A Modified Clustering Method with Fuzzy Ants

Jianbin Chen, Deying Fang and Yun Xue

Business College, Beijing Union University，Beijing, 100025

Abstract: Ant-based clustering due to its flexibility, stigmergic and self-organization has been applied in variety areas from problems arising in commerce, to circuit design, and to text-mining, etc. A modified clustering method with fuzzy ants has been presented in this paper. Firstly, fuzzy ants and its behavior are defined; secondly, the new clustering algorithm has been constructed based on fuzzy ants. In this algorithm, we consider multiple ants based on Schockaert's algorithm. This algorithm can be accelerated by the use of parallel ants, global memory banks and density-based 'look ahead' method. Experimental results show that this algorithm is more efficient to other ant clustering methods.

Keywords: Data mining, Fuzzy ants, clustering, algorithm

1.Introduction

Clustering has been widely studied since the early 60's. Some classic approaches include hierarchical algorithms, partitioning method such as K-means, Fuzzy C-means, graph theoretic clustering, neural networks clustering, and statistical mechanics based techniques. Recently, several papers have highlighted the efficiency of stochastic approaches based on ant colonies for data clustering [1,2,3,4].

While the behavior of individual ants is very primitive, the resulting behavior on the colony-level can be quite complex. A particularly interesting example is the clustering of dead nestmates, as observed with several ant species under laboratory conditions. By exhibiting only simple basic actions and without negotiating about where to gather the corpses, ants manage to cluster all corpses into 1or 2 piles. The conceptual simplicity of this phenomenon, together with the lack of centralized control and a priori information, are the main motivations for designing a clustering algorithm inspired by this behaviorLorem; sed ipsum?

2.Related Work

Ant-based clustering and sorting was originally introduced for tasks in robotics by Deneubourg [3]. This paper has proposed a basic model that explains the spatial structure of cemetery forms as a result of simple, local interactions without any centralized control or global representation of the environment. Holland et al. applied related model to robotics to accomplish complex tasks by several simple robots [5]. Lumer and Faieta modified the algorithm to extend to numerical data analysis by introducing a measure of dissimilarity between data objects [3].Kuntz et al. applied it to graph-partitioning[6], text-mining[7] and VLSI circuit design[8].Wu and shi applied Deneubourg's model in clustering to derive (Clustering based on Swarm Intelligence) CSI model and some important concepts, such as swarm similarity, swarm similarity coefficient and probability conversion function.

Monmarché[1] proposed an algorithm in which several items are allowed to be on the same cell. Each cell with a nonzero number of items corresponds to a cluster. Each ant *a* is endowed with a certain capacity *c(a)* . Instead of carrying one item at a time, an ant *a* can carry a heap of *c(a)* items. Probabilities for picking up, at most *c(a)* items from a heap and for dropping the load on a heap are based on characteristics of the heap, such as the average dissimilarity between items of the heap. When an ant decides go pick up items, the *c(a)* items whose dissimilarity to the centre of the heap under consideration is highest, are chosen. Two particularly interesting values for the capacity of an ant *a* are $c(a)=1$ and $c(a)=\infty$. Monmarché proposes to apply this algorithm twice. The first time, the capacity of all ants is 1, which results in a high number of tight clusters. Subsequently the algorithm is repeated with the clusters of the first pass as atomic objects and ants with infinite capacity. After each pass k-means clustering is applied for handling small classification errors. Inspired by Monmarché's paper, Schockaert *et al.* proposed a clustering method with only one fuzzy ant since the use of multiple ants on a non-parallel implementation has no advantage[9].

3.Fuzzy Ants

In papers[9,10,11,12,13], fuzzy ants have been discussed more than ones. Our algorithm is based on Schockaert's method. Because of the limited space we do not go into detail about the algorithm based on fuzzy ants. We only give same basic concept and focus on the optimization.

First we introduce some notations, and then give some definition.

