Proceedings of the 5th WSEAS Int. Conf. on Artificial Intelligence, Knowledge Engineering and Data Bases, Madrid, Spain, February 15-17, 2006 (pp423-428)

Frequent Nodesets by Swarm

KYAW MAY OO
University of Computer Studies, Yangon
MYANMAR

Abstract: - We present a framework of Swarm Technique application algorithm for frequent nodesets. This framework is based on recent research on the behaviour of real ant colonies, a field of Swarm Intelligent. The aim of proposed framework algorithm is to extract frequent nodesets for association rules discovery from data. The proposed algorithm is discussed from the frequent itemsets finding point of view in association rule discovery. A procedure for generating strong association rule from frequent itemsets is our ongoing research and will be presented in our near future paper. At present, a fundamental part of our ongoing research is presented in this paper. We compare the performance of proposed algorithm with the performance of the well-known original Apriori algorithm. The accuracy of proposed algorithm is competitive with that of Apriori. Moreover, our system has found considerably simpler nodesets.

Key-Words: - Ant Colony Optimization, Association Rule, Frequent Itemset, Data Mining, Swarm Intelligent

1 Introduction

Swarm intelligence is a field which studies "the emergent collective intelligence of groups of simple agents" [1]. In groups of insects, which live in colonies, such as ants and bees, an individual can only do simple tasks on its own, while the colony's cooperative work is the main reason determining the intelligent behavior it shows. Ant Colony Optimization (ACO) [4] is a branch of a newly developed form of artificial intelligence called swarm intelligence.

Knowledge discovery in databases (KDD) is the process of extracting models and patterns from large databases. The term data mining (DM) is often used as a synonym for the KDD process, although strictly speaking it is just a step within KDD. DM refers to the process of applying the discovery algorithm to the data. In [8], KDD is defined as "… the process of model abstraction from large databases and searching for valid, novel, and nontrivial patterns and symptoms within the abstracted model".

There has been a great interest in the area of data mining, in which the general goal is to discover knowledge that is not only correct, but also comprehensible and interesting for the user [5]. In DM, discovered knowledge is often represented in the form of $X \Rightarrow Y$ association rules. The rule contains a logical combination of predictor attributes, in the form: term1 AND term2 AND ... . Each term is a triple <attribute, operator, value>, such as <Gender = female>.

Rule Discovery is an important DM task since it generates a set of symbolic rules that describe in a natural way. The human mind is able to understand rules better than any other data mining model. Hence, the user can understand the results produced by the system and combine them with their own knowledge to make a well-informed decision, rather than blindly trusting on a system producing incomprehensible results. However, these rules need to be simple and comprehensive.

Association rule mining is a two-step process: 1) Find all frequent itemsets, and 2) Generate strong association rules from the frequent itemsets.

To the best of our knowledge the use of Ant Colony Optimization [3] as a method for finding frequent itemsets in association rules discovery, in the context of data mining, is a research area still unexplored by other researchers.

Actually, the Ant Colony algorithm developed for data mining that we are aware of is an algorithm for clustering [7], which is, of course, a data mining task very different from the task addressed in this paper. Also, Cordón et al. [2] have proposed another kind of Ant Colony Optimization application that learns fuzzy control rules, but it is outside the scope of data mining. Next, Ant-Miner is Ant Colony algorithm for classification rule discovery [6] which is also a data mining task very different from the association rule discovery task.

We believe the development of Ant Colony algorithms for data mining is a promising research

area, due to the following rationale. An Ant Colony system involves simple agents (ants) that cooperate with one another to achieve an emergent, unified behavior for the system as a whole, producing a robust system capable of finding high-quality solutions for problems with a large search space. In the context of rule discovery, an Ant Colony system has the ability to perform a flexible, robust search for a good combination of logical conditions.

This paper is organized as follows. The second section introduces the framework of Ant Colony system for discovering frequent nodesets (itemsets) proposed in this work. The third section describes the experimental results evaluating the performance of the proposed system and the generating of nodesets. Finally, the fourth section concludes on this work and discusses further directions for future research.

2 Ant Colony System for Discovery of NodeSet

In this section we discuss in detail our proposed framework system for discovery of nodeset. This section is divided into 5 parts, namely: an overview of our proposed system framework, nodeset construction, case pruning, pheromone updating, and system parameters.

