International Journal of Engineering & Technology, 7 (3.27) (2018) 621-626

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Sliding Window Approach to Mine Negative and Positive Regular Patterns in Incremental Databases Using Vertical Data Format

N.V.S. Pavan Kumar[1]*, K. Rajasekhara Rao[2]

[1]*Department of CSE, Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India*
[2] *Director, Usharama College of Engineering, Vijayawada, Andhra Pradesh, India*
**Corresponding author E-mail: nvspavankumar@gmail.com*

Abstract

Incremental databases are the repositories of most emerging realistic data from ecommerce sites and other sources. They are typical in nature as new transactions are added to the database along with the progression in time. Regular patterns are more advanced and reliable as they describe not only occurrence frequency but also occurrence behavior. Finding negatively associated positive patterns is very complex process because of search space and the size of the database. These not overlapping patterns play a vital role in decision making by extracting complex hidden knowledge from the transactional databases. Sliding window progresses with time leaving the old transactions from one end and keep on including new transactions from other end. Vertical format of database is very much handy in finding regular itemset. There was no much effort made earlier by the researchers in this area of KDD. Hence we have developed an algorithm INC_Nprism to find all the negative and positive regular itemset from incremental databases using vertical format with a sliding window. Unlike some earlier algorithms we need not construct any tree structure with this approach. Also multiple scans of database are also not required with this approach. Experimental results proved that our algorithm works efficiently and yields most satisfying results.

Keywords: *Incremental Mining; Negative Association Rules; Negative Patterns; Regular Patterns;*

1. Introduction

S k Tanbeer et al [4] proposed a tree structured algorithm for regular patterns. Frequent patterns are the fundamental approach in data mining concepts. They are discovered based on support count and correlation frame work in transactional data bases. These patterns are no longer satisfied the requirement of finding meaningful patterns. Periodic patterns are a well attempted technique to find patterns in temporal databases. These methods are developed for time-series databases and sequential databases. Still there is refinement in this approach proposed by the authors using a tree structure named as Regular Pattern tree (RP tree). In this method a user defined regularity threshold is used as the qualifying measure for any pattern to be regular. Maximum of the periods must be less than the regularity threshold.
Rakesh Agarwal et al [1] described about mining association rules between sets of items in large databases. They developed an algorithm on transaction database of traditional market basket analysis. This algorithm uses pruning mechanism of Apriori algorithm. To process the queries functionalities are required. They have been enhanced in their work. With the help of this work system one could answer the queries with consequent and antecedent. For this they ignored the confidence factor of earlier support confidence frame work. This also useful in shelf management in super markets easily. If the total set of rules that are generated for an itemset Ix in Y with k item then all the subsets X containing k-1 items are antecedents. Y-x is the set of consequents. For valid rules, confidence is calculated as support of y divided by support of x and should satisfy the given threshold.
Zaki at al [16] introduced vertical format of database. This is the transpose orientation of the traditional transaction databases. In transaction database, the tuples are of form transaction number and itemset as two attributes where as in vertical format the table is of two columns, itemset and the corresponding transactions.
Yi-ming et al [17] developed algorithm to find frequent patterns using vertical format of transaction database.
Jiawei han et al [4] in their earlier research stated about candidate set that it is expensive to generate candidate sets when the patterns are prolific in nature in other words loosing patterns. Then they proposed FP tree refers to frequent pattern tree. This is an extension of prefix tree. This follows prefix tree approach to accommodate crucial compressed information about frequent patterns. This is useful in finding complete frequent patterns by fragmenting them. Based on divide and conquer technique they developed a strategy to split the big problem into small divisions. This avoids candidate generation at each level by fragmenting the data base. This approach avoids generating not necessary candidate sets which are necessarily not realistic by making the search confined to a set of items rather than all possible combinations at each level and hence results in reducing search space. This can be described in other words as confined search in conditional database. This tree based approach certainly works efficiently than the precious expensive algorithms and remained as landmark in the frequent patterns as well as regular itemset mining.

