An Efficient Algorithm for Extracting Infrequent Itemsets from Weblog

Brijesh Bakariya[1] and Ghanshyam Thakur[2]
[1]Department of Computer Science and Engineering, I.K. Gujral Punjab Technical University, India
[2]Department of Computer Applications, Maulana Azad National Institute of Technology, India

Abstract: *Weblog data contains unstructured information. Due to this, extracting frequent pattern from weblog databases is a very challenging task. A power set lattice strategy is adopted for handling that kind of problem. In this lattice, the top label contains full set and at the bottom label contains empty set. Most number of algorithms follows bottom-up strategy, i.e. combining smaller to larger sets. Efficient lattice traversal techniques are presented which quickly identify all the long frequent itemsets and their subsets if required. This strategy is suitable for discovering frequent itemsets but it might not be worth being used for infrequent itemsets. In this paper, we propose Infrequent Itemset Mining for Weblog (IIMW) algorithm; it is a top-down breadth-first level-wise algorithm for discovering infrequent itemsets. We have compared our algorithm IIMW to Apriori-Rare, Apriori-Inverse and generated result in with different parameters such as candidate itemset, frequent itemset, time, transaction database and support threshold.*

Keywords: *Infrequent itemsets, lattice, frequent itemsets, weblog, support threshold.*

Received September 6, 2014; accepted March 24, 2016

1. Introduction

The collection of minimum frequent itemset might be important. An example can be drowning in drug analysis, market basket analysis, business analysis, etc. Most of the criteria are based on support and confidence, here the support consists number of times pattern occur in the transaction databases, moreover it's a frequency of itemset in a transactional database and the confidence determines the proportion value that shows how frequently a part of the pattern (premise), occurs among all the records containing the whole transaction dataset. For example, if the pattern has to satisfy the minimum support then that pattern is considered as frequent pattern or frequent pattern on the contrary, these patterns have to satisfy maximum support then that pattern considered as in frequent pattern or infrequent pattern [1, 2]. Infrequent patterns can be used in different domains such as biology, medicine and security [9, 15], etc. For example, in a clinical database analysis one can discover infrequent patterns that will help doctors to make decisions about the clinical care. As one can observe, each type of patterns expands the data seeking for specific types of knowledge. In other types of patterns 'infrequent and frequent' patterns that can be mined. Any item set is found interesting only when its frequency is less than the maximum threshold or more than the minimum threshold [6, 8, 10]. For searching 'frequent and infrequent patterns' is an NP- Hard problem whose complexity is exponential. This is complex from the computational point of view. A few algorithms have been developed which can search the frequent and infrequent patterns in NP-Complete time or we can say it's solved such problems in polynomial time. The algorithms which efficiently search the 'frequent patterns' are not necessarily be searching for 'infrequent patterns too'. Algorithms to search for both the patterns are infrequently available apart from the 'Rarity' these are many such problems which exist in different data mining algorithms. We have taken log data is collected which gets available at the Internet traffic archive [16]. This log data later partitioned on the basis of its attributes and we have chosen two field timestamp and web page after applying preprocessing techniques [4, 5].

2. General Terms and Definitions

2.1. Power Set

Let A be a set, then the power set of A is P (A) give by P (A) = {S: S (A}. Here A is the set of n elements, then the number of elements in P (A) is 2^n or n [P (A)] $=2^n$. For example, if {a, b, c} the P (S) = {ø, {a}, {b}, {c}, {a, b}, {b, c}, {a, b, c}}. Here S has 3 elements so p (S) has 23 =8 elements.

2.2. Lattice

A non empty set P, together with binary relation R is said to from a partially ordered set or a post if the following conditions are satisfied.

1. Reflexitivity: - aRa ∀ a∈ P.
2. Anti Symmetry: - If aRb and bRa then a = b (∀ a, b∈ P).
3. Transitivity: - aRc, bRc then aRc (∀ a, b, c ∈ P).

