



## Frequent Itemset Mining using Ant Colony Optimization

Sanjay Patel<sup>1</sup>, K. Kotecha<sup>2</sup>

<sup>1</sup> Computer Engineering Department, Vishwakarma Government Engineering College, Opposite Sangath Mall, Near Visat Cross Roads, Chandkheda, Ahmedabad, Gujarat, India

<sup>2</sup> Director, Institute of Technology, Nirma University, Sarkhej - Gandhinagar Highway, Post: Chandlodia, Ahmedabad, Gujarat, India

**Abstract**—Data mining extracts knowledge or useful information and discovers correlations or meaningful patterns and rules from large databases. The discovered knowledge can then be applied in the corresponding field to increase the working efficiency and improve the quality of decision making. Artificial Intelligence algorithm powered data mining opens up new ways to discover data. Genetic Algorithm, Ant Colony Optimization, Neural Networks, Particle Swarm Optimization etc. are the Computational Intelligence techniques. Existing Frequent Pattern mining algorithms concentrated on the Candidate Generation and Test and Pattern Growth Approach. Graph based, specially undirected graph based methods are unusual. Ant Colony Optimization (ACO) is a meta-heuristic which has shown better results to solve NP-Hard Problem. Here in this paper ACO is applied to solve the Frequent Pattern Mining problem. The results and comparison of the method is shown at the end of the paper.

**Index Terms**—Artificial Intelligence, Ant Colony Optimization, Data Mining, Frequent Pattern Mining

### I. INTRODUCTION

Data mining principles have been for many years, but with the advent of big data, it is even more prevalent. Data mining, the extraction of hidden predictive information from large databases, is a dominant new technology with great potential to help companies focus on the most important information. Data mining is ready for application in the business community because it is supported by three technologies that are now sufficiently mature:

- Huge Collection of Data
- Dominant multiprocessor computers
- Data mining algorithms

Data mining techniques include Association, Classification, Clustering, Prediction, Sequential Patterns, Decision Trees, Combinations and long-term (memory) processing [15]. Association is better known and most familiar and straightforward data mining technique. Here, you make a simple correlation between two or more items, often of the same type to identify patterns. Market Basket Analysis is the common example for Association Rule Mining, where people buying habits can be tracked. Association rule mining is a two step process. Frequent Itemset generation is the first step and two identify the rules from the generated itemsets is the second step. The second part is straight forward. Most of the research is concentrated on the first part. The AIS and SETM are the early algorithms, started to find the frequent patterns. Afterwards Apriori becomes the first algorithm which works on the Candidate Generation and Test approach. Apriori Algorithm is based on anti-monotone property: “If an itemset is frequent, then all of its subsets must also be frequent”. Due to large number of candidate generation and repeatedly scanning of the database, it is suitable only for small number of transactions [9]. J. Pei et al [16] suggested FP-Growth which is a part of Pattern-Growth approach, works on the principle of frequent pattern mining without candidate generation and test. This method requires two database scan. Association Rule Mining works on the trial and error basis. Minimum support needs to be changed if the results are not satisfactory. In the said situation, the process starts from the scratch which wastes the time and effort.

To overcome the drawbacks of FP-Growth method W. Cheung [43] suggested the concept of CATS (Compresses Arranged Transaction Sequences) Tree. CATS Tree works on the principle of Interactive mining, “Build once, mine many”. The problem with CATS Tree is swapping, merging, deletion of the node. It takes too much time. Also storage is the constraint for this type of tree structure. Researchers assumed unlimited amount of memory but in practical applications this is not possible. Due to unexpected database growth, the mechanism which supports incremental mining will be very much essential otherwise the complete mining procedure needs to be started from the scratch. With this idea in mind CKS Leung et al. [24] proposed a new tree structure called Can-Tree (Canonical Tree). In comparison to CATS Tree, here all the items are ordered according to some specific ordering, for example Lexicographical or Alphabetical. Available data can be in any order, to arrange the data in some specific sequence is also a typical task. This is the additional overhead of the mechanism. The tree size is also dependent on the items appearing in the transactions.

