3.9), $\rho \in (0,1)$ is a constant called coefficient of evaporation. At each iteration the pheromone on each path will be evaporated by a rate of ρ. The increment of τ_{ij} by ant k is denoted as $\Delta\tau_{ij}^{k}$ and can be computed by (3.10) where Q is a constant.

$$\Delta\tau_{ij}^{k} = \begin{cases} Q.Sim(i,j), if\ ant\ k\ passes\ \ path(i,j) \\ 0 \qquad\qquad\quad , otherwise \end{cases} \tag{3.10}$$

It can easily be seen from (3.10) that the more ants pass through an edge, the more pheromone deposited on it, and it will have more probability to be included in the same strong connected component of the weighted digraph constructed in the last part of the algorithm, and hence the two data items connected by this edge are more likely to be classified into one cluster.

3.5. Adaptively update of the parameters α, β

In (3.7), α,β determine the relative influence of the trail strength τ_{ij} and the heuristic information η_{ij}. At the initial stage of the algorithm, the pheromone value on each edge is relatively small. To speedup the convergence, the ants should select the path mainly according to the heuristic information η_{ij}. Therefore, the value of β should be relatively large. After some iterations, the pheromone values on the edges are increased, their influence become more and more important. Therefore the value of α should be relatively large. Since the adjustment of the values of α and β should be based on the distribution of pheromone on the edges, in (3.11) we define τ_{ave} as the average amount of pheromone on the pheromone digraph and in (3.12) define ψ as the pheromone distributing weight to measure the distribution of pheromone on the graph.

$$\tau_{ave} = \frac{\sum_{(i,j)\in E}\tau_{ij}}{|E|} \tag{3.11}$$

$$\psi = \frac{1}{|E|}\left[\sum_{(i,j)\in E}(\tau_{ave} - \tau_{ij})^2\right]^{\frac{1}{2}} \tag{3.12}$$

Here E is the set of valid edges in the digraph.

Using the pheromone distributing weight ψ, the algorithm updates the value of α,β as follows:

$$\alpha = e^{-\psi}, \beta = \frac{1}{\alpha} \tag{3.13}$$

The algorithm can accelerate the convergence and also can avoid local convergence and precocity by adjusting the value of α,β adaptively. Furthermore, since the amount of pheromone is an important measure for data clustering, the pheromone distributing weight ψ is also a critical factor to terminate the iterations of the algorithm.

4. Experimental results

The algorithm Ant-Cluster is tested on Windows XP, P1.7G, Matlab 6.0 with the basic parameters set as $m=n/2$, $c=1$, $\rho=0.05$, $q_0=0.95$. We not only test on the ant-based clustering data benchmark but also on several real datasets. To compare the performance of our method with that of other clustering algorithms, we also test on these datasets using the K-means and LF algorithm.

4.1. Test on ant based clustering data benchmark

We test a dataset with five data types each of which consists of 10 two-dimensional data (x,y) which belong to five classes as shown in Figure 1. The five types of data (x,y) are $[N(0.2,0.1^2),N(0.2,0.1^2)]$,

$[N(0.5,0.1^2),N(0.5,0.1^2)],[N(0.8,0.1^2),N(0.2,0.1^2)],[N(0.2,0.1^2),N(0.8,0.1^2)],[N(0.8,0.1^2),N(0.8,0.1^2)]$ respectively which obey normal distribution $N(u,\sigma^2)$.

The initial pheromone digraph of this dataset of 50 data items is shown in Figure 2. Figure 3 shows the modified pheromone digraph obtained after 60 iterations of Ant-Cluster.