2.3. Infrequent Itemset

In this section, we provide definitions of key terms that explain the concepts frequent and infrequent itemset, let A be the collection or set of items entailed by database records, e.g. the set of items a consumer collects in a shopping complex, according to market basket analysis it is referred to as an itemset. Moreover, let I = $\{i_1, i_2 ... i_n\}$ be a set of n different elements called items. Let the database DB is a collection of transactions over I, T is associated with each and every transaction and Tid is a unique index for each transaction. The subset of itemset X= $\{i_a, i_b, i_c ... i_z\} \in I$ and its length consist a number of itemset in X. A z-length item set consist transaction in DB with different itemset and length z. The frequency (Number of occurrences) of an itemset X called support count of X, and it is denoted by Supp (X).

Frequent and infrequent itemset are depend on f_s and r_s where f_s is a frequent support count threshold and r_s is a infrequent support count threshold and $f_s < r_s$. Moreover a particular itemsets are said to be frequent if and only if Supp(X)$\geq f_s$ and infrequent if and only if Supp(X) $\leq r_s$. The support count of superset of an itemset is related to its subsets itemset. Let we take two itemset A and B such that A$\subset$ B, the frequency of A itemset is at least B frequency, or we can say A is part of B then Supp(A) $\geq$ Supp(B), $\forall$ A$\subset$ B.

2.4. Property

2.4.1. Downward Closure Property

If an itemset is frequent then all its subsets must be Frequent, is usually used to mine all frequent itemsets from a large database. If {milk, bread, tea} is frequent, so is {milk, bread} i.e., every transaction having {milk, bread, tea} also contains {milk, bread}.

2.4.2. Anti-Monotonocity Property

If an itemset is Infrequent then all its supersets must be Infrequent, which is usually used to mine all infrequent itemsets from a large database is a very complex task, for example, if {computer, radio} is infrequent or infrequent, so is {computer, radio, television} also infrequent i.e., every transaction not having {computer, radio} also not contains {computer, radio, television}. Define f_s=3 and r_s=2, so each infrequent itemset is also infrequent itemset. We have mentioned the support count in this figure1. We have joined set of all infrequent itemset in a semi-lattice or we can say it is closed under join operation, i.e., A and B infrequent itemset the A and B is also infrequent. On the contrary, if A, B are infrequent then it does not contain A $\cap$ B or it does not meet in semi-lattice

Table 1. Transactional table.

DATASET					
Tid	**P_1**	**P_2**	**P_3**	**P_4**	**P_5**
1	√	√	√	√	
2	√	√	√	√	
3	√	√	√		√
4	√		√	√	
5		√			
6	√		√		
7	√				

In Table 1 we have shown a transaction table for five items and total 7 transactions included in the dataset. Each transaction consists a set of items. Here we consider the items as web pages.

3. Related Work

Association Rule Mining (ARM) is one the data mining technique [3, 11], In ARM there are two approaches as follows first is Apriori and the second is FP-Growth mining [7, 13, 14]. For searching frequent and infrequent itemset, it's a different approach, whereas the problem frequent itemset mining. Many solutions have been developed, but apart from that, to mine infrequent itemset if very hard task and research on searching a infrequent itemset mining is still going on.

3.1. Infrequent Itemset Mining

Various types of algorithm have been proposed for infrequent itemset mining and it is different from frequent pattern mining algorithms. Infrequent itemset consist those itemset that do not occur frequently but it may also generate interesting rules, so if a infrequent pattern consists high confidence rule then it should not be discarded completely. Koh and Rountree [10] proposed a more efficient algorithm name as Apriori-Inverse, which finds perfectly sporadic rules and imperfectly sporadic rules (irrelevant) without generating all the unnecessarily frequent items. They use three parameters in Apriori-Inverse such as Fixed Threshold, Adaptive Threshold, and Hill Climbing. In Apriori-Inverse finds all perfectly sporadic rules much more quickly than Apriori. Szathmary *et al*. [15] proposes two algorithms Minimal Infrequent Generators (MRG)-Exp and A Infrequent Itemset Miner Algorithm (ARIMA). First is MRG-Exp used to finding minimal infrequent generators, we focus on frequent itemsets generators in lattice. Second ARIMA is to get all infrequent itemsets from minimum rate itemset. Tsang *et al*. [18] propose a tree structure approach RP-Tree for mining a subset of infrequent association rules and an information get component that helps to identify the more interesting association rules. RP-Tree, examines all infrequent-item nodes in the initial tree, and all nodes that have less support than a infrequent-item are infrequent items themselves, RP-Tree must find all infrequent item itemsets.