P. Deepa Shenoy (2003) et al. [25] came with the novel idea for frequent pattern mining, based on directed graph based mechanism. The approach includes four phases:

- Compression/Decompression,
- Generation of frequent one item,
- Construction of Association Graph and Building Path Matrix and
- Traversing the path matrix to find the frequent patterns.

In the year 2008, R.S. Thakur et al. [30] proposed a graph based algorithm. Their focus was on reducing the database scan and avoiding candidate generation and test. During scanning the database, it creates a directed graph, which is stored in memory as an adjacency matrix. In directed graph based algorithm, it is not clear that how to identify the relationship between the items from the different transactions. FP-Growth-Graph was proposed by Vivek Tiwari et al. in 2010 [41]. Basically it consists of three step procedure:

- making a graph,
- pruning a graph and
- mining graph.

The performance of the algorithm depends on the minimum support. If the minimum support is large, FP-Growth runs faster, but for the low minimum support, FP-Graph runs faster. As the literature is concerned, all the graph based algorithms are working on the direction based algorithms. Ant Colony Optimization (ACO) [29], [42] is meta-heuristic which has shown a tremendous effect to solve the NP-Hard problems. So many combinatorial optimization problems have been resolved by this meta-heuristic with good results. The basic requirement to apply ACO on any problem is the undirected graph with certain parameters on the edge between two nodes. K. Kotecha et al. [20],[21] made the graph for the transaction database but the numbers of parameters were not fixed on the edge between two nodes. Here try has been made to represent the graph with exact number of parameters for all edges and also ACO is applied to find the solution. The paper is organised as follows. The next section discusses the back ground theory. In Section III proposed method is discussed. Section IV shows the implementation results. Conclusion and Future Scope are discussed in the Section V.

## II. RELATED WORK

Apriori is first known frequent pattern mining algorithm which was proposed in 1993. Since then a number of algorithms have been proposed for improving the performance of Apriori-based algorithms [1], [8], [14], [23], [27], [45]. The bottleneck of Apriori-like approaches comes from the requirement of multiple database scans and a large number of candidate patterns, many of which are proved to be infrequent after scanning the database. There are also some popular frequent pattern mining algorithms, which are based on the pattern growth paradigm. Among these methods, the FP-growth [16] and H-mine [17] algorithms are two representative ones. Their main difference lies in the data representation structures. FP-growth adopts a prefix tree structure while H-mine uses a hyperlinked array based structure. The pros and cons of these algorithms are available in [16], [17]. Association Rule Mining works on the trial and error basis so the initial mining may not give the accurate results. Based on the drawbacks of the FP-Growth algorithm, W.Cheung et al. [4] proposed a new data structure known as CATS Tree. The data structure allows data mining without referencing the original dataset. At the same time, the data structure is completely insensitive to and unaffected by user parameters. Users can perform data mining repeatedly with different parameters without having to rebuild the structure. The details and advantages as well as disadvantages of the proposed data structure are available in [4]. Based on some observations the CKS Leung et al. [24] investigated a different approach known as Can-Tree (Canonical Ordered Tree). In this approach items are arranged in lexicographical order. Also later on the same authors have made some improvements in the existing approach by introducing CanTries. The data structure and mining techniques in detail are available in [3] and [24]. Another tree based approaches are CTU-Mine [2], CT-ITL [44], and CP-Tree [36]. The methods discussed above are tree based methods. Directed graph based methods are also available for frequent pattern mining. The basic graph based approach proposed by P. Deepa Shenoy (2003) et al. [25]. The objective of the approach is to apply the compression on the databases and identify frequent itemsets at single and multiple levels. The algorithm is efficient in generating frequent itemsets and is faster than Apriori. It uses very small storage space and can handle very large databases. R.S. Thakur et al.(2008) [30] proposed a graph based algorithm. The sole feature of this algorithm is that it scans the entire database only once. Directed Graph is created during scanning the database. The algorithm will not have drawbacks of the FP-Growth Approach. The details of the algorithm are available in [30]. Based on FP-Growth the Graph based FP-Growth-Graph was proposed by Vivek Tiwari et al.[41] in 2010. FP Growth tree uses tree for arranging the items before mining, where more than one node can contain single item which causes repetition of same item and needs more space to store many copies of same item. Page fault is also one of the causes of it. PGMiner, a novel graph based algorithm proposed by H.D.K. Moonesinghe et al. [12] for mining frequent closed itemsets. The first step in this approach is to construct a prefix graph structure and decomposing the database to variable length bit vectors. Mining process in detail is available in [12]. As the literature is concerned, all the graph based algorithms are working on the direction based algorithms. Ant Colony Optimization (ACO) is meta-heuristic which has shown a tremendous effect to solve the NP-Hard problems. So many combinatorial optimization problems have been resolved by this meta-heuristic with good results. The basic requirement to apply ACO on any problem is the undirected graph with certain parameters on the edge between two nodes. Here Graph with certain parameters between two nodes is made using the concept of Prime Number Multiplication and GCD Concept. ACO is applied on the graph to find the frequent itemsets.

