Journal of Web Engineering, Vol. 13, No.3&4 (2014) 263-276

AN IMPROVED ANT COLONY ALGORITHM FOR EFFECTIVE MINING OF FREQUENT ITEMS

SURIYA SUNDARAMOORTHY

Department of Computer Science and Engineering,

Velammal College of Engineering and Technology, Madurai, Anna University, Chennai, India

suriyas84@gmail.com, ssu@vcet.ac.in

DR. S.P.SHANTHARAJAH

Department of Master of Computer Applications,

Sona College of Technology, Salem, Anna University, Chennai, India

spshantharaj@gmail.com

Received December 15, 2012

Revised January 11, 2014

Data Mining involves discovery of required potentially qualified content from a heavy collection of heterogeneous data sources. Two decades passed, still it remains the interested area for researchers. It has become a flexible platform for mining engineers to analyse and visualize the hidden relationships among the data sources. Association rules have a strong place in representing those relationships by framing suitable rules. It has two powerful parameters namely support and confidence which helps to carry out framing of such rules. Frequent itemset mining is also termed to be frequent pattern mining. When the combination of items increases rapidly, we term it to be a pattern. The ultimate goal is to design rules over such frequent patterns in an effective manner i.e in terms of time complexity and space complexity. The count of evolutionary algorithms to achieve this goal is increasing day by day. Bio Inspired algorithms holds a strong place in machine learning, mining, evolutionary computing and so on. Ant Colony Algorithm is one such algorithm which is designed based on behaviour of biological inspired ants. This algorithm is adopted for its characteristic of parallel search and dynamic memory allocation. It works comparatively faster than basic Apriori algorithm, AIS, FP Growth algorithm. The two major parameters of this algorithm are pheromone updating rule and transition probability. The basic ant colony algorithm is improved by modifying the pheromone updating rule in such way to reduce multiple scan over data storage and reduced count of candidate sets. The proposed approach was tested using MATLAB along with WEKA toolkit. The experimental results prove that the stigmeric communication of improved ant colony algorithm helps in mining the frequent items faster and effectively than the above stated existing algorithms.

Key words: association rule mining, support, confidence, biologically inspiration, stigmergic communication, pheromone updation, transition probability.

Communicated by: B. White & J. Vanderdonkt

1 Introduction

Data Mining Today size of databases ranges in uncountable manner so that within this mass of data lays hidden data of strategic importance from business point of view. Many organizations worldwide started using data mining techniques to increase their revenues. Data mining adopts a set of data analysis tools to discover patterns and their relationships. Data mining is data analyzing process [1] which ends up with extraction of required data by summarizing its correlation or patterns in different dimensions. The demand for data mining has been increased mainly due to growth of heavy storage of tremendous data. The various concepts and techniques of data mining include mining frequent patterns, associations, correlations, classification and prediction, cluster analysis, and so on. The most critical challenging problems [2] in data mining are development of a unifying theory, scalability in terms of high dimensional data and high speed data streams, sequential data mining, time series data mining, complex knowledge mining, high speed mining in networks, distributed and multi agent data mining, security problems while mining, unbalanced and cost sensitive data mining and so on.