Adda *et al*. [1] approach and algorithm name as

Apriori for Infrequent And Non-present Item-set Mining (ARANIM), ARANIM discover of non-present patterns and infrequent patterns using infrequent item-set mining and they have also proposed a framework to represent the different categories of patterns based on the frequency constraint which by means of an instantiation process leads to the representation of frequent, infrequent and non-present pattern mining problems. Troiano and Scibelli [17] propose Rarity, a top-down breadth-first level-wise algorithm; they explore the power set lattice from the top, reaching the border line of non-infrequent itemsets, this approach is applied in Rarity.

4. Algorithm

Apriori-Rare [15] and Apriori-Inverse [10] are the for discovering infrequent itemset but it is time and space consuming and these above algorithms are not able to mine both frequent and infrequent itemsets. Moreover, we propose an algorithm such as Infrequent Itemset Mining for Weblog (IIMW). In this algorithm use three different data structures in C, F and R. Here C is a candidate itemsets list, F frequent itemsets list and R is infrequent itemsets list.

4.1. Infrequent Itemset Mining for Weblog (IIMW)

Algorithm 1: Infrequent Itemset Mining for Weblog (IIMW)

Input: Dataset (D)
Output: Infrequent Itemset Collection (R)

Take server Web Log Data (WLD)
Preprocess WLD and remove the extension from the URL
Choose Timestamp (T) and Web Page (WP)
T & WL∃: SWL and SWL ∈ WLD
Divide a slot of SWL and calculate the frequency of each page in a slot individually.
Create a Dataset (D) contain time slot, web pages and its frequency
Find out highest length itemsets (p) and assign to n where [(p) ∀D]
len_n= highest length (p) ∀ D // Find out highest length itemsets (p) and assign to len_n variable
For (all itemsets p ∈D) do
Store C (len (t)) // Store all records to candidate list
End For
For (l= l_n to 1) do
If C (l)≠ ø then
For (all ip ∈ C(l)) do
If (supp (ip)> min_supp) then
Remove is from C (l)
Add is to F (l) // Store records in to frequent list
Else
Add is to R(l) // Store records in to infrequent list
If (len(ip)) >1 then
For (all psub ∈ subsets (ip) || length (psub) = l – 1) do
If (psub ∉F) then
Add sub to C (len(psub)) // Store records in to candidate list according to length of subset
For all ip ∈ F(l) do
If length (ip) >1 then
For k= l – 1 to 1 do
For all c ∈C (k) do
cip= c∩ip
Remove cip from C (len(cip)) when cip ∈C
Add cip to F (length (cip))
End for