### III. PROPOSED METHODOLOGY

The method described here is based on undirected graph. Each item is a node and items in a single transaction are linked with the edges means every item node is linked i.e. edge is made; with every other item node in a single transaction. If number of unique items (nodes) in a transaction is n then it has  $nC2$  edges in graph. Edge has two weights (attributes) namely fid(Frequent item set Identifier) and c (Counter). Each transaction is associated with a unique prime number starting from the first prime number 2. Consider the following database (Table 1) to create a graph.

TABLE 1 TRANSACTION DATABASE

Transaction No.	Prime id	Items in each transaction
1	2	A, B, C
2	3	B, C
3	5	B, C D
4	7	A,B, C, D
5	11	A, B, C, D, E

#### 3.1 Data Structures

- Open source jgrapht API available at <http://www.jgrapht.org> to implement graph.
- For traversal of graph ,used two LIFO (Last In First Out) stacks:
  - rootstack: to store currently traversing node and nodes that are in process of traversal.
  - edgestack: to store the directly connected nodes of the node on top of rootstack.
- List to store Directly Connected Nodes (DCN) from a node for traversal.
- List to store answer.

#### 3.2 Applying ACO

Proposed following rules in terms of ACO to find frequent item sets from the given data set.

- **Pheromone update on Graph construction**

$$\tau_{ij}^{k+1} = \eta_{k+1} \cdot \prod_{k=0}^k \tau_{ij}^k$$

where,

$k$  = represents number of ants visited edge connecting node i and node j.

$i, j$  = items as nodes in graph.

$\eta$  = Pheromone value deposited on edge which is the unique prime number assigned to the transaction to which the item belong.

Initialize, for  $k = 0, \tau_{ij}^0 = 1$ .

Assume a completely connected graph between items as nodes that are present in a transaction. According to the level of parallelism available, you can start with a number of ants. Here, consider number of ants equal to number of transactions. As the transaction arrives, ants travel in the completely connected path between the items in a single transaction.

For instance, following are the transactions with unique prime id assigned.

Prime	Transaction
2	A, B, C
3	B, C
...	...

Initialization,

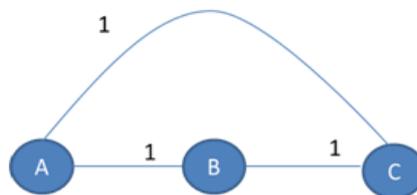


Figure 1: Initialization Process for transaction 1

Pheromone update on first transaction with items A, B and C.

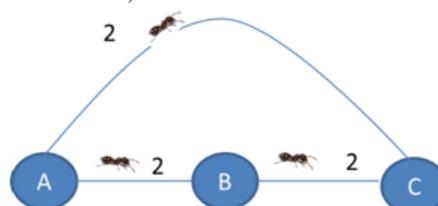


Figure 2: Pheromone Update after Transaction 1

On second transaction, items B and C.

$$\tau_{BC}^0 = 1$$

$$\tau_{BC}^1 = 2 \cdot 1 = 2$$

$$\tau_{BC}^2 = 3 \cdot (\tau_{BC}^1 \cdot \tau_{BC}^0) = 3 \times 2 = 6 \text{ and so on.}$$