2.1 An Overview of Proposed System Framework

Recall that each ant can be regarded as an agent that incrementally constructs/ modifies a solution for the target problem. In our case the target problem is the discovery of frequent nodeset. The proposed system is discussed from the frequent itemsets finding point of view in association rule discovery. Thus, in the rest of the paper, terms in the association rule and (frequent) itemset are referred to nodes and (frequent) nodeset, respectively.

We assume that all the transaction of dataset D be the existing path (i.e. already constructed nodesets) of traveling spaces S traveled by ants. So to determine the deposited amount of pheromone, let a colony of ants be traveled over S. After that pheromone deposited amount on each node of each trial is updated and iteration is started again. Thus after the ant traveled the whole paths in S, we can start to find the desired frequent nodesets by using the clue of the previous ants.

Each ant starts with an empty nodeset and adds one node (one term) at a time to its current partial nodeset. The current partial nodeset constructed by an ant corresponds to the current partial path followed by the ant. Similarly, the choice of a node to be added to the current partial nodeset corresponds to the choice of the direction to which the current path will be extended, among all the possible directions (all nodes that could be added to the current partial nodeset). The choice of the node to be added to the current partial nodeset depends on the amount of pheromone associated with each node, as will be discussed in detail in the later subsections. An ant keeps adding nodes one-at-a-time to its current partial noteset until the ant is unable to continue. This situation can arise in two cases, which are discussed in next section. When one of these two stopping criteria is satisfied, the ant has built a frequent nodeset (i.e. it has completed its path), and in principle we could use the discovered nodeset for generating the association rule.

This process is repeated for at most a predefined number of ants, as a parameter of the system, called No_of_ants. However, this iterative process can be interrupted earlier, when the constructed nodeset is equal to one of the discovered nodesets.

From above nodesets generated by ants, we have to determine that these generated nodesets are frequent or not. To do this, we use the original dataset again. After that the frequent nodesets can be defined.

Then all cases correctly covered by the just discovered nodeset are removed from the dataset. Hence, the proposed system is called again to find additional possible nodesets, if necessary, in the reduced dataset. So another additional ants start to construct other possible nodesets, using the new reduced dataset.

Note that the non-existing nodes in the dataset after pruning (removing) some cases should be deleted as it is not necessary to travel to non-existing nodes. By doing so, ants have to choose only existing nodes in the dataset and it make the system more efficient.

This process is repeated for as much iteration as necessary to find nodesets covering almost all cases of the dataset. More precisely, the above process is repeated until the constructed nodeset is equal to one of the discovered nodesets. A summarized description of the above-discussed iterative process is shown in the proposed system framework of

Figure 1.

When constructed nodeset equals to one of the discovered nodesets, the searching process stops. At this point the system has discovered several nodesets. The discovered nodesets are stored in an ordered rule list (in order of discovery).

Begin
dataset ← all data;
ε ← min-pheromone;
θ ← #ants;
Do
k=0;
Repeat
k = k+1;
Ant-k travels existing predefined nodesets;
Update pheromone amount of trial followed by Ant-k;
Until((end of existing predefined paths)or(k > θ)**)**
While *(! end of existing predefined paths)*
Define one-step-neighbours for each node;
Do
k=0;cont=T;
Repeat
k = k+1;
Ant-k constructs a nodeset of its travel;
If *(not possible to generate) cont=F;*
{ nodeset}←constructed nodeset;
Until (*(!cont) or (k > θ))*
Define frequent nodesets;
If *(cont) Prune the covered cases from dataset;*
While *(cont)*
End

Fig.1 Overview of proposed system framework

2.2 Nodeset Construction

Let $node_i$ be the current node in current partial nodeset traveled by ant. The probability that $node_j$ is chosen by ant-k to be added to the current partial nodeset is given by equation (1).

$$P_{ij}^{k}(t) = \begin{cases} \tau_{ij}(t), & \text{if } node_j \in N_i; \\ 0, & \text{otherwise.} \end{cases} \qquad (1)$$

where $\tau_{ij}(t)$ is the amount of pheromone currently available at time t in the transition i, j of the trail being followed by the ant, and N_i is the set of one-step neighbors of $node_i$.