First of all declare the candidate list C (l) and in this list contains those itemset which is having highest length. All infrequent l-itemset are passed by database to count l-itemsets support. In the first step frequent list (F) and infrequent list (R) are empty. This algorithm IIMW starts from the top of the lattice, which contains longest itemsets after that calculate l_n where l_n is the length of longest itemsets and selects those highest length itemsets and keeps it in the candidate list. For each l= l_n to 1, this algorithm is considering the candidate itemset ip∈C (l). If supp (ip) is greater than minimum threshold mt_f hen it move into frequent list F (l) and if itemsets (ip) is less than or equal to mt_f then those itemsets are infrequent and moved into the infrequent list (R). If the length of subset is l-1 then that itemsets are infrequent then it is assign to C (l-1) and after that it scan frequent itemset list F (l) so that each known itemset $fk_i \in F$ (l) and compare fk_i to smallest candidate sets $s_k \in C(w)$ with w < l. An intersection of $ist_{ik} = fk_i \cap s_k$ determine in order to find common sub-itemset and we can say those itemset which have been derived by fk_i are frequent and moved into frequent itemset list, at the last level (l=1) single itemset present. Even an algorithm stops early when C (l) is empty. The number of occurrences of each itemset can be calculated through support measure. IIMW follows downward closure property and anti-monotonicity property. In this property prove that all subsets of frequent itemset are frequent and all superset of infrequent itemset is infrequent. By applying this above property support can be a measure of itemset according to support.

$$F_{ip} = \sum_{gi \in Sset(ip)} F_{gi} \quad (1)$$

$$Supp(ip) = \sum_{k=l_{ip}}^{l_n} \frac{F_{ip}(K)}{(k - l_{ip})!} \quad (2)$$

Where *gi* is the generic itemset belonging to the set of ip super itemset Sset (*ip*).F contains set of itemset and l_n is the length of itemset. Suppose we have to calculate support of p_1p_3 then firstly overall path from p1p3 can be calculated by using Equation (1). Path is p_1p_3–$p_1p_2p_3$–$p_1p_3p_5$–$p_1p_3p_4$ and the level of p_1p_3 (0, 0, 0, 1, 0, 0); $p_1p_2p_3$ (0, 3, 0, 0, 0, 0); $p_1p_3p_5$ (0, 1, 0, 0, 0, 0); $p_1p_3p_4$ (0, 2, 1, 0, 0, 0) and the sum of all path of p_1p_3 is (0, 6, 1, 1, 0, 0).

$$Supp(p_1p_3) = \sum_{k=2}^{4} \frac{F_{ip}(k)}{(k-2)!} = \frac{1}{0!} + \frac{1}{1!} + \frac{6}{2!} = 5 \quad (3)$$

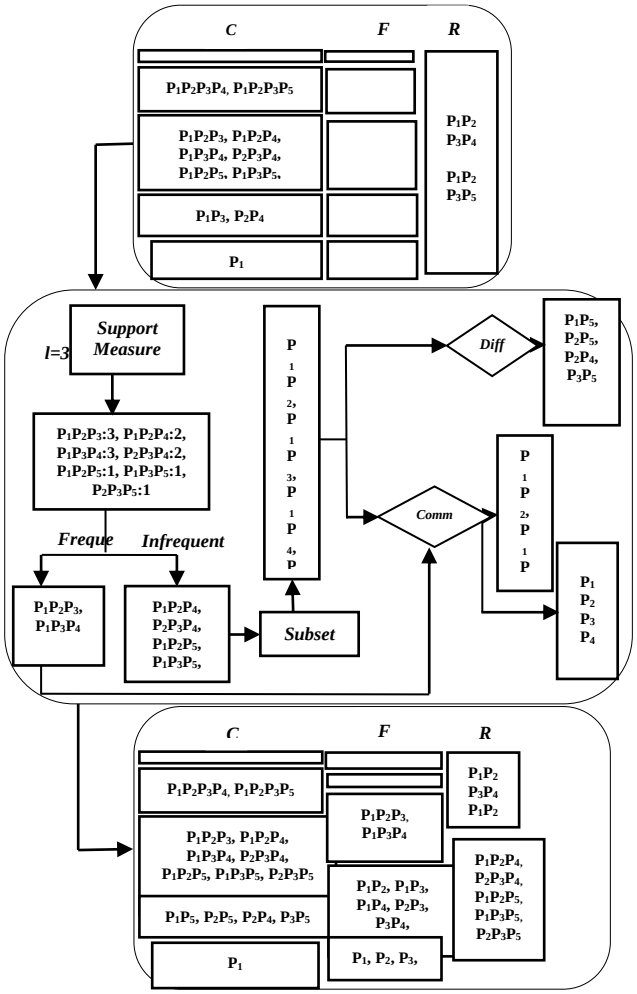

Figure 1. Evaluation of C (3) from l=3.