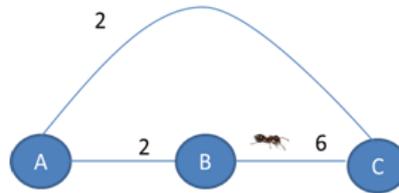


Figure 3: Pheromone Update after Transaction 2

- **Check whether to choose the edge while traversing graph for finding solution.**

Calculate the count of itemsets to determine its fitness. Function at the start of traversal (initialization from first node)

$$p = 1 - \frac{m}{f(\tau_{ij})}$$

$$S(p) = \frac{1}{1 + e^{-p}}$$

Where,

$m$  = required minimum support

$f(\tau_{ij})$  = factors of edge weight (pheromone) connecting nodes  $i$  and  $j$  that represents count of item set  $i$  and  $j$ .

If,  $S(p) \geq 0.5$ , path will be chosen starting from node  $i$ .

else, do not choose that path because it does not have required minimum support.

Note that, Higher the value of  $S(p)$ , more is the priority to start traversal from that node.

Consider a graph constructed as below (Figure 4).

- **Red path shows possible paths starting from node A.**

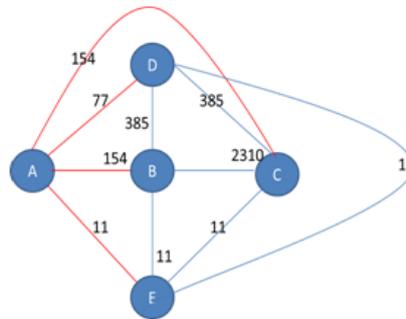


Figure 4: Possible Paths from Node A.

Suppose minimum support ( $m$ ) = 3.

$$p(\tau_{AB}) = 1 - 1 = 0$$

putting this value in equation,

$$S(p) = 0.5,$$

so as per given condition, it seems a better path to follow. In the same way,

$$p(\tau_{AD}) = 1 - \frac{3}{2} = -0.5$$

$S(p) = 0.38$ , so this path will not be followed.

$p(\tau_{AC}) = 1 - 1 = 0 \rightarrow S(p) = 0.5$ , this is also a path to be followed.

$p(\tau_{AE}) = 1 - \frac{3}{1} = -2 \rightarrow S(p) = 0.12$ , this seems unsuitable path.

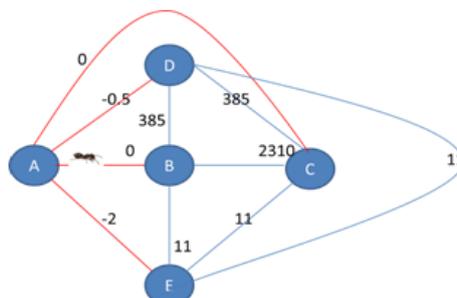


Figure 5: The  $p(x)$  values for all nodes from Node A

The next function is,

$$S(p) = \frac{f(g(\tau_{ij}, \tau_i))}{f(g(\tau_i))}$$

where,

$f(x)$  = Number of factors of  $x$  that represents count of item set

$g(\tau_{ij}, \tau_i)$  = Greatest Common Divisor of  $(\tau_{ij}, \tau_i)$

**Threshold value for function:**

$$S(p') = \frac{m}{f(g(\tau_i))}$$

Path is selected iff,  $S(p) \geq S(p')$

$i$  = Source node as well as partial solution got so far.

For instance, A-B-C is the path. Initially ' $i$ ' is A and ' $j$ ' is B. Then if AB is fit then ' $i$ ' is AB and ' $j$ ' is C.

$j$  = destination node.

Now ' $i$ ' is AB and ' $j$ ' can be C, D or E.

First, will find  $g(\tau_{BC}, \tau_{AB}) = g(2310, 154) = 154$ .

$f(154)=3$

function  $S(p) = 3/3=1$

Threshold is  $S(p') = 3/3=1$

So path is good.

Ant will select the path.

Now ' $i$ ' is ABC and ' $j$ ' is D or E. Let's calculate for ABC, D.

