IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
Vol. 07, Issue 07 (July. 2017), ||V1|| PP 18-23

An Improved Association Rule Mining Algorithm Based on Apriori and Ant Colony approaches

*Dr.Hussam M. Al Shorman, Dr.Yosef Hasan Jbara
Al-Balqa Applied University Buraydah Private Colleges
Corresponding Author: Dr.Hussam M. Al Shorman

Abstract: The Knowledge Discovery in Databases (KDD) field of data mining is useful in finding trends, patterns and anomalies in the databases which is helpful to make accurate decisions for the future. Association rule mining is an important topic in data mining field. Association rule mining finds collections of data attributes that are statistically related to the data available. Apriori algorithm generates all significant association rules between items in the database. Besides, ACO algorithms are probabilistic techniques for solving computational problems that are based in finding as good as possible paths through graphs by imitating the ants' search for food. The use of such techniques has been very successful for several problems. The collaborative use of ACO and DM (the use of ACO algorithms for DM tasks) is a very promising direction. In this paper, based on association rule mining and Apriori algorithm, an improved Ant Colony algorithm is proposed to solve the Frequent Pattern Mining problem. Ant colony algorithm is employed as evolutionary algorithm to optimize the obtained set of association rules produced using Apriori algorithm. The results and comparison of the method is shown at the end of the paper.

Date of Submission: 11-07-2017 Date of acceptance: 22-07-2017

I. INTRODUCTION

Data mining is the most instrumental tool in discovering knowledge from large database transactions [1][2][3]. It consists of techniques for discovering previously unknown, valid patterns and relationships in large data sets and has emerged as an important technology with numerous practical applications, due to wide availability of a vast amount of data. Recently, there has been a growing interest in the data mining area, where the objective is the discovery of knowledge that is correct, comprehensible and surprising to the user. Transactional database refers to the collection of transaction records such as medical records, banks records, sales records, and others. Data mining on transactional database focuses on the mining of association rules, finding the correlation between items in transaction records. Association Rule Mining is an important technique in data mining. Association rule is used to define the relationship between various modules. Association tells that how modules or various objects are associated with each other. A major concern today in Association Rule Mining is toimprove the algorithmic performances. Association Rule Mining was used to find association between the data in large dataset. Most of the association rule algorithms are based on methods proposed by Agrawal, Imielinski, and Swami [4] and Agrawal and Srikant [5], which are Apriori [4], SETM[6], AIS [4] and Pincer search [7] etc.

Recently, the swarm intelligence paradigm, where a number of individuals with limited capabilities are able to come to intelligent solutions for complex problems, received widespread attention in research. Ant Colony Optimization (ACO) is a popular swarm intelligence meta-heuristic proposed by Dorigo et al. [8] as a new heuristic to solve combinatorial optimization problems. It is an agent-based system which simulates the natural behavior of ants and develops mechanisms of cooperation and learning. The use of such techniques has been very successful for several problems. It has been shown to be robust and versatile approach for solving a wide range of hard combinatorial optimization problems.

The basic requirement to apply ACO on any problem is the undirected graph with certain parameters on the edge between two nodes. In this paper we, we propose a method using ant colony optimization to solve the Frequent Pattern Mining problem using fixed number of parameters for all edges in the graph. We represent the graph with exact number of parameters for all edges and also ACO is applied to find the solution. We have implemented the new algorithm and evaluate its performance by comparing to other recent algorithms. The rest of this paper is organized as follows. Section 2 presents a brief explanation of Ant Colony optimization algorithm. Section 3 presents association rule mining. In section 4 related works are discussed. The proposed method is presented in section 5.An example to show the proposed approach is given in Section 5.Implementation, experimental setup and results are presented in section 6.Conclusion and Future work are discussed in the Section 7.

II. ANT COLONY OPTIMIZATION

Ant Colony Optimization algorithm [8] was inspired from natural behavior of real ant colonies. This algorithm was first proposed by M. Dorigo, 1992. Ant Colony Algorithm is a multi-agent approach for solving difficult combinatorial optimization problems. ACO is a multi-agent system based on agents which simulate the natural behavior of ants, including mechanisms of cooperation and adaptation. Artificial ants are simple agents implementing constructive heuristics. The basic idea of constructive heuristics is incrementally construct solutions by adding, in each step, a solution component to a partial solution until to a complete solution is formed. ACO has been applied successfully to numerous hard combinatorial optimization problems [9, 10, 11, 12, 13, 14, 15, and 16]. Problems are defined in terms of components and states, which are sequences of components. ACO incrementally generates solutions paths in the space of such components, adding new components to a state.

