Proceedings of International Workshop on Data Mining and Artificial Intelligence (DMAI' 08)
24 December, 2008, Khulna, Bangladesh

Ant Colony Clustering by Expert Ants

Zahra Sadeghi [1], Mohammad Teshnehlab [2]
[1] Computer Engineering Department, Science and Research Branch, Islamic Azad University - Member of Young Researchers Club
[2] Faculty member, Electronic Engineering Department, K.N.Toosi University, Tehran, Iran

[1]z.sadeghi@seiau.ir, [2] teshnehlab@eetd.kntu.ac.ir

Abstract

In this article a new ant clustering algorithm based on case based reasoning (CBR) is presented. Every ant has a case base which is updated iteratively by the process of CBR. The ant which is successful in dropping an item becomes an expert and can use its knowledge for future picked up items. Also expert ants are capable of cooperating to share their knowledge for even better clustering. Our simulation results demonstrated better performance than previous approaches.

Keywords: ant based clustering, ant colony optimization

1. Introduction

Machine learning is divided into two primary sub-fields: supervised learning and unsupervised learning. Within the category of unsupervised learning, one of the primary tools is clustering [1]. Clustering means the act of partitioning an unlabeled dataset into groups of similar objects. Each group, called a `cluster', consists of objects that are similar between themselves and dissimilar to objects of other groups [2]. Swarm Intelligence emerged out of social insect collective behavior shows many interesting properties such as flexibility, robustness, decentralization and self organization [3]. The swarm intelligence clustering models and algorithms have advantages in many aspects, such as no need of priori information, self-organization, flexibility, robustness and decentralization [4]. Ant-based clustering is a heuristic clustering method that draws its inspiration from the behavior of ants in nature [5]. Some of the proposed ant based clustering algorithm can be found in [6-11].

In the present article, we have introduced a clustering algorithm based on expert ants. After each successful dropping, the value and the coordination of the object being dropped is saved in the ant's case base. In the beginning, the case bases of all ants are empty. We have introduced two types of ants: normal ants, which have empty case base and have not dropped an object yet, and expert ants, who were able to drop at least one object. Whenever an expert ant picks up a new load, it searches its case base for the best match and if such a case is found, it will be retrieved and then the ant immediately drops down its load in the nearest neighborhood of the grid coordination of the stored case. Otherwise it uses the drop down function. The case base of all expert ants updates each time they drop down an object. Also expert ants are able to cooperate with each other to find the best place for dropping their load. The results of our approach demonstrated that it can speed up the process of clustering and prevents from producing too many small clusters which is one of the main drawbacks of ant based clustering.

2. The general principle of ant clustering

Ant based clustering is a distributed process. The pioneers of this work are Deneubourg *et al.* [12]. Their model is known as basic model (BM) in which ants are simulated by simple agents that randomly move in an environment which is a square grid. Initially, each data object that represents a multi-dimensional pattern is randomly distributed over the 2-D space. Data items that are scattered within this environment can be picked up, transported and dropped by the agents in a probabilistic way. The picking and dropping operations are influenced by the similarity and density of the data items within the ant's local neighborhood. Deneubourg *et al.* proposed a computational model for spatial sorting. Their model is based on the following probability of picking and dropping functions:

1-4244-2136-7/08/$20.00 ©2008 IEEE

$$Pp = (\frac{Kp}{Kp + f})^2. \quad (1)$$

$$Pd = (\frac{f}{Kd + f})^2. \quad (2)$$

Where Pp is the probability of picking, Pd is the probability of dropping, and Kp and Kd are constants. f is the perceived fraction of items in the neighborhood of the ant. The probability of picking (Pp) is increased if a data object is surrounded by dissimilar data, or when there's no data in its neighborhood. Also, ants tend to drop data in the vicinity of similar ones, so the probability of dropping (Pd) is increased if ants are surrounded with similar data. Deneubourg *et al.*'s model was later extended by Lumer *et al.* [13] to allow its application to exploratory data analysis. They modified the dropping probability functions as the following:

$$Pd(i) = \begin{cases} 2f(i) & if \quad f(i) < Kd \\ 1 & if \quad f(i) \geq Kd. \end{cases} \quad (3)$$