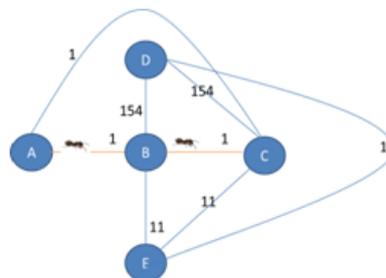


Figure 6: Min\_sup (m)  $\geq 3$  from Node A

$g(\tau_{CD}, \tau_{ABC})=77, f(77)=2,$

Threshold  $S(p')$  is  $3/3=1$

Function  $S(p) = 2/3=0.66$ , which is less than threshold that tells it is not a good path.

Now, calculate for ABC,E but it is also not good.

So final answer will be ' $i$ ' which ant will remember when it will return. Similarly, next path AC will be selected till all the paths are visited. This process is repeated from all the nodes in graph.

#### IV. EXPERIMENTAL RESULTS

The method described here is based on undirected graph. Try has been made to find the relevant methods such that proposed method is compared with the existing same type of method. As far as the literature is concerned, undirected graph based method for frequent pattern mining was not available, so here comparison of the proposed method with the existing tree based and directed graph based approaches is shown. All the experiments are performed on the same platform with Intel Core 2 Duo Processor.

The first comparison (Figure 7) is made between the CanTree (Tree Based Method), FP-Growth-Graph (Directed Graph Based Method) and the ACO-Based-Grpah (Proposed Methodology, Undirected Graph Based method) for Minimum support (In Percentage) Versus Time (In Minutes). ACO-Based-Graph performance is better when the minimum support is high. When it becomes low the performance gets degraded because of internal calculation of GCD and Prime Number multiplication. FP-Growth-Graph gives an average performance in all situations. CanTree performance is not better when the minimum support is high.

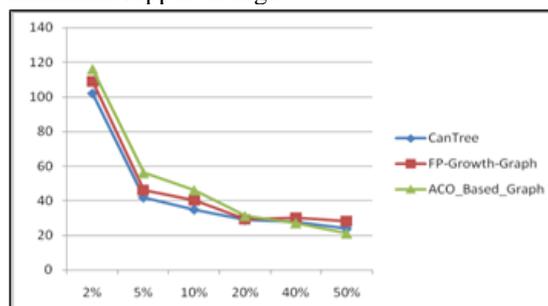


Figure 7: Minimum Support (%) Vs Time (Minutes)

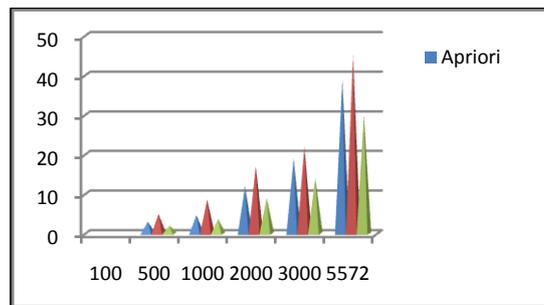


Figure 8: Number of Transactions Vs. Time

In figure 8, comparison of Apriori, Prime-Based-Graph and ACO-Based-Graph is shown for Number of transaction Vs. Time. Prime-Based-Graph gives worst performance in all situations.

The next comparison (Figure 9) is made between FP-Growth-Graph, Prime-Based-Graph and ACO-Based-Graph for the same parameters Minimum Support (%) Vs. Time (Minutes). In comparison with Prime-Based-Graph, ACO-Based-Graph gives good performance. This comparison is made for another dataset available on IBM Library. Also the performance depends on the items available in the transaction as well as transaction length. Also it depends on the order of items in the transaction.

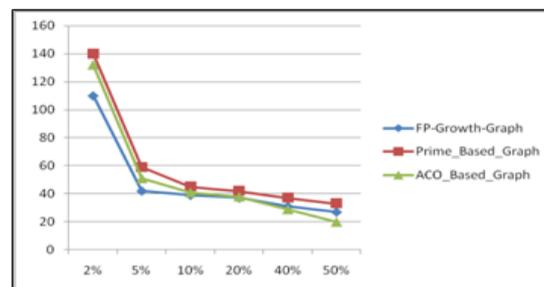


Figure 9 : Minimum Support(%) Vs. Time (Minutes)