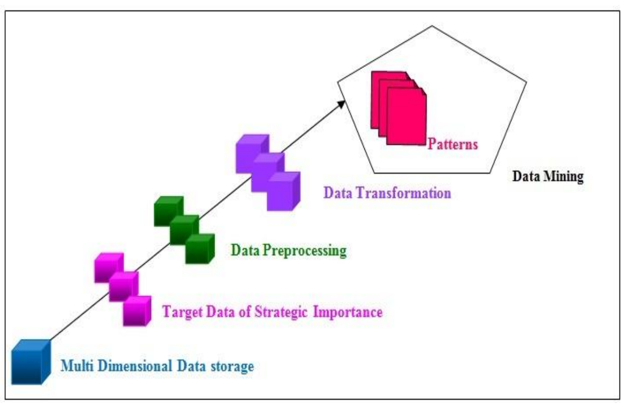


Figure 1: Path from scratch data storage to Data mining

The KDD process (shown in Figure1) involves three phase's namely (i) pre-processing, (ii) data mining process and (iii) post-processing [1]. The pre-processing phase is carried out before applying data mining techniques to sample datasets. It involves cleaning and integrating of databases. As data is brought together from different data sources, it is necessary to remove the noisy data, inconsistent data and duplicate data. Later integration of data from various data sources is done. The data mining process is the second and major phase of KDD [3]. It deals with selection of relevant data from the integrated sources. Later those data are transformed into a convenient format for mining. Various data mining techniques can be carried out over that format of data. Data mining technique makes use of various algorithms to extract data from multidimensional databases. Few examples are Apriori algorithm for association rules, neural network algorithm for prediction, classification and clustering, regression algorithm for classification and prediction. The last phase post-processing deals with evaluation and representation of results obtained from second phase. It completely deals with visualization of results in one of the following forms namely tables, data cubes, charts, decision trees, etc. Data mining strategies were classified into 3 categories namely descriptive modeling, predictive modeling and market basket analysis. Predictive modeling approach makes use of input attributes to determine and predict output attributes. It is further classified into 3 sub categories like classification, estimation and prediction. The predictive technique is mainly known for its strategic importance towards breast cancer survivability and future car market demand [4&5]. Descriptive modeling approach identifies the best set of attributes and evaluates the performance of the model with the help of input attributes. One of the best examples of descriptive data mining is the phrasal extraction in digital documents collection [6]. A new innovative process of mining textual patterns has been very effectively discussed in paper [7] (which includes the process of pattern deploying, and pattern evolving). Market Basket analysis determines the interesting relationship among the stocks that promotes the business. A pattern is said to be a frequent pattern or frequent item set only if has a frequency either greater than or equal to a user specified count. For example, a person who buys biscuits tends to buy cool drinks or fruit juices i.e., {biscuits, cool drinks}, {biscuits, fruits juices}. Similarly a person who buys a pencil tends to buy a sharpener later a rubber then we call it to be a sequential frequent pattern. These frequent patterns contribute towards identifying hidden interesting relationships among the data. Therefore frequent item set mining [8] has occupied a vital place in the research works of data mining.

Lemma: let T be a transaction database and S be a set of items {SI1, SI2, SI3… SIn} with 'n' number of items. The k-item set with frequency § is said to be a frequent pattern only if $\S > \mu|T|$ where $|T|$ is the total number of transactions in T.

In 1993, Agrawal [9] introduced association rule mining as one of most important techniques of data mining for point of scale (POS) systems in supermarkets. The main intention of association rule mining is to extract interesting pattern of data from huge data repositories [10]. A rule is defined as an implication of the form A=>B where $A \cap B \neq \emptyset$. The left-hand side of the rule is called as antecedent. The right-hand side of the rule is called as consequent. For example [11,12] the rule { Onions, Potatoes}=>{beef} found in the sales data of a supermarket would indicate that if a customer buys Onions and potatoes together then the customer is likely to buy beef also. Such information is useful to make decisions about marketing activities. Association rules are also used in many applications including Web usage Mining, Intrusion Detection and Bio-informatics. There are two phases for association rule mining namely (i) First Phase – all frequent item sets that satisfy minimum support are found (ii) Second Phase – those frequent item sets are used for generating association rules. I = {i1, i2, i3, …, im} is a collection of items. T is a collection of transactions associated with the items. Every transaction has an identifier TID [5]. Association rule A=>B is such that A∈I, B∈I. A is called as Premise and B is called as Conclusion. The support, S, is defined as the proportion of transactions in the data set which contains the item set. The confidence is defined as a conditional probability

Support(X=>Y) = Support (XUY) = P (XUY) and Confidence(X=>Y) = Support (XUY) / Support (X) = P(Y/X)

Lemma: The occurrence of a pattern1 { SI1,SI2, SI3,…, SIj } assures the occurrence of another pattern2 { SIj+1,…, SIk } i.e., P({ SI1,SI2, SI3,…, SIj } | { SIj+1,…, SIk }) can be determined as follows based on the scenario: the knowledge about { SIj+1,…, SIk } may make the occurrence of {SI1, SI2, SI3… SIj}

(i)more likely (i.e.,P({ SI1,SI2, SI3,…, SIj } | { SIj+1,…, SIk })>P({ SI1,SI2, SI3,…, SIj })),
(ii)less likely (i.e., P({ SI1,SI2, SI3,…, SIj } | { SIj+1,…, SIk })<P({ SI1,SI2, SI3,…, SIj })), or
(iii)have no effect (i.e.,P({ SI1,SI2, SI3,…, SIj } | { SIj+1,…, SIk }) =P({ SI1,SI2, SI3,…, SIj })).

Data Representation: Association Rule Mining makes use of one of the below two database layouts namely horizontal database layout or vertical database layout. The former layout is traditional layout which consists of set of transactions and each transaction in turn consists of a set of items. In case of latter layout, each item consists of a set of transaction IDs', popularly known as tidsets. This layout also adopts bit vector representation. The latter outperforms the former layout. Computing field follows two ways of solving a given problem i.e either by following exact method or heuristic approach. The former technique deals with logical or mathematical manner of solving problems latter deals with complex optimization problems which cannot be solved by traditional methods with acceptable time and space complexity. Heuristic approaches are integrated with biologically inspired concepts. Biologically inspired algorithms are designed based on the natural behavior of organisms. Bio-inspired computing has emerged as a new era for of providing computational solutions to complex problems in data mining [13]. These heuristic approaches play a vital role in various applications of web mining [14]. Categories of such bio inspired computing (inspired by nature) are (i) Swarm Intelligence (ii) Artificial Immune Systems (iii) Evolutionary Computation and (iv) Neural networks. The Swarm Intelligence includes (1) Ant Colony (2) Particle Swarm and (3) Bee algorithm.