The amount of pheromone $\tau_{ij}(t)$ is independent of the nodes which already occur in the current partial nodeset, but is entirely dependent on the paths followed by previous ants. Hence, $\tau_{ij}(t)$ incorporates an indirect form of communication between ants, where successful ants leave a "clue" (pheromone) suggesting the best path to be followed by other ants. When the first ant starts to build its nodeset, all trail transitions i, j have the same amount of pheromone. However, as soon as an ant finishes its path the amounts of pheromone in each position i,j visited by the ant is updated, as will be explained in detail in a separate subsection later. Here we just mention the basic idea: the better the quality of the nodeset constructed by the ant, the higher the amount of pheromone added to the trail positions visited by the ant. Hence, with time the "best" trail positions to be followed – i.e. the best nodes (attribute-value pairs) to be added to a nodeset – will have greater and greater amounts of pheromone, increasing their probability of being chosen.

The $node_i$ chosen to be added to the current partial nodeset is the node with the highest value of equation (1) subject to two restrictions. The first restriction is that the $node_i$ cannot occur yet in the current partial nodeset. Note that to satisfy this restriction the ants must "remember" which nodes (attribute-value pairs) are contained in the current partial nodeset. The second restriction is that a $node_{ij}$ cannot be added to the current partial nodeset if this makes the extended partial nodeset cover less than a predefined minimum support, called the Min_sup threshold.

2.3 Case Pruning

Pruning is a commonplace technique. The main goal of case pruning is to remove (reduce) the cases covered by just constructed nodesets that might have been unduly included in the dataset.

Case pruning potentially increases the predictive power of the node, helping to avoid the scanning the whole dataset. Another motivation for case pruning is that it improves the cost of the system, since a reduce dataset is in general more easily assessable by the ant than the whole dataset.

The case pruning procedure is performed for each ant as soon as the ant completes the construction of its noteset. The basic idea is to remove the cases

covered by the just constructed nodeset. That is all the cases in the dataset need to be pruned when all nodes in those each case is subset or identically equal to the node in the union of the discovered nodesets.

2.4 Pheromone Updating

Recall that each $node_{ij}$ corresponds to a position in some path that can be followed by an ant. At each node, local information (i.e. pheromone amount) maintains on the node itself and/or its outgoing transition. This local information is initialized with the same amount of pheromone. In other words, when the system is initialized and the first ant starts its travel all paths have the same amount of pheromone. The initial amount of pheromone deposited at each path position is inversely proportional to the number of nodes (i.e. the number of values of all attributes), as given by equation (2).

$$\tau_{ij}(t=0) = \frac{1}{\sum_{k=1}^{a} b_k} \qquad (2)$$

where $\tau_{ij}(t)$ = pheromone deposited on the transition of node i to j at time t;
a = the total number of attributes; and
b_k = the number of values in the domain of attribute k.

The value returned by this equation is already normalized, which facilitates its use in a single equation.

Each time an ant completes the construction of a nodeset (i.e. an ant completes its path) the amount of pheromone in all positions of all paths must be updated. This pheromone updating has two basic ideas, as followed.

The amount of pheromone associated with each $transition_{ij}$ occurring in the constructed nodeset is increased by a constant amount $\Delta\tau$ of pheromone. This pheromone updating equation (3) is

$$\tau_{ij}(t) = \tau_{ij}(t) + \Delta\tau \qquad (3)$$

The amount of pheromone associated with each $transition_{ij}$ that does not occur in the constructed nodeset is decreased, corresponding to the phenomenon of pheromone evaporation in real Ant Colony Systems. The decrease pheromone amount of an unused node is considered by the comparison of pheromone amount of used and unused nodes, in this paper.

2.5 System Parameters

Our System has the following two user-defined parameters:

- Number of Ants (No_of_ants): . This is the maximum number of complete nodesets constructed during a single iteration of the system, since each ant constructs a single nodeset (see Figure 1). Note that the larger the No_of_ants, the more nodesets are evaluated per iteration, but the slower the system is;
- Minimum support percentage (Min_sup): . Each note must cover at least Min_sup, to enforce at least a certain degree of generality in the discovered nodesets. Min-sup-count can be calculated from this parameter Min-sup and the original dataset.

The behavior of individual ants to produce a desired response in the colony behavior is done with the use of above system parameters. By optimizing these parameters, the optimal solution can be reached. In this paper, we have made no serious attempt to optimize these parameter values. Such an optimization will be tried in future research. From our experimental result, it is interesting to notice that even the above non-optimized parameters' setting has produced quite good results.

There is one caveat in the interpretation of the value of No_of_ants which defines the maximum number of ants per iteration of the system. The reason why in practice much fewer ants are necessary to complete an iteration of the system is that iteration is considered to be finished when all possible nodesets are equal to one of the discovered nodesets.