Table 1: Transaction database

TrnId	ItemSet
Trn1	I_{101} I_{103} I_{105} I_{109} I_{114}
Trn2	I_{105} I_{108} I_{111} I_{112}
Trn3	I_{102} I_{103} I_{104} I_{107} I_{113}
Trn4	I_{104} I_{105} I_{106} I_{112} I_{114}
Trn5	I_{101} I_{103} I_{105} I_{106} I_{109} I_{110}
Trn6	I_{105} I_{108} I_{111} I_{112} I_{114}
Trn7	I_{102} I_{103} I_{106} I_{107} I_{112} I_{113}
Trn8	I_{104} I_{105} I_{106} I_{114}
Trn9	I_{101} I_{103} I_{105} I_{109} I_{111}
Trn10	I_{105} I_{108} I_{111}
Trn11	I_{102} I_{103} I_{107} I_{108} I_{113} I_{114}
Trn12	I_{104} I_{105} I_{106} I_{110}
Trn13	I_{101} I_{103} I_{105} I_{108} I_{109}
Trn14	I_{105} I_{108} I_{111} I_{112} I_{114}
Trn15	I_{102} I_{103} I_{107} I_{111} I_{112} I_{113}
Trn16	I_{104} I_{105} I_{106}
Trn17	I_{101} I_{103} I_{105} I_{106} I_{109}
Trn18	I_{105} I_{108} I_{110} I_{111}
Trn19	I_{102} I_{103} I_{107} I_{113} I_{114}
Trn20	I_{104} I_{105} I_{106} I_{110}
Trn21	I_{101} I_{103} I_{104} I_{105} I_{109} I_{110} I_{114}

Eya Ben Ahmed et al [15] focussed on the structure of data warehouse. According to the mining is performed on the data from data warehouse. While presenting the data from transactional database to warehouse, data is compressed (aggregated) at multiple levels of granularity. Regular mining algorithms ignore this issue given less priority to this aspect. Hence they wanted to work on the multi-level granularity and to combine them to cubes of different dimensions and successfully derived cyclic patterns in their approach.
Xie Zhi-Jun et al [7] proposed one pass algorithm after Agarwal's one pass algorithm. In their work, they focussed on memory efficiency and accuracy. In their algorithmic approach FIET frequent itemset efficient tree is constructed to maintain equivalence classes. GLB and LUB are two measures used to divide the frequent itemsets into equivalence classes. It is obvious that the number of frequent itemsets is much higher than the number of equivalence classes.
Diana Martin et al [8] developed MOPNAR algorithm. This algorithm is multi objective and works on large databases evolves reduced set of positive and negative quantitative association rules. According to them large databases suffer from scalability and complexity. Earlier algorithms focused on binary data rather than quantitative data. But the real world data contains quantitative data only. Also many algorithms focus on positive quantitative association rules rather than negative Quantitative Association Rules. For complex problems both evolutionary algorithms EA genetic algorithms GA are suitable. Maximization of the objectives, interestingness and comprehensibility and of course performance are the three major aspects covered in their work. Very strong rules only extracted during execution. Comprehensibility is the property that an association rule should be easy to understand. Otherwise user will unlikely to use a complex rule which is not understood.

Table 2: Vertical database

Itemset	TrnIds	Periods	P_{max}
I_{101}	1 5 9 13 17 21	1 4 4 4 4 4	4
I_{102}	3 7 11 15 19	3 4 4 4 4 2	4
I_{103}	1 3 5 7 9 11 13 15 17 19 21	1 2 2 2 2 2 2 2 2 2 2 2	2
I_{104}	3 4 8 12 16 20 21	3 1 4 4 4 4 1	4
I_{105}	1 2 4 5 6 8 9 10 12 13 14 16 17 18 20 21	1 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1	2
I_{106}	4 5 7 8 12 16 17 20	4 1 2 1 4 4 1 3 1	4
I_{107}	3 7 11 15 19	3 4 4 4 2	4
I_{108}	2 6 10 11 13 14 18	2 4 1 2 1 4 3	4
I_{109}	1 5 9 13 17 21	1 4 4 4 4 4	4
I_{110}	5 12 18 20 21	5 7 6 2 1	7
I_{111}	2 6 9 10 14 15 18	2 4 3 1 4 1 3 3	4
I_{112}	2 4 6 7 14 15	2 2 2 1 7 1 6	7
I_{113}	3 7 11 15 19	3 4 4 4 4 2	4
I_{114}	1 4 6 8 11 14 19 21	1 3 2 2 3 2 4 2	4

2. Related Work

2.1 Incremental/Web Mining

Tanbeer at al [18] introduced regular patterns in transactional data bases using transaction databases. They constructed a regular pattern tree and proved the results are more reliable compared to frequent patterns. They used regularity threshold in place of minimum support.
Ahmed, C. F et al [19] used sliding window in their research over data streams to find the frequent patterns. Sliding window is best suitable for both data streams and incremental databases.
Chih-Hsiang Lin et al [20] proposed a sliding window method based on time stamp. In this method they took time stamp as measure and developed time sensitive algorithm.