Where Pd is the same as before, i.e., the probability of dropping and picking. In addition, they introduced the density function as the following:

$$f(i) = \max(0.0, \frac{1}{\sigma^2} \sum_{j \in L} (1 - \frac{\delta(i,j)}{\alpha})). \quad (4)$$

Where $\delta(i,j) \in [0,1]$ is the dissimilarity function between two data object i and j. $\alpha \in [0,1]$ is the scaling parameter that permits further adjustments of resulting values. σ^2 is the size of local neighborhood L around the ant's current position. The basic ant clustering algorithm is like the one summarized as bellow [14]:

Let $R \in [0,1]$ be a random number drawn for each use, the basic algorithm is then given by:

Randomly scatter data items on a square grid.

For a number of steps or until a criterion is met, repeat for each ant:

- If the ant is unladen and placed on location l occupied by object o, the object is picked up if $p_{pick\_up}(o) \geq R$.

- If the ant is carrying object o and placed on an empty location l, the object is dropped if $p_{drop\_down}(o) \geq R$.

- Move the ant to a new randomly selected (neighboring) location.

3. Case Based Reasoning (CBR)

CBR is a familiar process that is a normal, intuitive method of decision-making for humans in everyday life. It is a process of considering past cases and arriving at decisions on comparison between the current situation and the old cases [15].

In CBR terminology, a case usually denotes a problem situation. A previously experienced situation, which has been captured and learned in a way that it can be reused in the solving of future problems, is referred to as a past case, previous case, stored case, or retained case. Correspondingly, a new case or unsolved case is the description of a new problem to be solved [16].

CBR is done by considering the similarity between the current situation and old seen situations. The basic idea of CBR for solving a new problem is as follows:

- Remembering a previous similar situation.
- Comparing the new problem to the old solved problems
- Reusing information and knowledge of that situation.

At the highest level of generality, a general CBR cycle may be described by the following four processes [16]:

1. RETRIEVE the most similar case or cases.

The goal of this step is to retrieve old cases stored in the case library [17]. Retrieving is the selection of the most similar cases (source cases) to the new problem specifications (target case) from the library of cases.

2. REUSE the information and knowledge in that case to solve the problem.

3. REVISE the proposed solution.

Because a new case may not exactly match the old one, the old knowledge may often need to be fixed to fit the new one [17].

4. RETAIN the parts of this experience likely to be useful for future problem solving.

In "Fig. 1", this cycle is illustrated [16].

4. The Framework of the Algorithm

Our ant clustering algorithm uses the same probability and density function as the one proposed by Lumer *et al.* [13], but instead of consuming a long time for finding an item to pick, we have kept the location of all available items on the grid in an array. After an ant drops its load, it immediately picks up one item from the list of available items on the grid [18] and the ant jumps to the location of the picked up item.

We have considered a grid with s*s cells. The number of cells must be greater than the number of all objects. All ants move with different speeds on the grid. We have considered a maximum speed for all ants which is set as max_speed=s/3. The step of movement of each ant is generated randomly between 1 and max_speed before each movement. We decrease the max_speed by one every 200 iterations. The details of our proposed method are discussed in the following subsections.

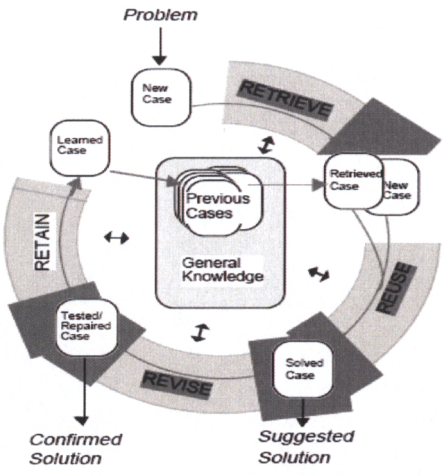


Figure 1. The CBR cycle [16].