The ACO system contains two rules [8]:
1. Local pheromone update rule, which applied whilst constructing solutions.
2. Global pheromone updating rule, which applied after all ants construct a solution.
On the other hand, an Ant Colony Optimization algorithm has two more mechanisms:
Trail evaporation: This decreases all trail values over time, in order to avoid unlimited accumulation of trails over some component.
Daemon actions: This used to implement centralized actions which cannot be performed by single ants.

Ant Colony Optimization (ACO) is inspired by shortest path searching behavior of various ant species. Ants go through the food while laying down chemical evidence on the path that it took in order to be followed by other ants, which is called pheromone trails. Shortest path can be found via pheromone trails. As an ant searches and finds food, it evaluates the quality and the concentration of it. While going back to the nest, some of the ants will drop pheromone in amounts proportional to the quality and the concentration of the food. More the pheromone trails better the path. Because ants follow the intense pheromone trails, it will be guided to the founded food.

The ACO meta-heuristic consists of three algorithmic components [8]:
1. Solution construction using pheromone trail: A solution construction starts with an empty partial solution then they add solution components one by one. The choice of a solution component is done at each construction step with a stochastic local decision policy that makes use of pheromone trails and problem specific information using the following probability decision rule:

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in \mathcal{N}_i}[\tau_{il}(t)]^{\alpha}[\eta_{il}]^{\beta}} \quad \forall j \in \mathcal{N}_i^k \qquad (4)$$

Where, N_i^k is the set of all feasible solution components still to be visited. τ_{ij} is the pheromone value associated with solution component cij, and η_{ij} is a weighting function that assigns at each construction step a heuristic value to each feasible solution component. The higher the value of solution component heuristic η, the higher its probability of being chosen. Furthermore, α and β, are positive parameters, whose values determine the relation between pheromone information and heuristic information.
2. Pheromone trail update: this is applied in two phases, the intensification phase, where each ant deposits an amount of pheromone which is proportional to the fitness of its solution, and evaporation phase, where a fraction of the pheromone evaporates in order to avoid too rapid convergence of the algorithm toward local optima.The first increases the pheromone levels by a certain value Δτ, and the second phase decreases all the pheromone values through pheromone evaporation. The pheromone levels are increased using the following function:

$$\tau_{ij}(t) \leftarrow \tau_{ij}(t) + \Delta\tau^{k}(t), \qquad (5)$$

Where, the rate of increment of pheromone, Δτij, is given by:

$$\Delta\tau_{ij} = \sum_{k=1}^{m}\Delta\tau_{ij}^{k} \qquad (6)$$

Where, m is the number of ants, and $\Delta\tau_{ij}$ is a problem specific dependent.
The trail intensity on each edge is updated according to the following function:

$$\tau_{ij}(t+1) = \rho \cdot \tau_{ij}(t) + \Delta\tau_{ij} \qquad (7)$$

Where, ρ (0 ≤ ρ ≤ 1) is a coefficient such that (1 - ρ) represents the amount of pheromone evaporated. t is the beginning of a tour, and t+1is the beginning of the next tour. This process is iterated until the given termination conditions are satisfied.
3. Daemon Action: This is the optional component which is used to implement centralized actions that cannot be performed by single ant. For example, a local search

III. ASSOCIATION RULES MINING

Given a large set of transactions, the problem of mining association rules is to find all interesting association rules that have support and confidence greater than the user-specified minimum support and minimum confidence threshold respectively. Association rules are used to detect the common usage of data items. For example purchasing of one product when another product is purchased represents an association rule. Association rule mining is one of the most important issues in data mining. A typical example of ARM is market basket analysis. In market basket analysis each record contains a list of items purchased by a customer. We are interested to find out the set of items that are frequently purchased together. The sets of items occurring together can be written as association rules. An association rule is written as X =>Y, where X and Y are set of items. A formal model is introduced in [17]. Let I = {i1, i2, …,im} be a set of binary attributes, called items. Let D a set of transactions and each transaction T is a set of items such that $T\subseteq I$. Let X be a set of items. A transaction T is said to contain X if and only if $X\subseteq T$. An association rule is an implication of the form $X\Rightarrow Y$, where $X\subset I$, $Y\subset I$, and $X\cap Y = \emptyset$. The support for an association rule X Y is the percentage of transactions in the database that contain X U Y. The confidence or strength for an association rule XY is the ratio of the number of transactions that contain X U Y to the number of transactions that contain X. The goal is to find all rules whose support and confidence are greater than the minimum support and confidence threshold specified by the user. The definition of a frequent pattern relies on the following considerations [17]. A set of items is referred to as an itemset (pattern). An itemset that contains *k* items is a *k*-itemset. For example the set {*name*, *semester*} is a 2-itemset. There are different real world applications of ARM including market basket analysis, customer segmentation, electronic commerce, medical, web mining, finance, and bio informatics.