In Figure 1 provides a processing, in this process C (3) is calculated. In this procedure, all frequent itemset are shift into the Frequent list F (3) and rest of items are shift into the Infrequent list R (l). All values at different levels could be calculated and moved into F and R. An itemset contain frequent and infrequent list according to minimum support threshold which have been predefined.

5. Experimental Analysis

In this section we compare our algorithm IIMW to Apriori-Inverse [10] and Apriori-Rare [15]. Both algorithms are able to find out infrequent itemset list. An experiment was carried out on Intel Core i3n 2.20 GHz processor with 4 GB of RAM and Windows 7 Home Basic operating system. We use the dataset which is available at The Internet Traffic Archive sponsored by ACM SIGCOMM for the time period 1 July to 31 July1995 [16]. In particular, we are operating on web log records and with the maximum transaction length of 254 items (web pages). In this transaction dataset the web pages is clicked by the user then write an occurrence of web page has been clicked and the web page is not clicked then write 0. For the support threshold we assume the different values of minimum support like 10% to 100%. In Figure 2 calculate the candidate count and our algorithm (IIMW) compare with the Apriori-Rare and Apriori-Inverse algorithm. In this comparison the number of candidate count generated with different support threshold. This process is candidate generation process after this process the frequent and infrequent itemset could be extracted.

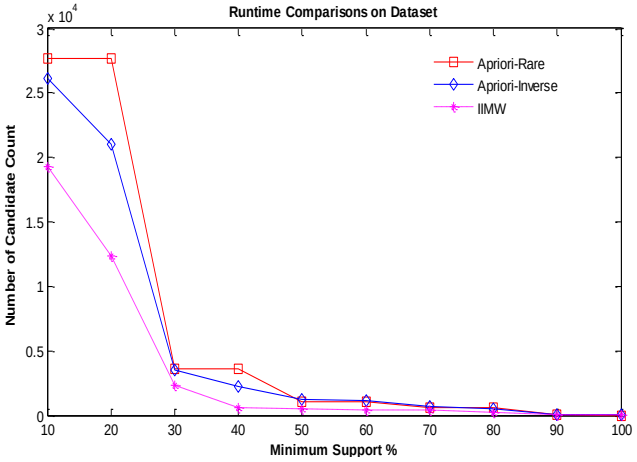

Figure 2. Candidate itemset versus minimum support.

In Figure 3. IIMW extract number of infrequent itemset with different minimum support (10% to 100%) and compare with Apriori-Rare and Apriori-Inverse.

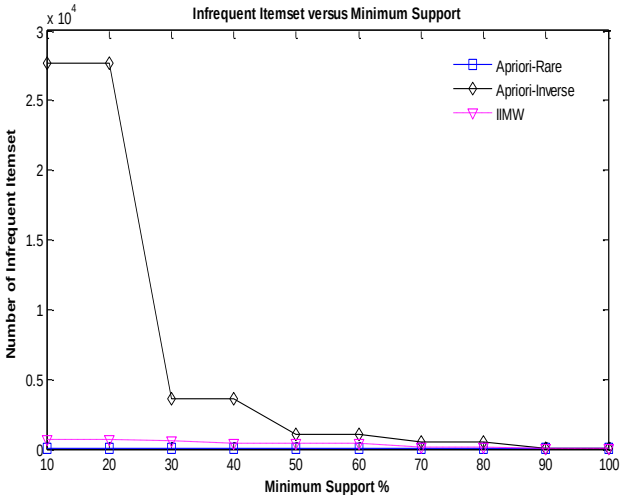

Figure 3. Infrequent itemset versus minimum support.