4.1 Expert Ants

In the beginning of clustering, all ants are identical, but only with different loads. The memory of all is empty and no one has any kind of information about the search space and the location of items in the grid. We call these ants as normal ants. They move randomly in grid and try to pick up loads using equation (3) and drop their load according to equation (4). An ant becomes an expert by memorizing the location of at least one dropped data. The method of reasoning of these expert ants is based on case based reasoning. Each normal laden ant when for the first time wants to drop an item, needs to use the probability of dropping. It obtains a new experience with each dropping and it keeps the value and location of that data in a repository called case base. After dropping at least one data, it is called an expert ant. When an expert ant is laden, it first refers to its case base, looking for a similar case. If it found such a case, it decides to put its load somewhere next to the location of most similar data it has dropped before. In this way there is no need to compute the probability of dropping and it causes the gathering of similar data in a single area.

4.2 Case and Case Base of Expert Ants

Each ant has a private case base for keeping the different cases it has seen. A case consists of value of attributes and the place they have been dropped. The place of dropping, is a two dimensional matrix (because the search space is a grid). The value of attributes of a data is a multidimensional matrix, in which the number of dimensions depends on the dimensions of data. There is also a variable for each ant which specifies the number of the existing cases in the case base.

4.3 Retrieve and Revise

Whenever an expert ant wants to put its load, it searches for a case in its case base, and compares the value of all cases with its current load to find the most similar case. For this, each ant first collects all the cases having a distance less than a threshold from its load in an array, then it sorts the array ascendingly, and finally selects the first element of the array which has the lowest distance value. But, before retrieving this case, it first checks whether the case is a valid one, or not. Because ants picks up and drop down the data iteratively, it is so probable that the value of case bases become invalid during clustering process. A case is valid for an ant, only if its current location on the grid is the same as the one recorded in the case base of that ant. If the first element of the array was invalid, the second element is checked. This process repeats until a case is retrieved or no other case remains in case base. If a case was retrieved it can be used. It is possible that whenever an ant is looking for a case, it faces a case with identical attribute values with its current load. Under this condition, the case must be revised and then updated. For this purpose, the dropping probability must be computed. When a suitable location for dropping that item is found, the old stored location in case base is replaced with a new one.

4.4 Cooperation between Ants

According to the number of successful droppings, each ant gains some kind of expertise so that the expertise of each ant is different from that of others. In this way we have a multi expert- multi agent system. Whenever a laden ant fails in dropping its load, and an

unladen ant fails for picking up an item, they can sense their neighborhood for getting help from other ants. So, a laden ant can put its load on an unladen ant, and an unladen ant can get load from a laden ant. If both ants were laden, they only exchange their loads. Nothing will happen if both were unladen. If there were more than one neighbor ant, one is chosen randomly. All the process of getting load from other ants and exchanging of loads happen with probability of 0.5. This kind of cooperation can prevent ants from moving aimlessly and it can also increase the speed of convergence.

4.5 The algorithm of clustering by expert ants

The details of our algorithm are as follows:

//Initialization part

1. Spread the data and ants randomly on the grid and give all ants a random load.

//main part

1. **While** iter < max_iter
2. **For** i=1 to #ants
3. a = Choose a random ant without replacement
4. step = Generate a random speed from interval (1,max_speed)
5. new_location=Move (a, step)
6. **if** ant a is unladen and there is an object o in location l then pick_done=try_pick(o)
7. **if** pick_done=0 then try to find a laden expert ant in the neighborhood of ant a and take the load of it with probability of .5
8. **end if**
9. **end if**
10. **if** ant a is laden then search its case base for a case which has the similarity of more than .9 with the current load,
11. if such a case was found then
12. // retrieve
    Drop the load in a neighborhood of 2 from the place of that case
13. **end if**
14. **if** a case was found with the same value with the current load then
15. // revise
    Find a place for dropping it and change the place part of the case with the new found location
16. **end if**
17. **if** no similar case was found and the current location of it is empty then
18. drop_done=try_drop(l)
19. **end if**
20. **if** drop_done=1
21. // retain
    Make a new case and save it in case base
22. **end if**
23. **if** drop_done=0
24. find an unladen expert ant in the neighborhood of 2 and give the load of ant a to it with probability of .5
25. if no unladen expert ant was found, find a laden expert ant in the neighborhood of 2 and exchange the load of ant a with the load of the neighboring ant
26. **end if**
27. **End For**