It has been shown that the problem of discovering association rules can be reduced to two sub-steps:

1) Find all frequent itemsets in a database for a predetermined minimum support.

2) Generate high confidence rules from each frequent itemset.

The overall performance of mining association rules is determined by the first step.

IV. INTRODUCTION TO APRIORI ALGORITHM

It is a classic algorithm used in data mining for learning association rules. Apriori algorithm [18] is used to efficiently discover large itemsets. Apriori algorithm uses the property that if an itemset is frequent, then all of its subsets must also be frequent. Based on this, the Apriori algorithm generates a set of candidate large itemsets whose lengths are (k+1) from the large k-itemsets (for k≥1) and eliminates those candidates, which contain not large subset. If the candidate item sets satisfies minimum support then it is frequent item sets. Then, for the rest candidates, only those with support over minsup threshold are taken to be large (k+1)-itemsets. The Apriori generate itemsets by using only the large itemsets found in the previous pass, without considering the transactions. The inputs to the Apriori algorithm is the itemset, Database of transactions, support and the output is large itemsets. Apriori uses breadth-first search and a tree structure to count candidate item sets efficiently. A number of researchers applied a modified version of Apriori algorithm to a large scale itemsets in large databases.

V. RELATED WORK

Badri Prasad Patel, Nitesh Gupta, Rajneesh K. Karn and Y.K. Rana [22] have proposed an algorithm based on ACO to optimize the association rule generated by using Apriori algorithm. In their paper they show that Ant Colony Optimization approach can be used to improve the quality of rule sets generated from the Apriori algorithm for association rule mining. Yan-hua WANG and Xia FENG [23] have proposed an algorithm based on the directed network. They have shown by experiments that the proposed algorithm promotes the efficiency of computing. Anshuman Sadh and NitinShukla[24] have proposed a mining based optimization techniques for rule generation based on apriori algorithm and ant colony optimization approach. They applied apriori algorithm to find the positive and negative association rules. Then they used ant colony optimization algorithm to optimize the association rules generated. Al-Dharhani Ghassan; Othman Zulaiha [25] have proposed a graph-based ant colony optimization approach for association rule mining. It consists of two phases. The first phase enhances the normal Apriori algorithm and engages in a data representation scheme. The second phase embellishes the ACO-ARM, which relies on the graph of 2-frequent items to generate the final frequent itemset.

VI. THE PROPOSED APPROACH

Given the original data set O of T transaction, find subset S, which contains p candidate generation (p < T, S < O), such that the generation of association rule is improved. τi the intensity of pheromone trail which returns the previous knowledge about the important of O_i. Where, O is the original dataset O = {o1, o2, o3……..on}. *Sj* = {S1, S2, S3………Sp} is a list that contains the selected candidate key subset and associated with each ant. Pheromone value is updated using the following rule:

Fitness = $(\tau_i *2)^{\alpha + 0.1}$ *conf (k)*l log(support(k)*length(k)+1)* $(Li^{Sj})^{\beta}$ /d

Where, Length (k) is the length of association. The larger the support and confidence, the greater the strength of the association. D is the number of transaction set, Li^{Sj} is the local importance of O_i given the subset S_j , and the parameters a and b control the effect of pheromone. The main steps of the algorithms are as follows:

1. Initialize number of ants a and the heuristic values;
2. Initialize pheromone value of all trails $\tau i = c$ and $\Delta\tau\ i = 0$, (i=1, 2 .....n), where c is a constant and $\Delta\tau_i$ is the amount of change at pheromone trail generating at support and confidence
3. Set maximum number of iteration.
4. Define e where p-e is the number of generated candidate key that each ant will start within the next step.
5. Initially set $m_k = \{S_j$ for all J: $F^{\sigma(\{Sj\})} >N^*$ Min_sup}
6. For $j = 1$ to T_a, let $p = p + 1$
7. Let O = Apriori generated (M_k–1)
8. On iteration completion, update pheromone value.
9. For each transaction T? O do $n\tau_i$ = subset (m_k, τi)
10. $p_k = \{c|c: M_k^{\sigma(c> N^* \text{Min\_sup})}$ until $m_k = \varnothing$
11. Reset = m_k
12. For each m_k do {C = {i|i: F_k}
13. Call generate-rules (F_k, na)
14. Display optimized rules