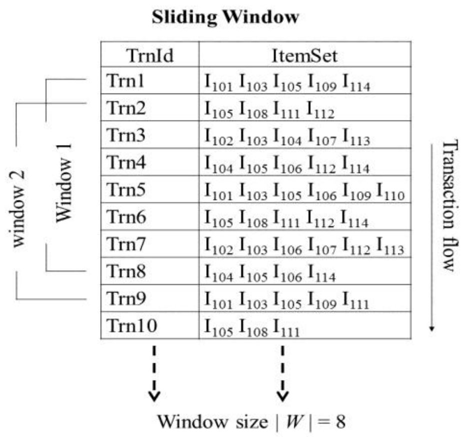

Fig. 1: Sliding window

Farhan Ahmed et al [2] proposed WAS methods (Web Access Sequence) to find non binary occurrences of variables in a web page by each user. This will extract most useful information from web logs by considering non binary variables such as time spent in a web page by each user. Earlier versions do not support two way web accesses and requires more database scans for level wise candidate set generation. They proposed a tree structure IUWAS-tree. This helps in interactive mining in incremental databases. Also for static databases, UWAS tree is proposed. According to the different users spend various timings in different webpages. Earlier frame works assign binary values to the web pages as 0/1 indicating whether a particular web page is accessed or not which is not realistic and cannot access backward references. This is far advanced than WAP mine (Web Access Pattern) algorithm by Pie et al. HUP (High Utility Pattern) mining model was proposed in order to overcome the earlier traditional frequent pattern mining model because they consider only the binary values as measure. In HUP a measure called MEU (Mining Expected Utility) was used. This fails to satisfy Apriori down ward closure property. Also in early page references the chances of overestimation is very high.
Yue-shi lee et al [9] proposed TWU (Transaction Weighted Utilization) to maintain Apriori down ward closure property. Each level of candidate set generation of TWU previous level elements is used.
The idea of interactive mining is about modify the min support required as and when web logs are updated or the structure of web site changes. Users tend to access web sites in an orderly way but this frequency will change depending on the changes made to the website. It is not quite suitable to use the unique support count all

the way. Another point incremental mining refers to the changes in the web logs. Based on time stamps, some of the records will be deleted and every time new sequences of transactions are performed on the website. It could be very much useful to navigate the user with appropriate actions based on their previous history. Full scan (FS), selective scan (SS) and MAFTP (Maintenance of frequent traversal patterns) were the earlier approaches useful in track down the activities of users. The knowledge generated with these algorithms is helpful for the developers to do necessary changes for various actions and options on a website.

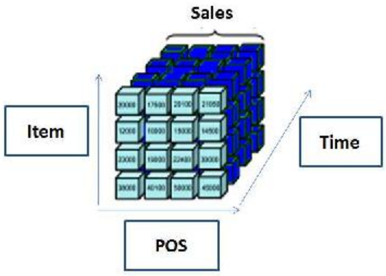

Fig.2: Sales data Cube in data wherehouse

2.2 Web Access Sequential Pattern Mining

Incremental mining is one of its own kinds in pattern mining and also in knowledge data discovery. Incremental databases are difficult to handle as they grow with time. Finding frequent items in these databases keep changing time to time. Sequences found at dome point become not interesting and also new patterns emerge when new data is mined and the data base is updated.

Earlier researchers developed few algorithms namely Apriori All, GSP by Agarwal and Srikanth. Later in 1997 Wang developed suffix tree algorithm. Zaki in 2000 developed SPADE algorithm. Pie et al also developed FreeSpan and Prefix Span. They also developed WAP-tree algorithm to work on single element set sequence. This data base is usually found in e-commerce websites where users click on a single item at any point of time. Also there will be forward and backward to and fro movement by the user in order to know different models and other details of models and add them to the cart.

Ezcife and LV in 2005 developed WAP-Tree, PLWAP-Tree, FM-Minor algorithms for web sequence mining. Key points in web sequential pattern mining in incremental databases lies in the scalability of large data sets by increasing. Spead algorithm also suits for dynamic real time responses. This enhances efficient utilization of I/O operations as well as CPU and memory resources. Along with PLWAP-Tree approach there are several other algorithms such as FS-miner, pattern growth tree also contributed have their own glory in the area of finding association rules of the form x=>y and x ∩ y={ø}.