DATA REPRESENTATION

Horizontal Database Layout

TID	ITEMS
T1	I_2, I_3, I_4
T2	I_1, I_2, I_5
T3	I_1, I_3, I_4, I_5
T4	I_4, I_5
T5	I_2, I_3, I_5
T6	I_1, I_2, I_4
T7	I_3, I_4, I_5

Vertical Database Layout

ITEMS	TIDSETS
I_1	T2, T3, T6
I_2	T1, T2, T5, T6
I_3	T1, T3, T5, T7
I_4	T1, T3, T4, T6, T7
I_5	T2, T3, T4, T5, T7

Figure 2: Data Representation

The proposed approach is based on Ant Colony algorithm. The source of inspiration of Ant colony algorithm is foraging behavior of natural ant colonies. It was introduced in 1990 as an optimization technique which is popularly termed as Ant Colony Optimization (ACO) [15]. The attractive feature of ACO is its parallel search over the sample data with the help of dynamic memory structure which stores the previous obtained good results. Ant-based models [16] are applied in accordance to data mining context to preform clustering, sorting, topographic mapping, etc. This algorithm is capable of adapting (at run-time) to the dynamic underlying environment [17]. The modified pheromone updating rule of ant colony algorithm proposed in my previous research paper [18] drives faster than the basic ant colony algorithm. The proposed approach is designed in such a way with an improved pheromone updating rule that it works faster than that proposed in my earlier research work [19]. It helps to generate new association rules for the dynamic changing database which consists of set of transactions along with different collection of items.

2 Association Rule Mining

Association Rule Mining [11, 20] involves mining over target data based on the predefined parameter minimum support and confidence calculation. It involves two phases namely identification of frequent item sets over the given data storage and generation of association rules [8, 9, 12, 21]. The former phase involves two stages namely candidate generation and generation of frequent item sets. The latter phase is carried out with help of confidence parameter to frame association rules. Agrawal et al., in 1993, proved the importance of these two parameters support and confidence in his research [9]. A sequential collection of items is popularly termed as an item set or a pattern. Support parameter helps to identify the percentage either an item or a pattern with respect to the total number of transactions. Confidence is a measure than is framed to strengthen the association rules. Confidence parameter is used to estimate the percentage of count of transactions that contain the required item or pattern with respect to total count of transactions that contain at least one combination of those subsets. Consider the following sample dataset.

TID	Item sets
T1	i1,i2,i5
T2	i2,i4
T3	i2,i3,i5
T4	i1,i2,i3
T5	i1,i3,i4
T6	i2,i3,i4
T7	i1,i3,i5
T8	i1,i2,i3,i5
T9	i1,i2,i5
T10	i1,i2

Table: 1 Sample Dataset

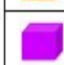
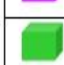
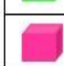
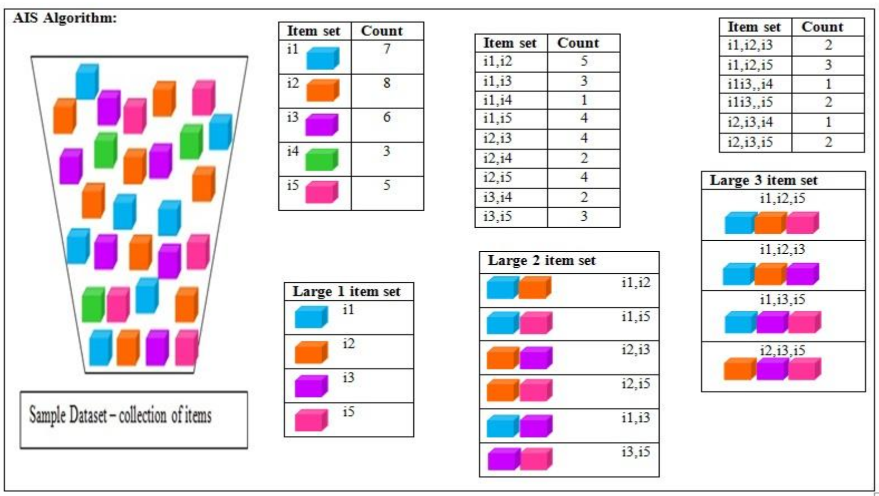
Item Color	Item Name
	i1
	i2
	i3
	i4
	i5

Table: 2 Item Representations

The association rule mining [22] provides two powerful measures for evaluating the association between items which is expressed by a rule. The confidence of a rule measures the degree of the correlation among items, while the support of a rule measures the significance of the correlation among items.