28. **End while**

try_pick(o)
1. r= a random number from interval (0,1)
2.p=compute probability function (1)
3.**if** p>r then
4.pick up the object o
5.pick_done=1
6.**else**
7.pick_done=0
8.**end if**

try_drop(l)
1.r= a random number from interval (0,1)
2.d=compute probability function (3)
3.**if** d>r then
4.drop down load l
5.drop_done=1
6.**else**
7.drop_done=0
8. **end if**

5. Experimental Results and Comparison

The algorithm is applied to six datasets. The datasets are summarized in Table 1 and all except for Iris and Wine datasets are shown visually in "Fig. 2". Iris and Wine datasets are not illustrated because their dimensionality is more than 3.

We have compared our algorithm with k-means algorithm, which works as follows: Starting from a partitioning of the data into k randomly generated clusters, the k-means algorithm repeatedly (i) computes the current cluster centers (i.e., the average vector of each cluster in data-space) and (ii) reassigns each data item to the cluster whose center is closest to it. It terminates when no more reassignments take place. As k-means can sometimes generate empty clusters, these are identified in each iteration and are randomly

reinitialized. This enforcement of the correct number of clusters can prevent convergence, and we therefore set the maximum number of iterations to 100. To reduce suboptimal solutions k-means is run repeatedly (20 times) using random initialization, and only the best result in terms of intra-cluster variance is returned. [7].

Table 1. Summary of the used data sets. k gives the number of clusters, size shows the number of members of a cluster, and dim is the dimensionality,.

Name-k-size	dim	Source
Square6- 4*50	2	N([0,0],[2,2]),([0,6],[2,2]),([6,0],[2,2]),([6,6],[2,2])
Square5- 4*50	2	N([0,0],[2,2]),([0,5],[2,2]),([5,0],[2,2]),([5,5],[2,2])
Iris- 3*50	4	UCI
Wine-3-(59,71, 48)	13	UCI
D2C2-2*50	2	Artificial data
D2C4-4-(35,57,84,52)	2	Artificial data

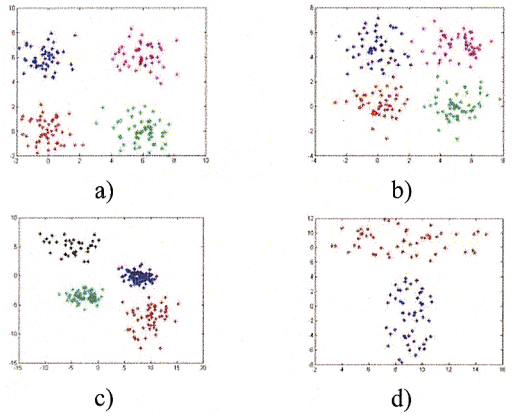
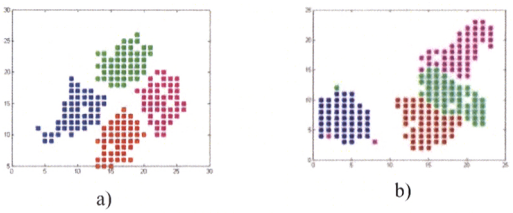


Figure 2. Benchmark data. a) Square6, b) Square5, c)D2C4, d)D2C2

The results of applying k-means algorithm and ant clustering by normal ants are compared to the results of our algorithm in Table 2 and the best result out of a few runs is selected and is illustrated in "Fig. 3".