Yeu-shilee et al [9] in their publications stated that web mining is useful in improving web access. To make it friendlier to the users we need to have certain knowledge from behavior of patterns in web accessing. This is performed on web logs containing the user's access habits. These logs are always increasing and for that we need to understand the incremental data base nature and apply suitable methodology to mine the hidden knowledge. With these results obtained from the web logs one can suggest or direct the end users towards their requirements in and easy and comfortable way by making appropriate options at each and every level of web browsing. This process can be performed in two ways namely static and dynamic. The latter is the better approach as it can interact with the user at each level of browsing. This is also known as interactive mining. In this case the major aspect to consider is the changes made to web pages time to time. This leads to changes in the options of website or the total structure of web site might be updated. Accordingly minimum support is to be updated along with the increase in time sequence of web pages accessed also travels. These sequences are also known as traversal sequences of various lengths. The database of traversal sequences is constituted by allotting a transaction id to each sequence.

Farhan et al [2] in their research work stated that WAS (Web Access Sequence) discovery is one of the interesting and useful aspect in KDD from web logs. Utility based WAS were proposed by them using UWAS algorithm. Here the major issue lies in the realistic aspects like the time spent on a web page are taken into consideration. Also earlier methods depend on binary approach such as a web page is accessed or not. Here this approach is suitable for incremental databases and further for user interaction. Hence this method more useful than static web access mining and ore interactive. This approach also supports multilevel association by generating candidate sets at various levels. This algorithm is based on UWAS tree and IUWAS tree structures. This also covers both forward and backward references to the web pages. This tree structure is built only once but can be mined many times with different mining approaches. In this approach different profit values are attached to each and every webpages unlike earlier algorithms assign equal profit binary values to each webpage.

Earlier HUP mining follows MEU (Mining with expected Utility). But this cannot satisfy downward closure property. In the beginning stages it suffers from over estimation where all combinations satisfy the criteria of candidates and hence this approach is not practical. These algorithms use a fixed threshold. If threshold is to be changed then the entire structure is to be re built. Hence this is against the user tendency. They continuously change their priorities and behavior of accessing the webpage access. Hence the tree should be built only once and as and when the website structure is changed, no need to construct tree with this approach and hence no need of scanning database again and again. In this they used hash table to construct tree and could be easily updated whenever new node enters in to the tree without changing the existing structure.

Lin zhon et al [11] web logs contain gigabytes of data every day about the dynamics of web. These webpages are accessed in sequences. To discover these patterns weblogs are mined and they are named as path traversal patterns. Different users have different habits of accessing webpages. Hence we need a measure to help the IIS for decision making. Depending on how interesting and helpful the webpage, the measure utility is defined. Two-phase utility mining method is best suitable to find high utility path traversal patterns. Result are proven that these paths are well ahead of traditional frequent pattern mining in helping decision making easy to users. Web log contains the information about the IP address, time stamp soon information of each request to the webpage. A sequence contains the order of pages accessed by the user from the beginning. Both subjective and objective values are considered in this algorithm. Usefulness is the measure of utility. In other words the measures like how much time as user spent on a webpage can be a measure of interestingness.

Farhan Ahmed et al [10] in their research stated that the Real world scenarios will not be reflected with binary occurrences approach of webpages. Hence consider the time spent on webpage by user as a utility measure. Apriori algorithm generates too many candidate sets and leads to several database scans in order to find the web traversal path. A novel algorithm EUWPTM was proposed by them is based on divide and conquer rule and divides the search space recursively. As a result of this generated candidate sets will be reduced to a maximum level. WEBMINER is a web mining system useful in applying the data mining techniques on World Wide Web (www). Mining language MINT is used in WUM (Web Utilization Miner) to investigate dynamically specified web interesting patterns. This is a pruning based algorithm to prune the not interesting patterns. Full Scan (FS) and SS (Selective Scan) are also well-known algorithms in this area of research. They reduce I/O by reducing not necessary multiple scans of database. There are several other well known in this area such as MEU (Mining with Expected Utility), UMining-H and so on. MEU so not satisfy downward closure property. UMining-H uses Utility upper bound property is used as pruning strategy.

Closed Patterns, compressed patterns and so on comes under compact frequent patterns. Mining these patterns gives more efficient results than the traditional frequent pattern mining. The changes made on databases reflect changes in sequences. Hence these sequences are dynamic rather than static in nature many times.

LIechang et al [12] proposed CSTrea construction in order to keep downward closure property. IMCSA and IMCSD are two algorithms developed based on this tree. The updates made to the database forces the algorithm to re scan the database right from the beginning in earlier methods. While using this CSTree the updating need not run from scrap and no modifications required to the obsolete nodes. This will improve the efficiency of the algorithm. The pre-FUFP algorithm by Chu-Wei is to find frequent itemset. For this we need to identify candidate sets in the former methods. Later FPtree is an efficient tree data structure to find the patterns successfully. For both the approaches, the transaction database is processed batch wise. Whenever new transactions are included it is necessary to update the tree. For this purpose fast updated FP tree structure was invented. In their research they developed Pre-FUFP algorithm. They maintain a pair of upper support threshold as well as lower support threshold. This allows us not to re scan the database for newly occurred transactions. Hence this is one of the land mark algorithm in finding frequent patterns in incremental databases. This is far better than COFP-Tree (conditional FP-tree), QFP-growth and generalized FP-tree and so on.