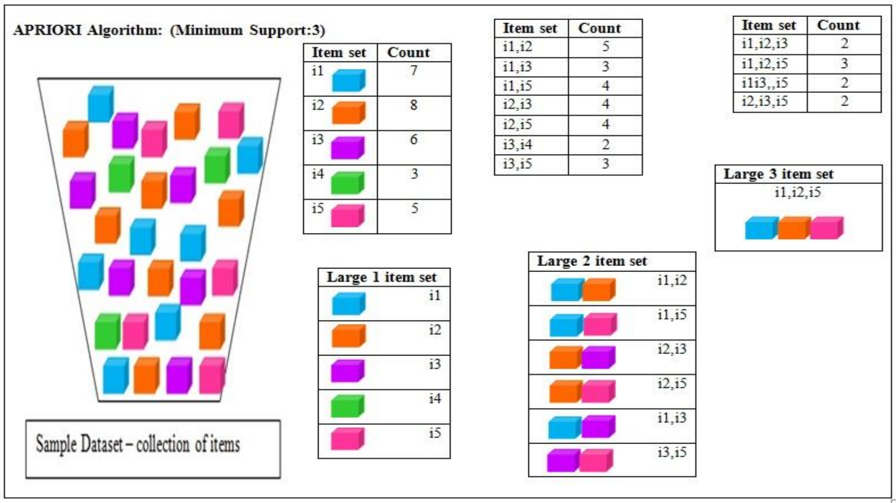
AIS Algorithm

AIS (Agrawal, Imielinski, Swami) algorithm [9] was first proposed algorithm for mining association rules in 1993. They have applied this algorithm over customer transactions in such a way that when it is over the sales data, significant association rules between the data are generated. The strength of this algorithm is proper buffer management and novel pruning and estimation techniques. Large item sets remain in disk and not in memory during every scan over the database. The pruning function is used as an optimization technique to eliminate any particular collection of items whose value is less than that of minimum support value. The key problems of this algorithm are high memory space requirements; increasing number of scan over the sample database, too many candidate item sets that finally turned out to be small was generated. For the above dataset in Table 1, the working of AIS algorithm is traced out in the following figure.



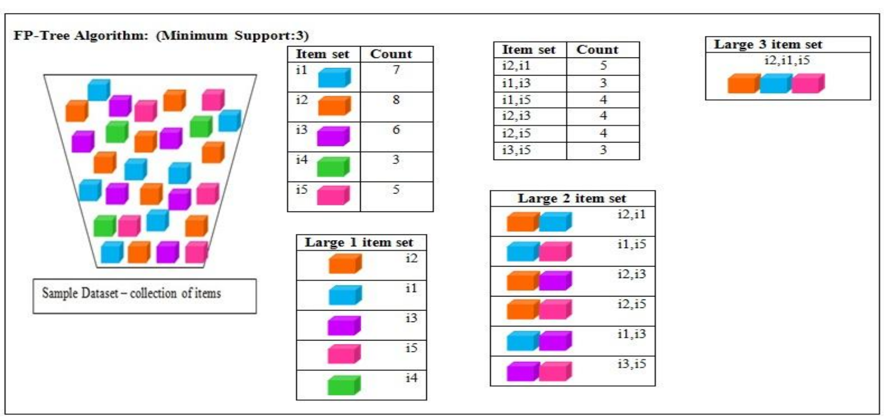
APRIORI Algorithm

Apriori algorithm was most efficient association rule mining algorithm of 1990's. It was proposed by Agrawal in 1994 [1]. It was able to overcome the drawbacks of the previous AIS algorithm. The best features of apriori algorithm are efficient candidate generation process and usage of pruning techniques to avoid measuring non frequent item sets. But still, the two major bottlenecks [21] of this algorithm are usage of most of the time, space and memory for candidate generation process and multiple scans over the sample database. Lei [23] and Sheng Chai [24] has introduced a new improvement over the basic apriori algorithm with the intension to reduce the number of candidate item sets and to avoid multiple scans over database. For the above dataset in Table 1, the working of Apriori algorithm is traced out in the following figure.



FP-Tree Algorithm

FP-Tree (Frequent Pattern) Mining algorithm was introduced by Han in 2000. It works comparatively faster than apriori algorithm. FP-Tree is a prefix tree structure algorithm which stores information related to frequent item sets. It involves two steps namely construction of FP-Tree and generation of frequent patterns from the tree. The major drawbacks of this algorithm are lack of interactive mining system (where users don't have the freedom of changing the support values) and not suitable for incremental mining environment. In 2005, Grahne [25] used this FP Tree algorithm for compressed storage. Though the algorithm consumes much of memory space, it remains suitable for cases when the minimum support value is low for sparse data. In case of dense data, it consumes less memory and remains fastest. In 2009, Shengwei Li [26] introduced the same algorithm for mining closed frequent itemsets, where the algorithm scans the database only once. For the above dataset in Table 2, the working of FP-Tree algorithm is traced out in the following figure.



3 Improved Ant Colony Algorithm

Basic Ant Colony Algorithm: Marco Dorigo was the first person who was inspired by behaviour of natural ants and framed the famous Ant Colony algorithm [27]. From earlier of 1990's, he started his research activities over this algorithm. Initially he found the algorithm to be one of the best optimization algorithms over the computing domain. Though various swarm based algorithms has emerged after it, still it has occupied its space among researchers interest. This algorithm [28] is known for its speed and adaptability to the environment, time and space complexity. This algorithm has its root idea grasped from the behaviour of natural ant colonies during their food search. Many researchers wondered how these ants move one after another without any food prints. Later it was found that these lay a chemical substance popularly known as pheromone. This pheromone is found in the ground surface for a period of time and later gets evaporated. This plays as a source for indirect communication among the fellow ants. This type of communication through the pheromone is termed as stigmeric communication. During food search, initially ants move in all random direction (in various paths) but after some time all ants are found to move in a single line i.e in the shortest path. This is because the shortest path will have the evaporation rate of pheromone to be low and makes it easy for fellow ants to find the path.