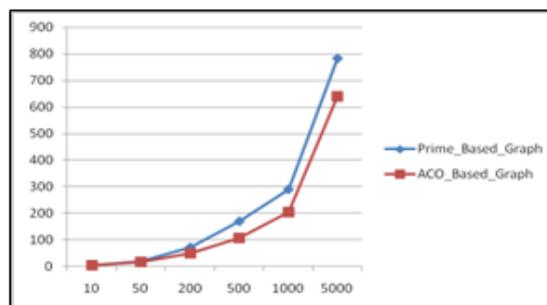


Figure 10 : Database (In Transaction) Vs Memory (MB)

In Figure 10, the comparison is available between the Prime-Based-Grpah and ACO-Based-Graph for The transaction numbers Vs. Memory Occupied by both the methods. When the Ant will complete the route from one node, it releases the memory so the ACO-Based-Graph occupies less memory as compared to Prime-Based-Graph.

## V. CONCLUSION AND FUTURE SCOPE

Experimental result shows that for small datasets and large minimum support the ACO based frequent pattern mining gives good results. As the minimum support decreases the intermediate calculation takes more time. Also when the number of transaction increases the prime number multiplications and the GCD calculation becomes more complex. Memory usage can be reduced after finding the alternatives of the prime number multiplication and GCD calculation for frequent itemset mining. Also parallel processing can reduce calculation. The framework of Artificial Bee Colony leads to the concept of parallel Processing. Artificial Bee Colony can be considered as an alternate for parallel processing of this concept as a future work.

## REFERENCES

- [1] A.J.T. Lee, R.W. Hong, W.M. Ko, W.K. Tsao, H.H. Lin (2007), "Mining spatial association rules in image databases", *Information Sciences* 177 (7) (2007) 1593–1608.
- [2] Alva Erwin, Raj P.Gopalan and N.R.Achuthan(2007), "CTU-Mine: An efficient High Utility Itemset Mining Algorithm using Pattern growth approach" *Seventh International conference on Computer and Information Technology*,2007.
- [3] Carson Kai-Sang Leung et al. (2007), "CanTree: a canonical- order tree for incremental frequent-pattern mining", *Knowledge and Information Systems*, 2007, 11(3), Page 287-311.