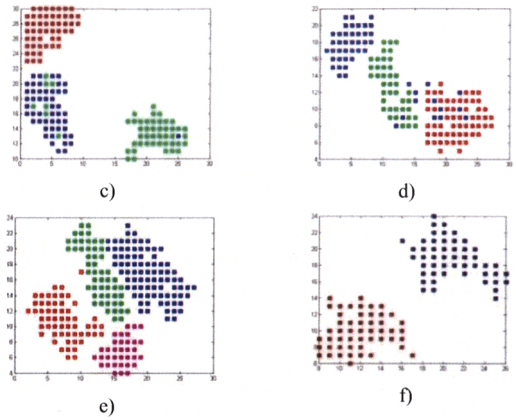


Figure 3. Best visual results of benchmark data. a) Square6, b) Square5, c) Iris, d)Wine, e)D2C4, f) D2C2

A crucial matter in ant based clustering is the requirement for spatial separation between clusters. Fig.3 shows that there is a good boundary between clusters that are generated by our algorithm, which makes the cluster retrieval process to be done with less ambiguity. The results of Table 2 shows the number of clusters that our algorithm can find is nearer to the true number of clusters than normal ants, the average error of clustering which is computed based on the number of objects that were assigned to incorrect clusters is decreased in our approach and the number of iteration needed for our proposed method is much less than that of normal ants.

Table 2. Test results on benchmark dara for K-means, normal ants and expert ants.

Square6	**K-means**	**Normal Ants**	**Expert Ants**
Min Error	0	0	0
Max Error	10	7	6
Average Error	2.9	2.8	2.77
Average Number of Detected Clusters	4	4.4	4.05
Maximum Iterations	100	50000	10000
Square5	**K-means**	**Normal Ants**	**Expert Ants**
Min Error	0	2	2
Max Error	12	15	16
Average Error	3.9	6.7	5.3
Average Number of Detected Clusters	4	4.37	4.31
Maximum Iterations	100	50000	10000
Iris	**K-means**	**Normal Ants**	**Expert Ants**
Min Error	4	8	8
Max Error	17	16	15

Average Error	8.8	9.85	8.72
Average Number of Detected Clusters	3	3.12	2.91
Maximum Iterations	100	500000	100000
Wine	**K-means**	**Normal Ants**	**Expert Ants**
Min Error	4	6	4
Max Error	8	10	9
Average Error	8.13	8.21	7.75
Average Number of Detected Clusters	3	3.28	3.12
Maximum Iterations	100	500000	100000
D2C2	**K-means**	**Normal Ants**	**Expert Ants**
Min Error	0	0	0
Max Error	9	4	3
Average Error	2.7	1.02	0.8
Average Number of Detected Clusters	2	2.1	2.21
Maximum Iterations	100	50000	10000
D2C4	**K-means**	**Normal Ants**	**Expert Ants**
Min Error	0	0	0
Max Error	9	8	7
Average Error	6.8	2.7	2.5
Average Number of Detected Clusters	4	5.85	4.17
Maximum Iterations	100	50000	10000

6. Conclusion

In this article we presented a novel method for ant clustering which uses case based reasoning. We have introduced "expert ants" who are able to decide about the best place for dropping their load using their case bases. The case base of each expert ant contains different information about promising places for dropping loads. We have added a mechanism for taking advantage of the knowledge of other ants. Those ants that were unsuccessful in dropping their loads can give their loads to other unladen expert ants in their neighborhood, and those that were unsuccessful in picking up an object can take the loads of other laden ants in their neighborhood. If two neighboring ants both were laden they can exchange their loads. Our proposed method has shown better results in terms of speed, correctness and compactness of clusters.

7. References

[1] G. Fung, "A Comprehensive Overview of Basic Clustering Algorithms", June 22, 2001

[2] A. Abraham,S. Das,S. Roy, "Swarm Intelligence Algorithms for Data Clustering", In Soft Computing for Knowledge Discovery and Data Mining. Springer Science, 2007.