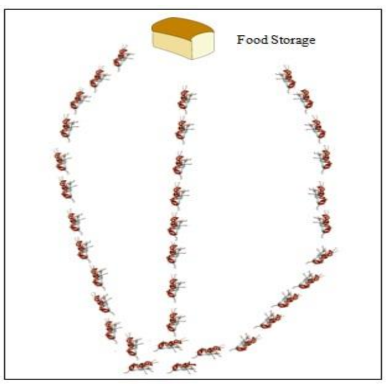


Figure: 3 Initial Stage of search for food source

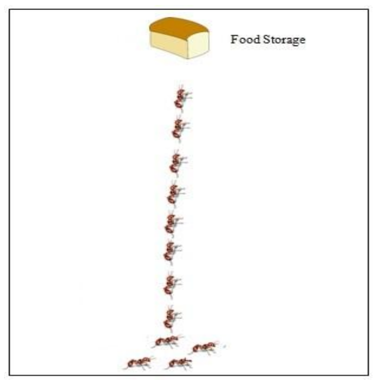


Figure: 4 Final Stage of search for food source

Consider the above example (in figure 3), where a group of ant colonies are observed to move in search for food source. At Initial stage of search, they randomly move in possible directions. As they move, pheromone is laid on the ground on all the paths wherever they travel. After some time, it is observed that the pheromone on the other paths except the shortest path gets evaporated as shown in figure 4. Thus the fellow ants follow the shortest path with the help of pheromone laid on that path. This algorithm has been adopted towards various applications like task scheduling, vehicle routing, grid computing, travelling salesman problem, etc. This paper aims to improve the performance of ant colony algorithm technique for effective mining of frequent itemsets. This proposed approach is initiated with a minimum support value for ant agents to start mining. The existing algorithms as discussed above (AIS, Apriori, FP-Tree) mining the frequent itemsets but their execution never addresses the situations when the transactions are of dynamic nature. The pheromone trail [19] is updated in order to allow ants to share information related to good solutions. The best ant is selected for retaining the pheromone value at the end of every iteration. Pheromone is represented using the parameter ρ (0< ρ<1). When the value of ρ reaches 1 then decrementing is stopped. When the value of ρ reaches 0 then pheromone value is erased. Pheromone updating rule is given by: τij(t)new = [τij(t)old] + [{ρ} Δ τij(t)] ---(1)

Where

τij(t) → Trail intensity of the edge(i,j).

ρ → Evaporation rate.

Δτij(t) → Additional pheromone when ant moves from one transaction to another.

This updating rule was modified in my previous research work[19] to check the performance of this algorithm over frequent pattern mining. Heuristic information and pheromone trail values are used for mining the databases by the ant agents. The ant agents start with the minimum support value and the transaction list as shown in table1. Each transaction T_i is then probabilistically chosen to mine next based on the pheromone value. The probability of selecting an item to be included for frequently occurring item is generated using the equation (2). Equation (2) consists of two parameters namely α and β. The parameter α indicates the relative weightage given to pheromone and the parameter β indicates the relative weightage given to heuristic information. The probability selection is given as $P_{ij}(t)^k = [\tau_{ij}(t)]^{\alpha} * [\eta_{ij}(t)]^{\beta} / \Sigma_{u \in allowed(k)} [\tau_{iu}(t)]^{\alpha} * [\eta_{iu}(t)]^{\beta}$ .... (2)

Where

$P_{ij}(t)$ → Probability to move along the path (i to j).

$\tau_{ij}(t)]$ → Trail intensity of the edge(i,j).

$\eta_{ij}(t)$ → Visibility (1 / $distance_{ij}$).

At the end of each iteration [19], the identified item satisfying the minimum support value is allocated to the best selected ant. This process is repeated until all the transactions in the list are traversed and a complete solution is built. Every ant in the system follows the same manner to build the solution. The pheromone trail is updated after all the ants build a solution. Ants make use of the pheromone value (minimum support value) in identifying the item which is frequently occurring in many transactions. According to the dynamic change in the transactions in the database and also to change in minimum support value, ants generate the frequently occurring items. The pseudo code of basic ant colony algorithm for mining frequent item set is as follows:

```
Algorithm BasicACO_Mining (Nant, Ts, Supportvalue)
//Problem Description: To identify frequent patterns over dynamic storage.
//Inputs: Number of ant agents(Nant), Sample Database (Ts), Minimim Support Value (Supportvalue)
//Output: Frequent item patterns.
Begin
   Initialize the pheromone (Supportvalue)
   While lists in the Ts is not empty do
      For each transaction Ti in the list Ts do
          Start each ant agent to mine for frequent_check
      end for
     Choose next item with the help of state transition rate
        Pij(t)k = [τ ij(t)]α*[ηij(t)]β/Σu€allowed(k) [τ iu(t)]α*[ηiu(t)]β
     until every ant has build a solution
     Update the pheromone (if required)
        τij(t)new = [ τij(t)old ] + [ {ρ} * Δ τij(t) ]
   end while
end
```