- [4] Cheung W.(2002), "Frequent Pattern mining without candidate generation or support constraint." Master's thesis, University of Alberta, 2002, SPRING '03, doi.ieeecomputersociety.org/10.1109/IDEAS.2003.1214917.
- [5] Christian Borgelt(2005), " An Implementation of the FP-growth Algorithm" OSDM'05, August 21, 2005, Chicago, Illinois, USA.
- [6] Elena Baralis et al. (2013), " P- Mine: Parallel itemset mining on large datasets", Data Engineering Workshops (ICDEW), 2013 IEEE 29th International Conference on, Date 8 – 12 April, 2013.
- [7] En Tzu Wang et al. (2011) , " Mining Frequent Itemsets over distributed data streams by continuously maintaining a global synopsis" Data Mining and Knowledge Discovery, Springer, September 2011, Volume 23, Issue 2, pp 252-299
- [8] G. Chen, Q. Wei (2002), "Fuzzy association rules and the extended mining algorithms", Information Sciences 147 (1–4) (2002) 201–228.
- [9] G. Piatetsky-Shapiro et al. (1991), " Knowledge Discovery in Databases" , MIT Press, 1991.
- [10] Geoffrey I. Webb(2010), " Self-Sufficient Itemsets: An Approach to Screening Potentially Interesting Association Between Items" , ACM Transactions on Knowledge Discovery from Data, Volume 4, No.1, Article 3, January 2010.
- [11] Guimei Liu et al. (2013), " A Flexible Approach to finding Representative Pattern Sets", IEEE Transactions on Knowledge and Data Engineering, 04 Feb. 2013. IEEE computer Society Digital Library. IEEE Computer Society.
- [12] H.D.K. Moonesinghe, S. Fodeh, P.N. Tan(2006), "Frequent Closed Itemset Mining using Prefix Graphs with an efficient Flow-Based Pruning Strategy" , Proceedings of the sixth International Conference on Data Mining (ICDM '06) IEEE Computer Society.
- [13] Irina Tudor(2008), "Association Rule Mining as a Data Mining Technique" , Buletinul Universitatii Petrol Gaze din Ploiesti, Seria Matematica-Informatica-Fizica , Vol. LX, No. 1/2008, page 49 – 56.
- [14] J. Han, G. Dong, G. Yin(1999), "Efficient mining of partial periodic patterns in time series database" , In: Proceedings of IEEE International Conference on Data Mining, 1999.
- [15] J. Han and Micheline Kamber, Book : "Data Mining, Concept and Techniques", 578 pages, books.google.co.in.
- [16] J. Han, J. Pei, Y. Yin(2000). "Mining frequent patterns without candidate generation." SIGMOD 2000.
- [17] J. Pei, J. Han, et al.(2001), " H-Mine: Hyper-Structure Mining of Frequent Patterns in Large Databases." ICDM 2001.
- [18] Jien Pei, "Pattern-Growth methods for frequent pattern mining " Ph. D. Thesis, Simon Fraser University, 2002.
- [19] Jilles Vreeken et al.(2011), " KRIMP : mining itemsets that compress" Data Mining and Knowledge Discovery, Springer, July 2011, Volume 23, Issue 1, pp 169-214.
- [20] K. Kotecha et al. (2011), "Frequent Pattern Mining Using Graph Based Approach" , International Journal of Computer Science Research and Application 2011, Vol. 01, Issue. 02
- [21] K. Kotecha et al. (2013), "Incremental Frequent Pattern Mining using Graph based approach" , International Journal of Computers & Technology, Volume 4 No. 2, March-April, 2013, ISSN 2277-3061
- [22] Liang Wang et al.(2012),"Efficient Mining of Frequent Itemsets on Large Uncertain Databases", IEEE Transactions on Knowledge and Data Engineering, Vol. 24, No. 12, December 2012.
- [23] Q. Li, L. Feng, A. Wong(2005), "From intra-transaction to generalized inter-transaction: landscaping multidimensional contexts in association rule mining", Information Sciences 172 (2005) 361–395.
- [24] Q. I. Khan, T. Hoque and C.K. Leung(2005), " CanTree : A Tree structure for Efficient Incremental mining of Frequent Patterns". Proceeding of the Fifth International conference on Data Mining (ICDM'05),csdl2.computer.org/.../&toc=comp/proceedings/icdm/2005/2278/00/2278toc.xml&DOI=10.1109/ICDM 2005.
- [25] P. Deepa Shenoy et al.(2003) , " Compress and Mine: An Efficient Graph Based Algorithm to Generate Frequent Itemsets" , Advance Computing and Communicating Society , 2003.
- [26] P. M .Kanade, et al. (2007), " Fuzzy Ants and Clustering", IEEE Transactions on Systems, Man and Cybernatics, Volume 37, No. 5, September 2007.
- [27] R. Agrawal, R. Srikant(1994) , "Fast algorithms for mining association rules, in: Proceedings of the 20th International Conference on Vary Large Data Bases", 1994, pp. 487–499.
- [28] R. J. Kuo et al. (2007) , " Association Rule mining through the ant colony system for National health Insurance Research Database in Taiwan" , An International Journal of Computers and Mathematics with applications 54, Science Direct, pp 1303-1318.
- [29] R. S. Parpinelli et al. (2002), " Data Mining with an Ant Colony Optimization Algorithm", IEEE Transactions on Evolutionary Computing, Volume 6, No. 4, August 2002.
- [30] R. S. Thakur, R.C. Jain and K.R. Pardasani(2008), " Graph Theoretic Based Algorithm for Mining Frequent Patterns" Neural Networks, IJCNN (IEEE world Congress on Computational Intelligence). 1-8 June, 2008.
- [31] Rajnish Dass and Ambuj Mahanti, " An efficient heuristic search for Real-Time frequent pattern mining".International Conference on System Sciences – 2006, ieeexplore.ieee.org/iel5/10548/33362/01579371.pdf