Let E be a fuzzy relation in X, *i.e.* a fuzzy set in X^2, which is reflexive and T_W-transitive(*i.e.* $T_W(E(x,y),E(y,z)\leq E(x,z))$, for all x,y and z in X) where X in the set of items to be clustered and T_W the Lukasiewicz triangular norm defined by

$T_W(x,y)=max(0,x+y-1)$, for all *x* and *y* in [0,1]. For *x* and *y* in *X*, *E(x,y)* denotes the degree of similarity between the items *x* and *y*. For a heap $H \not\subset X$ with centre *c* in *X*, we define $avg(H)=\frac{1}{|H|}\sum_{h\in H} E(h,c)$ and $min(H)=min_{h\in H}E(h,c)$. Let $E^*(H_1,H_2)$be the similarity between the centres of the heap H_1 and the heap H_2. Because of the limited space we do not go into the detail about how to define and /or compute the centre of a heap, as this can be dependent on the kind of the data that needs to be clustered.

Definition1 A heap is defined as a collection of 2 or more data items. A heap is spatially located in a single cell and has unique ID in this algorithm.

Definition2 The probability that an ant starts performing a task with stimulus *s* and response threshold value θ is given by

$$T_n(s;\theta)=\frac{s^n}{s^n+\theta^n} \tag{1}$$

Where *n* is a positive integer.

Definition 3 The probability of dropping the load is given by

$$P_{drop}=T_{n_i}(s_{drop};\theta_{drop}) \tag{2}$$

Where $i \in \{1,2\}$ and n_1,n_2 positive integers.

Dfinition4 The probabilities for picking up one item and picking up all the items are given by

$$P_{pickup\_one}=\frac{s_{one}}{s_{one}+s_{all}}\cdot T_{m1}(s_{one};\theta_{one}) \tag{3}$$

$$P_{pickup\_all}=\frac{s_{all}}{s_{one}+s_{all}}\cdot T_{m2}(s_{all};\theta_{all}) \tag{4}$$

Where m_1 and m_2 are positive integers, s_{one} and s_{all} are the respective stimuli , θ_{one} andθ_{all} the response threshold values.

The values of the stimuli are calculated by evaluating fuzzy if-then rules as explained in paper[9]. Compared with those algorithms such as FCM, this algorithm has several advantages. Firstly, it is the first time for the fuzzy rules to be used in clustering algorithm. Secondly, it is not sensitive with the initial value of cluster center. Thirdly, it is robust and efficiently. But it is not so perfect and can be optimized on several points.

4.ALGORITHM OPTIMIZATION

Contrasting with those traditional clustering methods, ant-clustering boasts a number of advantages due to the use of mobile agents, which are autonomous entities, both proactive and reactive, and have the capability to adapt, cooperate and move intelligently from one location to the other in the bi-dimensional grid space. Generally said that an ant-based algorithm should be autonomy, flexibility and parallelism. But with the fuzzy ants clustering algorithm discussed in paper[9], there are several drawbacks. Firstly, used only one ant, this method has loosen the parallel characteristic of ants colony. Second, it is too complex to calculate the similarity based on rough set theory. Thirdly, there may be some data objects which have never been assigned to an ant when the algorithm is terminate. Fuzzy ants clustering method needs to be improved in these aspects.

4.1parallelize algorithm

Ant based clustering have some advantages, such as

Autonomy: Not any prior knowledge (like initial partition or number of classes) about the future classification of the data set is required. Clusters are formed naturally through ant's collective actions.

Flexibility: Rather than deterministic search, a stochastic one is used to avoid locally optimal.

Parallelism: Agent operations are inherently parallel.

But if we have only one ants in algorithm, it is not a real ant colony method. So, we preserve multi-ants to perform parallel clustering. There are *n/3* ants, where *n* is the number of data items.

4.2memory bank

The clustering process on the grid can be accelerated significantly by the use of a device of memory bank for the fuzzy ant, a modified version of the 'short-term memory' introduced by Lumer and Faieta[3].

No like Lumer and Faieta's approach that there are ant agents and data items respectively in the system, we have only ant agents. To bias the direction of the agent's random walk, we keep a global memory to store the ever-moved cells for each ant. In the algorithm iteration, the loaded ants can referentially search their memory banks for a best direction to move. The best cell *i*, defined by a pair of coordinate *(x,y)*, is a cell with heap *H* that the load *L* was most similar to.

The decision whether drop its load at cell *i* still be taken probabilistically with the threshold IF-THEN rules defined in paper[9]. Memory bank brings forth heuristic knowledge guiding ants' moving in the bi-dimensional grid space. So the randomness of ants' motion decreases, meanwhile the algorithm's convergence speeds up.