Figure: 5 Basic Ant colony algorithm for mining frequent item sets

```
Algorithm ImprovedACO_Mining (Nant, Ts, Supportvalue)
//Problem Description: To identify frequent patterns over dynamic storage.
//Inputs: Number of ant agents(Nant), Sample Database (Ts), Minimim Support Value (Supportvalue)
//Output: Frequent item patterns.
Begin
   Initialize the pheromone (SupportValue)
   While lists in the Ts is not empty do
       For each transaction Ti in the list Ts do
           Start each ant agent to mine for frequent_check
      end for
     Choose next item with the help of state transition rate
         Pij(t)k = [τ ij(t)]α*[ηij(t)]β/Σu€allowed(k) [τ iu(t)]α*[ηiu(t)]β
     until every ant has build a solution
     Update the pheromone (if required)
         τij(t)new=[{(1- ρ)/(1+ρ)}*τij(t)old]+[{ (ρ(1-ρ)) / (1+ρ) }*Δτij(t)]
   end while
end
```

Figure: 6 Improved Ant colony algorithm for mining frequent pattern

The proposed improved ant colony algorithm has a new pheromone updating rule. It is shown in equation (3). It is found that the proposed algorithm works faster than the algorithm that I proposed in my previous research work[19]. This algorithm exhibits a good time space trade off. It is found that this proposed approach is much suitable for databases of dynamic nature. The algorithm of improved ant colony algorithm is shown in Figure 6.

$$\tau_{ij}(t)new=[\{(1-\rho)/(1+\rho)\}*\tau_{ij}(t)old]+[\{(\rho(1-\rho))/(1+\rho)\}*\Delta\tau_{ij}(t)] \quad (3)$$

where

$\tau_{ij}(t)$ → Trail intensity of the edge(i,j).

ρ → Evaporation rate.

$\Delta\tau_{ij}(t)$ → Additional pheromone when next transaction is traversed.

4 Experimental Results

WEKA can be integrated with MATLAB for various applications like selection of genes[29] and mining of frequent patterns [19]. WEKA is one of the famous tools used for Machine learning[30]. The proposed approach is implemented in MATLAB environment along with WEKA tool. WEKA along with MATLAB became a powerful tool for implementing Bio-inspired algorithms[31]. This proposed approach makes use of these above ideas to implement the proposed approach.

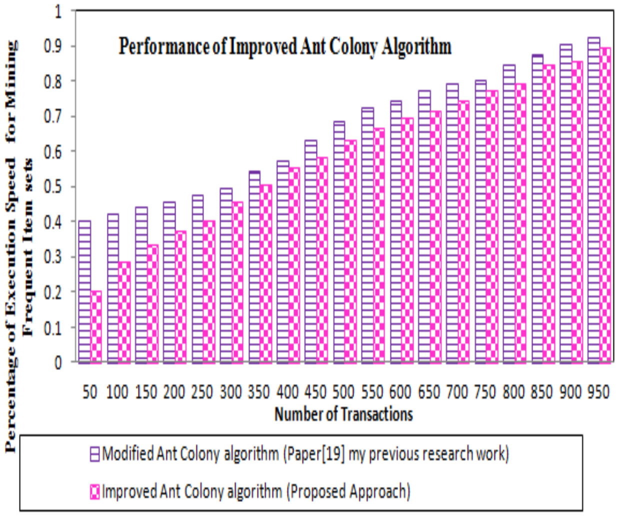
Table 3 shows the experimental results of Modified Ant colony algorithm (Paper [19] my previous research work) and Improved Ant colony algorithm for mining frequent items. As in previous research work, weka data stored in java weka Instance are converted as objects in MATAB. It involves four basic parameters namely (i) WEKA input – input java instances, (ii) transaction list, (iii) Minimum Support value and (iv) Frequent pattern. MATLAB code is used for preprocessing activity. The function called trainWekaAttrSelector is used for item selection. Filter function is used for training, validating and testing activities. This helps in filtering out non frequent items. Table 3 shows the comparison of execution speed of proposed approach over my previous research approach [19].

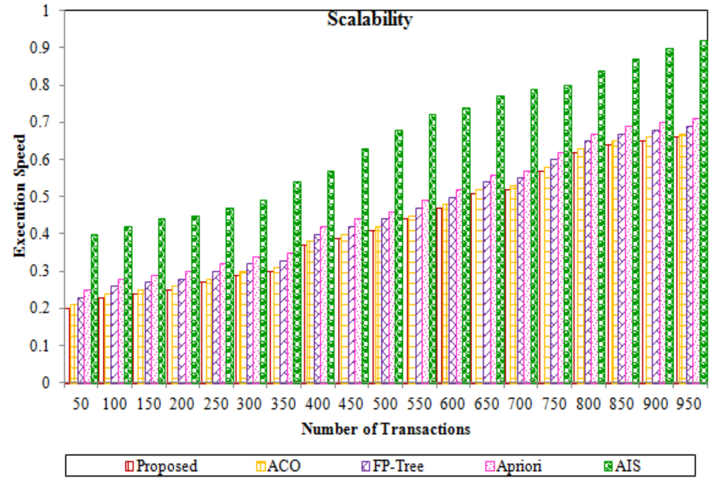
Graph 1 reflects the performance of proposed approach over the modified algorithm in [19]. Dynamically set of transactions are increased in terms of 50's to evaluate their comparative performance. This graph proves the performance improvement using proposed approach.