VII. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The proposed algorithm is implemented using Microsoft Visual Basic for Applications (VBA) and run on a 3.2 GHz Pentium 4 PC with 4 GB of RAM and 250GB Hard Disk running the XP operating system. Three data sets of 12000, and 32000 transaction are randomly sampled from the Food Mart transaction database. The test database contains 32 and 51 items respectively. The longest transaction contains 15 and 23 items respectively and the lowest transaction contains 8 and 14 items respectively. Our proposed approach run with three values of minimum support (Minsup) which are 0.30, 0.45%, 0.55% and three values of minimum confidence (Minconf) which are 0.35%, 0.40%, 0.65%.

We have observed that employing ACO to optimize the obtained set of association rules produced using Apriori algorithm generated less number of association rules, as compared to using Apriori algorithm without employing ACO approach. Table 1 presents our experimental results. It is also notable that this reduction is dependent on the support and confident values chosen. In Figure 1, comparison of Apriori and the proposed ACO-Apriori Based algorithm is shown for the number of rules generated vs. different support values using dataset 1. The next comparison (Figure 2) is made using dataset 2. From both figures, we can note that the smaller the value of minimum support, the more the number of rules generated.

Table 1: Experimental Results

Data set 2	Data set 1	
32000	12000	no. of transactions
51	32	No. of items
23	15	Max items/ transaction
14	8	Min items/ transaction
0.30%, 0.45 %, 0.55%	0.30%, 0.45 %, 0.55%	Support %
0.35%, 0.40 %, 0.65%	0.35%, 0.40 %, 0.65%	Confidence %
714, 656, 579	**422, 398, 366**	**Avg # rules /Apriori**
509, 443, 377	**312, 277, 221**	**Avg # rules / Apriori-ACO**

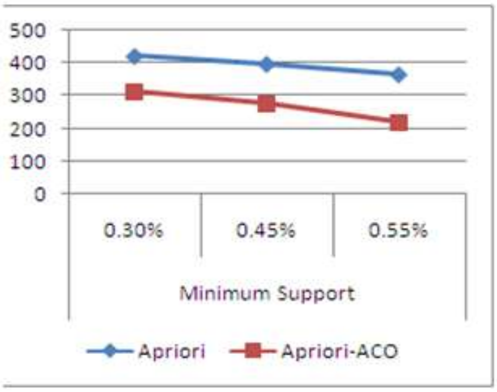

Figure 1: No. of rules generated vs. different support values using dataset 1

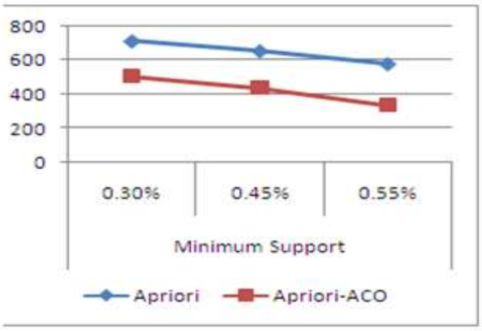

Figure 1: No. of rules generated vs. different support values using dataset 2

VIII. CONCLUSION AND FUTURE WORK

In this paper, based on association rule mining and Apriori algorithm, an improved Ant Colony algorithm is proposed to solve the Frequent Pattern Mining problem. Ant colony algorithm is employed as evolutionary algorithm to optimize the obtained set of association rules produced using Apriori algorithm. We presented experimental results, showing that the proposed algorithm outperform other algorithms found in the literature. An important contribution of our approach is that it drastically reduces the CPU time associated with other algorithms. We demonstrate the effectiveness of our algorithm using sample databases. Future work includes applying these algorithms to real data like retail sales transaction, medical transactions, Molecular Biology, etc. to confirm the experimental results in the real life domain. Proposing an efficient way of tuning the algorithm's parameters which would result in better solutions. Modifying our algorithm to effectively and efficiently applied to large-scale distributed data mining, particularly in the Internet environment.

REFERENCES

[1] Chen C.-H., Hong T.-P., and Tseng V.S., A Cluster-Based Fuzzy-Genetic Mining Approach for Association Rules and Membership Functions, IEEE International Conference on Fuzzy Systems, pp. 1411 - 1416, 2006.

[2] Kayaa M., Alhajj R., Genetic algorithm based framework for mining fuzzy association rules, Fuzzy Sets and Systems , Vol. 152, No. 3, pp. 587-601, 2005.