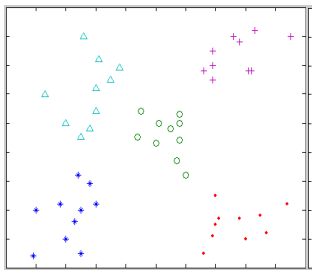

Figure 1. The initial distribution of datasets

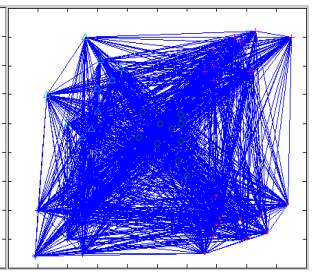

Figure 2. The initial pheromone digraph

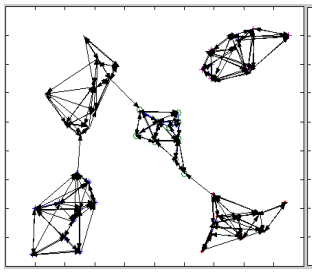

Figure 3. The transferred pheromone digraph after 60 iterations

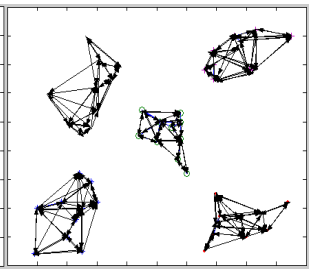

Figure 4. The final clusters(strong connected components)

Five strong connected components of the transferred digraph are found as shown in Figure 4. The finial clusters are represented by these strong connected components .

Two more datasets which have 300 and 600 data items are also tested to compare the performance of Ant-Cluster with LF and K-means. Each dataset has two attributes and three classes where data items are generated at random in distributions of [N(0.3,0.12),N(0.3,0.12)], [N(0.8，0.12),N(0.3,0.12)], [N(0.5，0.12),N(0.8,0.12)] respectively. In the dataset with 600 data items, there are little rough boundary between classes. Table 1 shows the average results of 50 trails on these two types of data using Ant-Cluster, LF and the classical K-means algorithm.

Table 1. The results of three algorithms on artificial studies

Clustering Algorithms	300 data items		600 data items	
	Error rate	Time cost(s)	Error rate	Time cost(s)
LF	0	135.08	1.98%	322.55
K-Means	0	112.30	2.57%	243.12
Ant-Cluster	0	53.27	0.38%	101.73

It is can be easily seen from Table 1 that although all the algorithms can successfully get clusters on 300 data items without error, Ant-Cluster costs much less computational time than K-means and LF algorithm. For the test on 600 data items, Ant-Cluster is much better than the other two methods in terms of error rate and time cost. The reason is that LF algorithm spends much time for the ants to search for the proper location to pickup or drop the data objects.

4.2. Test on real databases benchmarks

In addition, we also test on the real databases benchmarks of Glass and Soybean(small) using Ant-Cluster, LF and K-means. Glasses dataset consists of 214 examples each with 9 continuous attributes. There are 6 classes of glasses. As a kind of biology dataset, Soybean is a real dataset benchmark which has 47 data instances and 35 numeric predictive attributes. The dataset contains 4 classes of 10,10,10,17 data objects respectively. Large number of experimental results show that the clustering results of Ant-Cluster after 500 iterations are mostly better than those of LF after 10000 iterations. Results of 50 trials of each algorithm on the two datasets are shown in Table 2 and 3.

Table 2. Comparison of the results on Glass datasets

Parameters	Algorithms		
	K-Means	LF	Ant-Cluster
iterations	k=6	10000	500
number of clusters	6	6	6
max error numbers	17	14	11
min error numbers	9	7	6
average error numbers	12.15	10.31	7.82
average error rate	5.67%	4.68%	3.65%
time cost (s)	92.42	115.54	40.24

Table 3. Comparison of the results on Soybean datasets

Parameters	Algorithms		
	K-Means	LF	Ant-Cluster
iterations	k=4	10000	500
number of clusters	4	4	4
max error numbers	5	7	2
min error numbers	0	1	0
average error numbers	2.48	3.21	0.78
average error rate	5.27%	6.83%	1.66%
time cost (s)	29.37	38.25	9.33

From Table 2 and 3, we can see that Ant-Cluster require less iterations and hence less computation time than K-means and LF. The tables also show the error rates of Ant-Cluster is much lower than that of K-means and LF.

The experimental results clearly show that the clustering quality of Ant-Cluster is much better than the other two algorithms. The reason of low speed of LF algorithm is that it cannot deal with isolated data items efficiently. Since it is very difficult for ants carrying an isolated data object to find a proper position to drop it down, they possibly make long time idle moving which consumes large amount of computational time. With referred to clustering quality, LF lacks adaptive adjustment of parameters and can not speed up the process of clustering. The K-means algorithm has lower speed than Ant-Cluster because it is unsupervised and sensitive to the initialization.