The table 4 shows the comparative performance of AIS algorithm, Apriori algorithm, FP-Tree algorithm, modified ant colony algorithm[19] and improved ant colony algorithm. The results are tabulated for intervals of every 50 transactions as in case of table 3. Graph 2 make us realize the tremendous performance of proposed algorithm when compared to some existing algorithms for mining frequent item sets.

No: of Transactions	**Percentage of Execution Speed for Mining Frequent Item sets**	
	Modified Ant colony algorithm (Paper [19] my previous research work)	**Improved Ant Colony algorithm (Proposed Approach)**
50	0.4	0.20
100	0.42	0.28
150	0.44	0.33
200	0.45	0.37
250	0.47	0.40
300	0.49	0.45
350	0.54	0.50
400	0.57	0.55
450	0.63	0.58
500	0.68	0.63
550	0.72	0.66
600	0.74	0.69
650	0.77	0.71
700	0.79	0.74
750	0.8	0.77
800	0.84	0.79
850	0.87	0.84
900	0.9	0.85
950	0.92	0.89

Table: 3 Modified Ant colony algorithm VS Improved Ant colony algorithm for mining frequent items





Graph: 1 Performance of Basic ACO VS IACO

Graph 2: Performance Rain of Proposed approach

Table 4:Comparison of AIS, Apriori, FP-Tree, ACO and Proposed approach

No: of Transactions	Percentage of Execution Speed				
	Proposed	Apriori	FP-Tree	ACO	AIS
50	0.2	0.21	0.23	0.25	0.4
100	0.23	0.24	0.26	0.28	0.42
150	0.24	0.25	0.27	0.29	0.44
200	0.25	0.26	0.28	0.3	0.45
250	0.27	0.28	0.3	0.32	0.47
300	0.29	0.3	0.32	0.34	0.49
350	0.3	0.31	0.33	0.35	0.54
400	0.37	0.38	0.4	0.42	0.57
450	0.39	0.4	0.42	0.44	0.63
500	0.41	0.42	0.44	0.46	0.68
550	0.44	0.45	0.47	0.49	0.72
600	0.47	0.48	0.5	0.52	0.74
650	0.51	0.52	0.54	0.56	0.77
700	0.52	0.53	0.55	0.57	0.79
750	0.57	0.58	0.6	0.62	0.8
800	0.62	0.63	0.65	0.67	0.84
850	0.64	0.65	0.67	0.69	0.87
900	0.65	0.66	0.68	0.7	0.9
950	0.66	0.67	0.69	0.71	0.92

5 Conclusion and future work

This proposed approach is found to be the fastest of all the existing algorithms. Experimental results prove the effective frequent pattern mining over dynamic nature of datasets. Future work is planned as a hybrid approach of this proposed algorithm with either transaction mapping algorithm or any other swam intelligence algorithm like bee colony algorithm or particle swarm algorithm.

References

1. Jiawei Han, Micheline Kamber, "Data Mining: Concepts and Techniques", Second Edition, Morgan Kaufmann publishers, 2006.
2. Qiang Yang, Xindong Wu, "10 Challenging Problems In Data Mining Research", International Journal of Information Technology & Decision Making ,Vol. 5, No. 4, page no: 597–604, 2006.
3. Heikki Mannila, Hannu Toivonen, "Levelwise Search and Borders of Theories in knowledge Discovery", International Journal of Data Mining and Knowledge Discovery (Springer), Volume 1, Issue 1, pg: 241 – 258, 1997.
4. Sarvestani.A.S., Safavi.A.A, Parandeh, N.M, Salehi.M, "Predicting breast cancer survivability using data mining techniques", IEEE Second International conference on Software Technology and Engineering, volume 2, Page no:227 -231,3-5 October 2010.
5. Al-Noukari.M, Al-Hussan.W, "Using Data Mining Techniques for Predicting Future Car market Demand; DCX Case Study", IEEE Third International conference on Information and Communication Technologies: From Theory to Applications, page1-5, 7-11 April, 2008.
6. Ahonen. H, Heinonen.O ,Klemettinen.M, Verkamo, A.Inkeri, "Applying data mining techniques for descriptive phrase extraction in digital document collections", IEEE International Forum on Research and Technology Advances in Digital Libraries, page no:2-11, 22-24 April 1998.
7. Ning Zhong, Yuefeng Li, Sheng-Tang Wu, "Effective Pattern Discovery for Text Mining," IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 1, pp. 30-44, January, 2012.
8. Gkoulalas-Divanis, "A Hybrid Approach to Frequent Itemset Hiding", 19th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2007, 29-31 Oct. 2007, Volume 1, Page(s): 297 – 304, 2007.
9. Rakesh Agrawal, Tomasz Imielinski, Arum Swami, "Mining Association Rules between sets of items in large databases", Proceedings of the ACM SIGMOD international conference on Management of data, Volume 22 Issue 2, Pages 207-216, 1993.
10. Feng Yucai, "Association Rules Incremental Updating Algorithm", Journal of Software, Sept., 1998.
11. Qiankun Zhao and Sourav S. Bhowmick, "Association Rule Mining: A Survey", Technical Report, CAIS, Nanyang Technological University, Singapore, No. 2003116, 2003.
12. "Privacy-Preserving Data Mining: Why, How, and When", IEEE Security & Privacy, published by the IEEE Computer Society, 2004.
13. Crina Grosan, Ajith Abraham, Monica Chis, "Swarm Intelligence in Data Mining", Studies in Computational Intelligence Volume 34, page no 1-20,2006.
14. Andrew Secker, Alex A. Freitas, Jon Timmis, "A Danger Theory Inspired Approach to Web Mining", Artificial Immune Systems, Lecture Notes in Computer Science Volume 2787, page no 156-167, 2003.
15. Marco Dorigo, Mauro Birattari, and Thomas Stutzle, "Ant Colony Optimization - Artificial Ants as a Computational Intelligence Technique", IRIDIA – Technical Report Series, Technical Report No. TR/IRIDIA/2006-023, September 2006.