3 Experiments

Using the well-known T40I10D100K and T10I4D100K datasets and the mushroom dataset (570kB, with 8124 transactions), we try to test the ability of our proposed system. Here we present the experimental results of proposed algorithm and Apriori. Tests were run on a PC with 594 MHz Pentium processor and 448MB RAM. The operating system was WindowsXP. The following 2 figures present the test results of the proposed algorithm and Apriori on the 3 databases. Each test was carried out 3 times; the figures shows the averages of the results.

In figure 2, we compare the execution time of

both algorithms over various supports for three different databases. This figure indicates that the proposed system constantly performs well as Apriori for various supports. Figure 3 shows the total number of the frequent nodesets found for three databases with various support values. Depending on the support value used, the total number of the frequent nodesets found in proposed algorithm is smaller than that of Apriori.

In all the experiments reported in this paper, the parameter No_of_ants were set as 10000 for all databases. But the actual number of ants per iteration was less than this and thus all the experiments were required one iteration only. This means that all the results in this paper were totally required two scans of databases: one for pheromone deposition and other for production of frequent nodesets. We have made no serious attempt to optimize this parameter values. Although, this non-optimized parameter setting has produced well results. Such an optimization will be tried in future research.

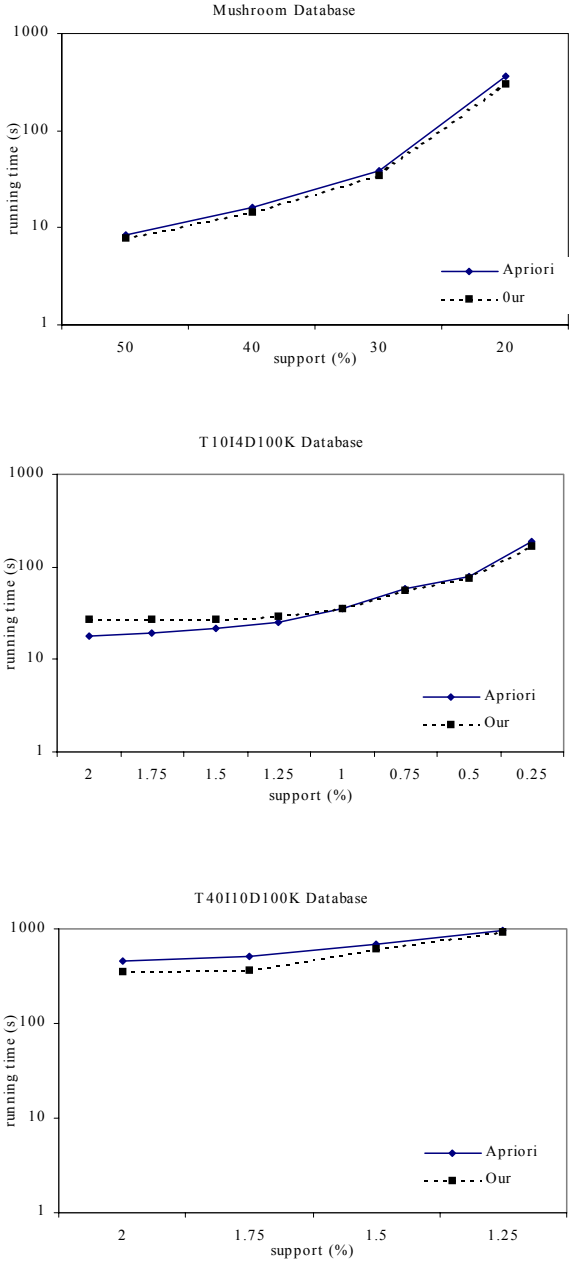

Fig.2 Proposed System versus Apriori

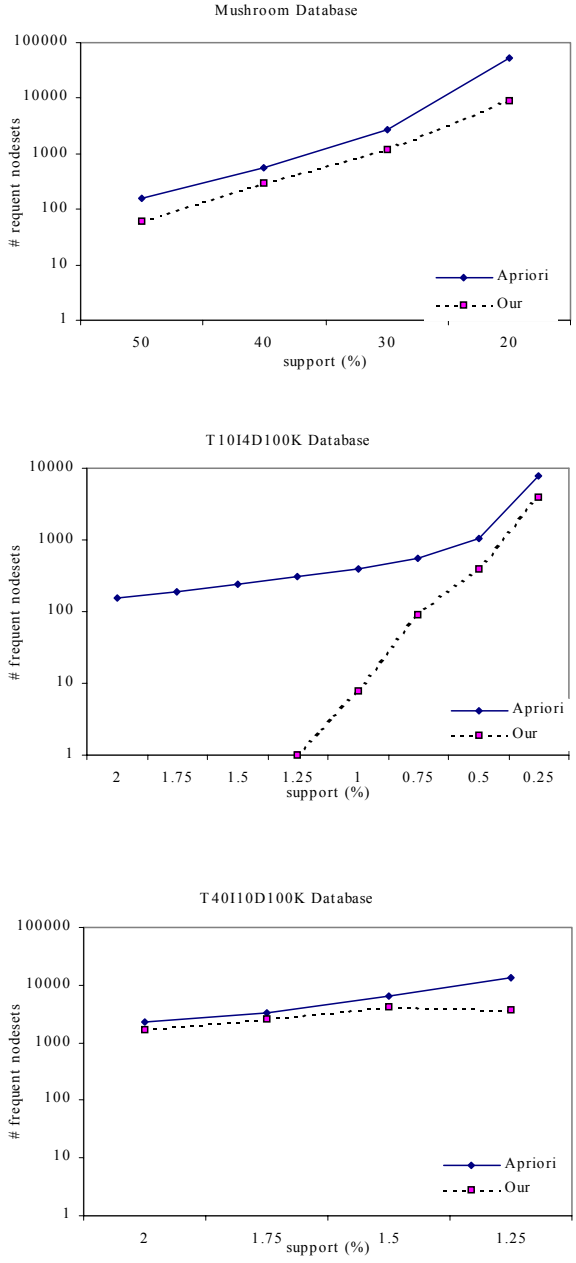

Fig.3 Set Cardinality

From our results, the proposed algorithm intuitively produces simpler set of nodeset, which is covered the nodesets generated by Apriori. Moreover, it only need to be scan the reduced size of dataset, i.e. may be more efficient to find frequent nodesets, but more detail analysis will be needed.

4 Conclusion and Future Work