[3] Q. Li, Z. Shi, J. Shi, and Z. Shi, "Swarm Intelligence Clustering Algorithm Based on Attractor", Springer Berlin / Heidelberg, 2005, pp.496-504

[4] W. bin, S. Zhongzhi, "A clustering algorithm based on swarm intelligence", Proc. of the Int. Conf. on Info-tech and Info-net, Beijing, China, 2001, pp. 58-66

[5] E. Bonabeau, M. Dorigo, and G. Theraulaz. "Swarm Intelligence – From Natural to Artificial Systems", Oxford University Press, New York, NY, 1999.

[6] J. Handl, B. Meyer, "Improved ant-based clustering and sorting in a document retrieval interface", In J. J. Merelo, P. Adamidis, & H.-G. Beyer (Eds.), Lecture notes in computer science: Vol. 2439, 2002.

[7] J. Handl, J. Knowles, M.Dorigo, "Ant-based clustering and topographic mapping", Artificial Life, 12(1), 35–61, 2006.

[8] A. L. Vizine, L. N. de Castro, R. R. Gudwin, "Text document classification using swarm intelligence", In International conference on the integration of knowledge intensive multi-agent systems", Piscataway: IEEE Press. problem solving from nature—PPSN VII, 2005, pp. 134–139.

[9] A. L.Vizine, L. N. de Castro, E. R. Hruschka, , R. R. Gudwin, "Towards improving clustering ants: an adaptive ant clustering algorithm" Informatica, 29, 2005,pp. 143–154.

[10] V. Ramos, J. Merelo, "Self-organized stigmergic document maps: environments as a mechanism for context learning", In Proc. of the first Spanish conf. on evolutionary and bio-inspired algorithms, 2002, pp. 284–293.

[11] S. Schockaert, , M. D. Cock, C. Cornelis, E. E. Kerre, "Efficient clustering with fuzzy ants", In D. Ruan, P. D'hondt, M. D. Cock, M. Nachtegael, & E. E. Kerre (Eds.), Applied computational intelligence, proceedings of the 6th international FLINS conference, 2004, pp. 195–200.

[12] J.-L. Deneubourg, S.Goss, N.Franks, C.Detrain, and L.Chretien "The Dynamics of Collective Sorting: Robot-Like Ant and Ant-Like Robot", In J.-A. Meyer & S. Wilson (Eds.), from animals to animats: proc. of the first int. conf. on simulation of adaptive behavior, Cambridge: MIT Press, 1991, pp. 356-365.

[13] E. Lumer, B. Faieta, "Diversity and Adaptation in Populations of Clustering Ants", Cambridge: MIT Press, In D. Cliff, P. Husbands,J.-A. Meyer, & S. W. Wilson (Eds.), from animals to animats: proceedings of the third international conference on simulation of adaptive behavior, 1994, pp. 501-508.

[14] E. Magnus, H. Pedersen, "Ant Colony Clustering & Sorting", 2003.

[15] J. Dahlgren, A. Khattak, P. McDonough, I. Banerjee, Ph. Orrick, A. Sharafsaleh," Developing Case-Based Reasoning and Expert Systems and Incorporating New Material", 2004.

[16] A. Aamodt, E. Plaza, "Case-Based Reasoning: Foundational Issues", Methodological Variations Methodological Variations and System Approaches. AI Communications, Vol. 17(1), 1994.

[17] Mu-Jung Huang, Mu-Yen Chen, Show-Chin Lee."Integrating data mining with case-based reasoning for chronic diseases prognosis and diagnosis", 2007.

[18] J. Handl, , B. Meyer, "Improved Ant-Based Clustering and Sorting in a Document Retrieval Interface", In J.J. Merelo, J.L.F. Villacañas, H.G.Beyer, P. Adamis Eds. Proceedings of the PPSN VII – 7th Int. Conf. on Parallel Problem Solving from Nature, Granada, Spain, Lecture Notes in Computer Science 2439, Springer-Verlag, Berlin, 2002, pp. 913-923.