5. Conclusion

In this paper, a novel adaptive data clustering algorithm by ant colony on a digraph is presented. Different from other existing clustering algorithms, Ant-Cluster makes full use of the ant colony system to transform a weighted pheromone digraph and abstracts the strong connected components of the final digraph as the clusters. By analyzing the effect of each parameter used in Ant-Cluster, we proposed effective strategies for the ants selecting the edge, updating the pheromone, and adjusting the parameters adaptively to speed up the clustering procedure and to improve the clustering quality. Compared with K-means and LF algorithm, Ant-Cluster is direct, easy to implement, and self-adaptive. It produces higher quality clusters, and is much more computational efficient than previous methods.

Acknowledgements

This paper is supported in part by the Chinese National Natural Science Foundation under grant No. 60473012, and Chinese National Foundation for Science and Technology Development under contract 2003BA614A-14.

References

[1] Kaufman. L, Pierreux. A, Rousseuw, P., Derde, M.P., Detaecernier, M.R., Massart, D.L., Platbrood, G, "Clustering on a Microcomputer with an Application to the Classification of Coals", Analytica Chimica Acta, 153, pp. 257–260, 1983.

[2] Lawson, R.G., Jurs, P.C, "Cluster Analysis of Acrylates to Guide Sampling for Toxicity Testing", Journal of Chemical Information and Computer Science, Vol 30, no.1 pp. 137–144, 1990

[3] Beckers, M.L.M., Melssen, W.J., Buydens, L.M.C, "A self-organizing feature map for clustering nucleic acids", Application to a data matrix containing A-DNA and B-DNA dinucleotides. Comput. Chem., 21,pp. 377–390, 1997.

[4] Kennedy, J., Eberhart, R.C, "Swarm Intelligence", Morgan Kaufmann Publishers, San Francisco CA, 2001.

[5] Bonabeau, E., Dorigo, M., Théraulaz, G, "Swarm Intelligence: From Natural to Artificial Systems", Santa Fe Institute in the Sciences of the Complexity, Oxford University Press, Oxford New York, 1999.

[6] Dorigo, M., Maniezzo, V., Colomi, A, "Ant system :Optimization by a colony of cooperating agents", IEEE Transactions on Systems, Man and Cybernetics-Part B, vol 26, no.1, pp. 29–41,1996.

[7] Dorigo, M., Gambardella, L.M, "Ant colony system: a cooperative learning approach to the traveling salesman problem", IEEE Trans. On Evolutionary Computation, vol 1, no.1, pp. 53–66, 1997.

[8] Stutzle, T., Hoos, H, "MAX-MIN Ant systems", Future Generation Computer Systems, 16, pp.889–914, 2000.

[9] Dorigo, M., Gambardella, L.M, "Ant colonies for the traveling salesman problem", BioSystems, vol 43,no.2, pp. 73–81, 1997.

[10] Chang, C.S., Tian, L., Wen, F.S, "A new approach to fault section in power systems using Ant System", Electric Power Systems Research, vol 49,no.1, pp. 63–70, 1999.

[11] Gambardella, L.M., Dorigo, M, "HAS-SOP: An Hybrid Ant System for the Sequential Ordering

Problem", Tech. Rep. No. IDSIA 97-11, IDSIA, Lugano Switzerland 1997.

[12] Kuntz, P., Layzell, P., Snyder, D, "A colony of ant-like agents for partitioning in VLSI technology", In: Husbands, P., Harvey, I.(eds.): Proceedings of the Fourth European Conference on Artificial Life, MIT Press, Cambridge MA pp. 412–424, 1997.

[13] Kuntz, P., Snyder, D, "New results on ant-based heuristic for highlighting the organization of large graphs", In: Proceedings of the 1999 Congress or Evolutionary Computation, IEEE Press, Piscataway NJ, pp. 1451–1458, 1999.

[14] Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., Chretien, L, "The Dynamic of Collective Sorting Robot-like Ants and Ant-like Robots", In: Meyer, J.A., Wilson, S.W. (eds.): SAB'90-1st Conf. On Simulation of Adaptive Behavior: From Animals to Animates, MIT press, pp. 356–365, 1991.

[15] Lumer, E., Faieta, B, "Diversity and adaptation in populations of clustering ants", In: Meyer, J.A., Wilson, S.W. (eds.): Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animates, Vol 3. MIT Press/ Bradford Books, Cambridge MA, pp. 501–508, 1994.