4.3density-based 'look ahead' method

Depending on the definition of items and heap/clusters similarity, we can decide whether a item belongs to a cluster or not. According the idea in DBSCAN that a cluster can be looked as a neighborhood of a given radius which contains at least a minimum number of points, i.e. the density in the neighborhood has to exceed a threshold [14]. So after a few iterations of clustering, the data objects in a high density cell can be approximately classified into the same cluster. Thus these itmes have no need to be visit frequently. The main job is to survey those items in less density cells and determine where to move. Then those items which belong to a cluster already will be resting for a long time, and has lower probability to be moved again. This method will not lead to high misclassification error rate.

Definition: A cell is '***dense-cell***' if the data item number, *Pts*, exceed a certain threshold, *MinPts*. Else if *Pts* less than *MinPts* but bigger than zero, we call this cell as '*sparse-cell*'. Else *Pts=0*, i.e. there is no any items in this cell, we call it "*empty-cell*".

MinPts can be defined as follows:

$$MinPts = k \times \frac{N_{item}}{N_{cell}} \qquad (5)$$

Where *k* is an adjusting coefficient, *k>1*.

We propose here a '**look ahead**' strategy: Before an ant moved to cell, it firstly estimates the cell's density into three types. If the cell is a *dense-cell*, it will go to next position. If the cell is a *sparse-cell*, it will drop a item based on IF-THRN rules. If the cell is *empty-cell*, i.e. there is no data item at all, then the ant agent will moving directly.

5.ALGORITHM IMPLEMENTATION

In our algorithm, there are n data items, and n/3 ants. Initially the items are scattered randomly on a discrete 2D board, the board can be considered a matrix of m×m cells, and m2=4n. We maintained a cell list, each cell has five parameters.

$U = \{X_1 ,X_2,......, X_{m\times m}\}$, and

$$X_i = \{s,t,O,C,Pts\}, i = 1,2,...,m\times m$$

Where *s* and *t* are the position parameter, *O* is a binary value to identify if an ant is visiting this cell, *C* is the center of heap in this cell, and *Pts* is number items in it.

There are also a cell list for each ant to save its visit history, which is a memory bank discussed above.

$MB_i = \{X_j \big| \}$, where *i=1,..., n/3*, and j=1,……, *m*×*m*.

The algorithm is given as follows.

```
Initialize {Cell list, data items, ants, NumberOfIteration=0}
Parallel for each ant ( )
While NumberOfIteration<$NumberOfIteration_{max}$
Get a random heap from cell list and check the Pts
Let $O_i$=1
If the current ant load a heap
    Check whether to unload the loaded item
    Unload and form a new heap
    Calculate the center C and Pts for this heap
    Update the memory bank of this ant
Let $O_i$=0
  Return
End if
Else if  the current ant is unloaded
    Check whether to load the whole heap or the dissimilar item
   Load the whole heap or the dissimilar item
    If the dissimilar item is taken away
     Calculate the center C and Pts for this heap
Check the memory bank to decide next position
Return
    End if
End Else if
NumberOfIteration++
End While
Wait for ending of each task and merge heaps
Display the clustering result
```

Figure1:Tthe modified algorithm

6.EXPERIMENT RESULTS AND ANALYSIS

We have applied our new algorithm to several numerical databases including synthetic ones and real databases from the Machine Learning repository（Machine Learning Repository, http://www.ics.uci.edu/~mlearn/MLRepository .html）.

We have used 4 evaluation measures to evaluate the resulting partition obtained by the three clustering algorithms. They are the number of identified clusters(#Clusters)、Inner Cluster Variance(Variance)， Classification Error Rate

(Cl.Err) [2] and the overall running time of the algorithm(Runtime). Table 1 gives the parts of experiment results.

Table 1: Results for k-means, CSI and AMC on three synthetic data sets: ant1～ant3, and two real data sets: Iris, Soybean.