[3] Tsay Y. J., Chiang J. Y., CBAR: an efficient method for mining association rules, Knowledge] Based Systems,Vol. 18, No. 2-3, pp. 99-105, 2005.

[4] Agrawal R., Imielinski T., and Swami, A., Mining association rules between sets of items in large databases, In proceedings of ACM SIGMOD conference on management of data, pp. 207-206, 1993.

[5] Agrawal R., Sirkant R., Fast algorithms for mining association rules, In proceedings of the 20th international conference on very large databases, Santiago, Chile,1994.

[6] Houtsma A., Swami M., Set-oriented mining of association rules, Research Report, 1993.

[7] Lin D. I., Kedem Z. M., Pincer-search: An efficient algorithm for discovering the maximal frequent set, In Proceedings of sixth European conference on extending database technology, 1998.

[8] Dorigo, M., Maniezzo, V., & Colorni, A., The Ant system: optimization by a colony of cooperating agents, IEEE Trans. Systems, Man, Cybernetic. B, Vol. 26, No. 1, pp. 29-42, 1996.

[9] M. Dorigo and L. M. Gambardella, Ant colony system: A cooperative learning approach to TSP, IEEE Trans. Evol. Comput., vol. 1, 1997, pp. 53–66.

[10] Jbara.Yosef, New ant colony algorithm for continuous function optimization in 2d and 3d search spaces, In Proceeding of the 21st IASTED International Conference 696 (027), 117, January 2010.

[11] M. Dorigo, G. Di Caro, and L. M. Gambardella, “Ant algorithms for discrete optimization,” Artificial Life,1999,5(2), pp. 137-172.

[12] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, Data mining with an ant colony optimization algorithm, IEEE Transactions on Evolutionary Computation, vol. 4, 2002,pp. 321-332.

[13] R. Zitar, Yosef Jbara, "Application of ACO to the Terrain Generation in 3 Dimensional Continuous Search Space", with. The International Conference on Applied Simulation and Modeling (ASM 2008). Corfu, Greece.

[14] K. M. Sim and W. H. Sun, Ant colony optimization for routing and load-balancing: survey and new directions, IEEE Trans. on systems man, and cybernetics -part A: system and humans, vol. 33, 2003, pp. 560-572.

[15] M. Dorigo and L.M. Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation, 1(1):53–66, 1997.

[16] Riyad Al-Shalabi, Jbara. Yosef, “Application of 2D-3D continuous ant colony approach to pipe distribution network design optimization problems”. In Proceedings of the Fifth IASTED International Conference 711, 043-76, 2010.

[17] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," Proc. ACM SIGMOD, May 1993.

[18] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules in Large Databases,” Prof. 20th Int’l Conf. Very Large Data Bases, pp. 478499, 1994.

[19] Farah AL-Zawaidah, Yosef Jbara and Marwan Abu-Zanona, "An Improved Algorithm for Mining Association Rules in Large databases ". In Proceeding of the world of computer science and information technology (WCSIT 2011). ISSN: 2221-0741.

[20] Hussam Al-shorman, Yosef Jbara, "An Efficient Algorithm for Mining Association Rules for Large Itemsets in Large Databases". In Proceedings of the 2009 International Conference on Information & Knowledge Engineering, IKE 2009 (727-730), USA, 2 Volumes, ISBN 1-60132-116-3.

[21] M. Al-zoubi, H. Al-shorman and Yosef Jbara, "An Experimental Comparison of Five Association Rule Algorithms Using Synthetic and Real-World Datasets". In proceedings of the 2008 International Conference on Information & Knowledge Engineering, IKE 2008, 2008 (61-66), USA. ISBN 1-60132-075-2.

[22] Badri Prasad Patel, Nitesh Gupta, Rajneesh K. Karn and Y.K. (2011), Optimization of Association Rules Mining Apriori Algorithm Based on ACO, International Journal on Emerging Technologies 2(1): 87-92, ISSN: 0975-8364.

[23] Yan-hua, Wang, Xia Feng, ’The optimization of Apriori algorithm based on directed network. 3rd international Symposium on intelligent information technology application, 2009.

[24] Anshuman Sadh and Nitin Shukla, "Apriori and Ant Colony Optimization of Association Rules, International Journal of Advanced Computer Research, ISSN-P: 2249-7277, Volume-3 June-2013.

[25] Al-Dharhani Ghassan; Othman Zulaiha; Abu Bakar Azuraliza, "A Graph-Based Ant Colony Optimization for Association Rule Mining', Arabian Journal for Science & Engineering (Springer Science & Business Media B.V.). Jun2014, Vol. 39 Issue 6, p4651-4665. 15p.