In Figure 4 we measure in IIMW with different number of transactions and execution time of transaction and compare with Apriori-Rare and Apriori-Inverse algorithm, this algorithm generate candidate counts and infrequent itemset. In our algorithm IIMW generate candidate count, frequent patterns and infrequent patterns. In IIMW we don't count those infrequent itemset which is having 0 frequencies because in the log file or web data; the web page is not visited. We only accept those infrequent itemset which length is 1 or greater than 1.

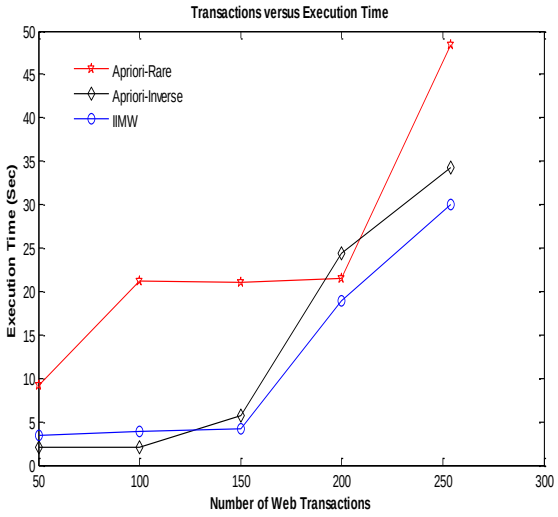

Figure 4. Transactions versus execution time.

6. Conclusions

In this paper face the problem of discovering infrequent itemset, most of research focused on finding frequent itemset, but infrequent itemset mining could be expose interesting or valuable knowledge. An algorithm Apriori is the most standard algorithm from frequent itemset mining and most of algorithm refers this algorithm. We have also taken the inspiration from Apriori but it is different from this. For finding infrequent itemset we used power set and lattice traversal approach. In this approach follow Top-Down mechanism, in this larger itemset at top place and further it will up to bottom place. By applying this approach computation of support count for smaller itemsets is easier rather than larger itemset. In this paper we discovered infrequent itemset as well as frequent itemset, for discovering this itemset we proposed an algorithm IIMWD is used for extracting total candidate itemset, infrequent itemset and frequent itemset; infrequent pattern could be useful for business rules, statistical analysis, web advertisement etc.

References

[1] Adda M., Wu L., White S., and Fengr Y., "Pattern Detection with Rare Itemset Mining," *International Journal on Soft Computing, Artificial Intelligence and Applications*, vol. 1, no. 1, pp. 1-17, 2012.

[2] Agrawal R. and Srikant R., "Fast Algorithms for Mining Association Rules," *in Proceedings of 20th International Conference on Very Large Data Bases*, San Francisco, pp. 487-499, 1994.

[3] Agrawal R., Imielinski T., and Swami A., "Mining Association Rules between Sets of Items in Large Databases," *in Proceedings of ACM SIGMOD International Conference on Management of Data*, New York, pp. 207-216, 1993.

[4] Bakariya B., Mohbey K., and Thakur G., "An Inclusive Survey on Data Preprocessing Methods Used in Web Usage Mining," *in Proceedings of 7th International Conference on Bio-Inspired Computing: Theories and Applications*, India, pp. 407-416, 2013.

[5] Bakariya B., Mohbey K., and Thakur G., "An Inclusive Survey on Data Preprocessing Methods Used in Web Usage Mining," *in Proceedings of 7th International Conference on Bio-Inspired Computing: Theories and Applications*, Gwalior, pp. 407-416, 2013.

[6] Han J., Pei J., and Yin Y. "Mining Frequent Patterns without Candidate Generation," *in Proceedings of ACM SIGMOD International Conference on Management of Data*, Texas, pp. 1-12, 2000.

[7] Han J., Pei J., Yin Y., and Mao R., "Mining Frequent Patterns without Candidate Generation: a Frequent-Pattern Tree Approach," *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53-87, 2004.