ANT1	k-means	CSI	Ours
#Clusters	4	4	4
Variance	0.408532	0.331668	0.332412
Cl.Err	2.15%	3.02%	1.26%
Runtime	6.0	10.2	6.3

IRIS	k-means	CSI	Ours
#Clusters	3	3	3
Variance	0.531222	0.411138	0.412018
Cl.Err	5.28%	5.66%	3.37%
Runtime	10.4	13.4	9.8

The results demonstrate that, if clear cluster structures exist within the data, the ant clustering algorithm including: CSI and Ours, is quite reliable at identifying the correct number of clusters. In contrast with the *k*-means, Our algorithm shows its strength in its ability to automatically determine the number of clusters within the data.

Compare the runtimes of the three algorithms, we can see our algorithm is the fastest algorithms and its time consumer changes little with the scale of data set. So it is a fast clustering algorithm with prefect scalability.

7.CONCLUSION

In this paper, we have proposed a modified clustering algorithm with fuzzy ants. With the use of IF-THEN rules, it can be simplify the calculating in clustering. Based on the single fuzzy ant algorithm, we extended it to parallel ants, added a memory bank for each ant, and proposed a density-based method permits each ant to "look ahead", which reduces the times of cell-inquiry. Consequently the clustering time gets saved. We made some experiments on real data sets and synthetic data sets. Compared with other classical clustering algorithm, our algorithm is a viable and effective clustering algorithm.

REFERENCE

[1] Nicolas Monmarché, Mohamed Slimane, Gilles Venturini. AntClass: discovery of clusters in numeric data by an hybridization of an ant colony with the Kmeans algorithm，Internal Report No 213,E3i,January 1999

[2] Deneubourg J L , Goss S , Frank N , Sendova-hanks A ,Detrain C ,Chrerien L. 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] E. Lumer and B. Faieta. Diversity and adaption in populations of clustering ants. In Proceedings of the Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 3, pages . 501–508. MIT Press, Cambridge, MA, 1994.

[4]B.wu,Y.zheng,S.liu and Z.shi, SIM:A Document Clustering Algorithm Based on Swarm Intelligence. IEEE World Congress on Computational Intelligence,Hawaiian,PP.477-482.2002

[5] O.E.Holland and C.Melhuish. Stigmergy, self-organization, and sorting in collective robotics, Artificial Life,5,1999,pp.173-202

[6] P. Kuntz, D. Snyers, and P. Layzell. A stochastic heuristic for visualizing graph clusters in a bi-dimensional space prior to partitioning. Journal of Heuristics, 5(3):327–351, 1998.

[7] K. Hoe, W. Lai, and T. Tai. Homogenous ants for web document similarity modeling and categorization. In Proceedings of the Third International Workshop on Ant Algorithms (ANTS 2002), volume 2463 of LNCS, pages 256–261. Springer-Verlag, Berlin, Germany, 2002.

[8] P.Kuntz,P.Layzell,D.Snyers. A colony of ant-like agents for partitioning in VLSI technology, in: P.Husbans,I.Harvey(Eds.), Proceeding of the Fourth European Conference on Artificial Life, MIT Press, Cambridege,MA,1997,pp.417-424

[9] S. Schockaert, M. De Cock, C. Cornelis, E. E. Kerre. Efficient Clustering with Fuzzy Ants [A]. Applied Computational Intelligence[C] (D. Ruan, P. D'hondt, M. De Cock, M. Nachtegael, E. E. Kerre, eds.), World Scientific,2004. P. 195-200.

[10] P. Kanade. Fuzzy ants as a clustering concept[D]. M.S dissertation. University of South Florida, Tampa, FL.2004.

[11] P. M. Kanade and L. O. hall. Fuzzy ants clustering with centroids[A]. FUZZ-IEEE’04[C], 2004.

[12] S. Schockaert, M. De Cock, C. Cornelis, E. E. Kerre. Fuzzy Ant Based Clustering[A]. Ant Colony Optimization and Swarm Intelligence, 4th International Workshop (ANTS 2004)[C], LNCS 3172. P. 342-349.

[13] Valeri Rozin, Michael Margaliot .The Fuzzy Ant[A]. IEEE International Conference on Fuzzy Systems Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada July 16-21[C], 2006. P.1679-1686

[14] M.Ester, H.-P.Kriegel, J.Sander and X.Xu. A density-based algorithm for discovering clusters in large spatial databases. In Proc.1996 Int.Conf. Knowledge Discovery and Data Mining (KDD’96), page 226-231, Portland, OR, Aug.1996