- [32] Rajanish Dass et al. (2006) ,” An Efficient Algorithm for Real-Time Frequent Pattern Mining for Real-Time Business Intelligence Analytics”, Proceedings of the 39th International Conference on System Sciences- IEEE, 2006
- [33] Rakesh Agrawal et al. (1993), “ Mining Association Rules between set of items in large databases”, In proceedings of ACM SIGMOD Conference on Management of Data (SIGMOD ‘ 93) pages 207- 216, May 1993.
- [34] S. Shankar et al.(2009),”Utility Sentient Frequent Itemset Mining and Association Rule Mining : A Literature Survey and Comparative Study” , International Journal of Soft Computing, ISSN: 1453 – 2277 Issue 4 (2009), pp 81-95.
- [35] S. Shankar et al. (2009), “ A Novel Utility Sentient Approach for Mining Interesting Association Rules” , IACSIT International Journal of Engineering and Technology, Volume 1, No. 5, December 2009.
- [36] S. K. Tanbeer et al. (2008), “ Efficient single – pass frequent pattern mining using a prefix tree” , Information Sciences 179 (2008) ,
- [37] Shailesh Kumar, et al.(2012) , “ Logical Itemset Mining “ IEEE 12th International Conference on Data Mining Workshops, December 2012.
- [38] Tianyi Wu et al.(2010), “ Re-examination of interestingness measures in pattern mining : a unified framework” Data Mining and Knowledge Discovery, Springer, November 2010, Volume 21, Issue 3, pp 371-397
- [39] Tias Guns et al. (2013), “ k- Pattern Set Mining under Constraints” , IEEE Transactions on Knowledge and Data Engineering, Vol. 25, No. 2, February 2013.
- [40] Vincent S. Tseng et al.(2013), “ Efficient Algorithms for mining high utility Itemsets from Transactional Databases” , IEEE Transactions on Knowledge and Data Engineering, Vol. 25, No. 8, August 2013.
- [41] Vivek Tiwari et al., (2010) , Association rule mining: A Graph Based Approach for mining Frequent Itemsets, 2010 International Conference on Networking and Information Technology, 978-1-4244-7578-0 © 2010 IEEE
- [42] W.J. Jiang et al.(2005), “ A Novel Data Mining Algorithm based on Ant Colony System” ,Proceedings of the fourth International Conference on Machine Learning and Cybernetics, Gaungzhou, 18-21 August 2005.
- [43] William Cheung et al. (2003), “ Incremental Mining of Frequent Patterns without candidate Generation or Support Constraint”, IDEAS’03, doi.ieeecomputersociety.org/10.1109/IDEAS.2003.1214917.
- [44] Y. G. Suchayo et al. (2003), “ CT-ITL: Efficient Frequent Itemset Mining using a Compressed Prefix Tree with Pattern Growth” , 14th Australasian Database Conference Adelaide, Australia (ADC 2003).
- [45] Y.J. Tsay, J.Y. Chiang (2004), “An efficient cluster and decomposition algorithm for mining association rules”, Information Sciences 160 (1-4) (2004) 161–171.

#### AUTHOR’S BIOGRAPHY



**Sanjay Patel** - B.E. in Computer Engineering from North Gujarat University, Gujarat, India in 2002 and M.E. (Computer Engineering) from Sardar Patel University, Vallabh Vidyanagar, Gujarat in 2008. He is currently working as an Assistant Professor in Computer Engineering Department at Vishwakarma Government engineering college, Chandkheda, Ahmedabad, Gujarat. Before that he has also worked as a lecturer in Computer Engineering department at Sankalchand Patel college of Engineering, Visnagar, Gujarat. His research interests are Data Mining and Artificial Intelligence.



**Dr. Ketan Kotecha** – B.E. in Electronics from Sardar Patel University and M.Tech. and Ph.D. from IIT, Bombay. He has over 15 years of experience in teaching UG students of Computer Engineering and over 9 years teaching PG students. He has published many International Journal papers. He visited several countries like USA, Canada, Europe, Singapore, Hongkong and South Africa to present papers in International Conferences. He has guided several masters’ thesis in Computer Engineering and guiding 6 Ph. D. students. He served as the Principal of G H Patel College of Engineering and Technology, Vallabh Vidyanagar for three years before joining as a Director of Institute of Technology, Nirma University. His research interests include Artificial Intelligence & Computer Algorithms