This work has proposed a framework of an algorithm for nodeset discovery. The goal is to extract frequent nodeset from data. The algorithm framework is based on recent research on the behavior of real ant colonies as well as in some data mining concepts.

We have to compare the performance of our algorithm with the performance of the well-known original Apriori algorithm.

One research direction consists of generating association from the result of proposed system; this is our ongoing primary research.

Other research direction consists of performing several experiments to investigate the sensitivity of our algorithm to its user-defined parameters. In addition, it would be interesting to investigate the performance of other kinds of pheromone updating strategy.

References:

[1] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. New York: Oxford University Press, 1999.

[2] O. Cordón, J. Casillas, and F. Herrera, "Learning Fuzzy Rules Using Ant Colony Optimization," in Proc. ANTS'2000 – From Ant Colonies to Artificial Ants: Second International Workshop on Ant Algorithms, 2000, pp. 13-21.

[3] M. Dorigo, A. Colorni, and V. Maniezzo, "*The Ant System: Optimization by a colony of cooperating agents*," IEEE Trans. Systems, Man, and Cybernetics-Part B, vol. 26, no. 1, pp. 1-13, 1996.

[4] M. Dorigo, and G. D. Caro, "*Ant Algorithms for Discrete Optimization*," Artificial Life, vol. 5, no. 3, pp. 137-172, 1999.

[5] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, From data mining to knowledge discovery: an overview. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P. & Uthurusamy, R. (Eds.) *Advances in Knowledge Discovery & Data Mining*, 1-34. Cambridge: AAAI/ MIT, 1996.

[6] B. Liu, H. A. Abbass, and B. Mckay, "Classification rule Discovery with Ant Colony Optimization," Prec IEEE/WIC Int Conf on Intelligent Agent Technology, IAT-2003, vol. 18, pp. 83-88, Oct 2003.

[7] N. Monmarche, "On data clustering with artificial ants," In: A. A. Freitas, Ed., AAAI-99 & GECCO-99 Workshop on Data Mining with Evolutionary Algorithms: Research Directions, Florida,1999, pp. 23-26, 1999.

[8] R. Sarker, H. Abbass, and C. Newton, "Introducing data mining and knowledge discovery," In R. sarker & H. Abbass & C. Newton (Eds.), *Heuristics and Optimisation for Knowledge Discovery*, pp. 1-23: Idea Group Publishing, 2002.