16. Handl J, Knowles J, Dorigo M. , “Ant-based clustering and topographic mapping”, Artificial Life, MIT Press Journals, volume 12 , Issue 1 – winter, page no35-62, 2006.
17. Marco Dorigo, Gianni Di Caro and Luca M. Gambardella, “Ant Algorithms for Discrete Optimization”, Artificial Life, MIT Press Journals, Spring 1999, Vol. 5, No. 2, Pages 137-172, Posted Online March 11, 2006.
18. P. Mathiyalagan, S. Suriya and S.N. Sivanandam, “Hybrid enhanced ant colony algorithm and enhanced bee colony algorithm for grid scheduling”, International Journal of grid and Utility Computing, volume 2, Issue 1, page no: 45-58, 2011.
19. S.Suriya, Dr.S.P.Shantharajah, “Selective Marketing for Retailers to promote stock using improved Ant Colony algorithm", International Journal of Engineering and Technology (IJET), Volume 5 Issue 5, Page No: 3715-3725, October- November 2013.
20. Chengqi Zhang, Shichao Zhang, “Association rule mining: models and algorithms”, Book from ACM DL, Springer-Verlag Berlin, Heidelberg 2002, ISBN:3-540-43533-6.
21. Sotiris Kotsiantis, Dimitris Kanellopoulos, “Association Rules Mining: A Recent Overview”, GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), pp. 71-82, 2006.
22. Weiyang Lin, Sergio A. Alvarez, Carolina Ruiz, “Efficient Adaptive-Support Association Rule Mining for Recommender Systems”, Data Mining and Knowledge Discovery, Volume 6, Issue 1, page no 83-105,January 2002.
23. Lei Ji, Baowen Zhang, Jianhua Li, “A New Improvement on Apriori Algorithm”, International Conference on Computational Intelligence and Security, Page No:840 – 844, Volume 1, November 2006.
24. Sheng Chai, Jia Yang ; Yang Cheng, “The Research of Improved Apriori Algorithm for Mining Association Rules”, International Conference on Sevice Systems and Service Management, Page 1-4, June 9 -11,2007.
25. Grahne G, Zhu, “Fast algorithms for frequent itemset mining using FP-trees”, IEEE Transactions on Knowledge and Data Engineering , Volume 17, Issue 10, page No:1347 – 1362, October 2005.
26. Shengwei Li, Lingsheng Li, Chong Han, “Mining closed frequent itemset based on FP-Tree”, IEEE International Conference on Granular Computing, Page No: 354 – 357, August 17-19, 2009.
27. M. Dorigo. Optimization, Learning and Natural Algorithms (in Italian). PhD thesis, Dipartimento di Elettronica e Informazione, Politecnico di Milano, IT, 1992.
28. Binitha S, S Siva Sathya, “A Survey of Bio inspired Optimization Algorithms”, International Journal of Soft Computing and Engineering (IJSCE), ISSN: 2231-2307, Volume-2, Issue-2, May 2012.
29. Yi Zhang, Dingding Wang, Tao Li, “LIBGS: A MATLAB software package for gene selection”, International Journal of Data Mining and Bioinformatics, Inderscience Publishers, Volume 4, Number 3, Pages 348-355, 2010.
30. Remco R. Bouckaert , Eibe Frank, Mark A. Hall, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten, “WEKA---Experiences with a Java Open-Source Project”, The Journal of Machine Learning Research, Volume 11, Pages 2533-2541, 2010.
31. Baharak Goli and Geetha Govindan, “WEKA – A powerful free software for implementing Bio-inspired Algorithm”, Technical Trends, CSI Communications, Page No:9-12, December 2012.