[8] Huang D., Koh Y., and Dobbie G., "Infrequent Pattern Mining on Data Streams," *Data Warehousing and Knowledge Discovery Lecture Notes in Computer Science*, Vienna, pp. 303-314, 2012.

[9] Iwanuma K., Takano Y., and Nabeshima H., "On Anti-Monotone Frequency Measures for Extracting Sequential Patterns from a Single Very Long Data Sequence," *IEEE Conference on Cybernetics and Intelligent Systems*, Singapore, pp. 213-217, 2004.

[10] Koh Y. and Rountree N., "Finding Sporadic Rules Using Apriori-Inverse," *Advances in Knowledge Discovery and Data Mining Lecture Notes in Computer Science*, Nanjing, pp. 97-106, 2007.

[11] Liu B., Hsu W., and Ma Y., "Mining Association Rules with Multiple Minimum Supports," *in Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, pp. 337-341, 1999.

[12] Pei J., Han J., Lu H., Nishio S., Tang S., and Yang D., "H-Mine: Fast and Space-preserving Frequent Pattern Mining in Large Databases," *IEEE Transactions*, vol. 39, no. 6, pp. 593-605, 2007.

[13] Prati R., Monard M., Andre C., and Carvalho L., "A Method for Refining Knowledge Rules Using Exceptions," *Electronic Journal of Informatics and Operations Research*, vol. 27, no. 4 pp. 53-65, 2004.

[14] Song M. and Rajasekaran S., "A Transaction Mapping Algorithm for Frequent Itemsets Mining," *IEEE Transactions on Knowledge and*

Data Engineering, vol. 18, no. 4, pp. 472-481, 2006.

[15] Szathmary L., Napoli A., and Valtchev P., "Towards Infrequent Itemset Mining," *in Proceedings of 19th IEEE International Conference on Tools with Artificial* Intelligence, Patras, pp. 305-312, 2007.

[16] The Internet Traffic Archive, available at: http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html, Last Visited, 2013.

[17] Troiano L. and Scibelli G., "A Time-Efficient Breadth-First Level-Wise Lattice-Traversal Algorithm To Discover Infrequent Itemsets," *Data Mining and Knowledge Discovery*, vol. 28, no. 3, pp. 773-807, 2014.

[18] Tsang S., Koh Y., and Dobbie G., "Finding Interesting Infrequent Association Rules Using Infrequent Pattern Tree," *Transactions on Large-Scale Data- and Knowledge-Centered Systems VIII Lecture Notes in Computer Science*, pp. 157-173, 2013.

Brijesh Bakariya received Graduation degree from Barkatullah University Bhopal M.P. in 2005, and Post Graduation Degree in Computer Applications from Devi Ahilya Vishwavidyalaya Indore M.P. in year 2009. He received Ph.D. Degree in the Department of Computer Applications, Maulana Azad National Institute of Technology Bhopal M.P. in 2016. He is Assistant Professor in Department of Computer Science and Engineering, I.K. Gujral Punjab Technical University (IKGPTU) Jalandhar, Punjab. He has been teaching since 2009 and guiding M.Tech/ Ph.D students. In the mean time he published many research papers in SCI publications in the area of Data Mining, Image Processing, and Social Networking. He has attended various short term training programs, refresher course, workshops and seminars. He is a member of the IACSIT, APCBEES, APCBEES and UACEE.

Ghanshyam Thakur has received BSc degree from Dr. Hari Singh Gour University Sagar M.P. in 2000. He has received MCA degree in 2003 from Pt. RaviShankar Shukal University Raipur C.G. and PhD degree from Barkhatullah University, Bhopal M.P. in year 2009. He is Assistant Professor in the department of Computer Applications, Maulana Azad National Institute of technology, Bhopal, M. P. India. He has eight year teaching and research experience. He has 26 publications in national and international journals. His research interests include Text Mining, Document clustering, Information Retrieval, Data Warehousing. He is a member of the CSI, IAENG, and IACSIT.