Jigyasa Bisaria et al [13] in their research article stated that in incremental databases any infrequent pattern with time stamp can become frequent after a delta time. With t+Δt time stamp. This requires a separate methodology to handle all these infrequent patterns. This approach maintains a limited number of patterns in knowledge base. This incremental sequence extraction ISE method works based on rough sets. According to rough sets the partition should satisfy certain conditions.

1 $y_i \neq ø$

2 $y_i \cap y_j = ø$

3 $V = \bigcup_{i=1}^{k} yi$

4 Then the equivalence class formed by setp2 form a partition in V. in this scenario, they maintain frequent, semi frequent, partially frequent and infrequent in different partition D and D` and establish the relation between them.

Chun -jung chu [14] described utility is one of the criteria of the interestingness of the sequence. As the utility is high sequence is more useful. It is observed that utility be positive value always. Hence finding high utility sequences associated with negative utility is also important aspect in data mining. HUINIU-mine is introduced to find such sequences. This will identify transaction weighted items with high utility values which are few in number.

In our earlier work Vijay kumar at al [21, 22] developed algorithms for finding regular itemset using vertical format and extended to incremental databases [23] and data streams [24] to mine regular frequent patterns in Vertical Format. C. Cornells et al [24] and W. Xindong et al [25] in their research, worked on both positive and negative pattern mining. They have successfully generated positive and negative association rules.

Hence in this paper we were presenting an algorithm to find the negatively associated positive regular patterns. They can be also known as non overlapping patterns or contradicting patterns. We used vertical format of database with a sliding window with different sizes and with different thresholds.

3. Problem Definition and Process

A transactional database TrnDB is constituted by a group of transactions of form Trn={trn1, trn2, trn3…..trnm}. The size of the database is the number of transactions in the relation denoted by |TDB| is m. Let $I = \{i_1, i_2, ... i_n\}$ be a set of items. An itemset or a pattern is a sub set of form $X = \{i_l, ... i_q\} \subseteq I$ where l an q lies between 1 and n say $l \leq l, q \in [1, n]$. All the transactions in TrnDB are of form T=(TrnId, S) where S is a sub set of I and is of the form $S = \{i_a, i_b, i_c \dots i_h\}$ for some a,b,c….h belongs to I. Therefore $S \subseteq I$ and 1<=a,b,c…,h<=n. Let the vertical format of the TrnDB is VDB. The tuples in vertical database are of from V= $\{i_j$, TrnS$\}$ where TrnS={ trnq, trnr, trns…trnt }where all these transactions contains i_j and 1<=q,r,s…t<=m. Let the period is the difference between two consecutive occurrence of an item i_j. All such periods of an itemset S constitutes the set P^S={p1,p2, p3, …. px}. Let $P^S max$ is the maximum value of periods for a given itemset S. Positive regular itemset is defined as $P^S = \{i_u, i_v, \dots i_w\}$ for any given $P^S max < \lambda max\_reg$ (user defined regularity threshold) where $i_u, i_v \dots i_w \subseteq I$. Let NR = $\{P^S, P^Q\}$ be the negative regular itemset where P^S and P^Q are two positive regular itemsets and there exists no common transactions $P^S \cap P^Q = \{\phi\}$. Let the newly added transactions be db+ to TrnDB. The updated database is denoted as UTrnDB, is obtained from TrnDB ∪ db+. Incremental negative regular pattern mining is to discover the complete set of positive regular patterns having regularity not more than λ and negatively related patterns in the updated database UTrnDB.

Boolean FindReg (i_y, $TrnId^{Iy}$, λ_{max_reg},m)
{ // $TrnId^{Iy}_{first}$ and $TrnId^{Iy}_{last}$ are the first and last transactions of I_y
 i_y_First =$TrnId^{Iy}_{first}$-0;
 if (i_y_First> λ_{nax_reg})
  return FALSE;
  i_y_Regularity=i_y_First;
  for all z in $TrnId^{Iy}_{first}$ +1 to $TrnId^{Iy}_{last}$
   { i_y_NextP= $TrnId^{Iy}_{z}$ - $TrnId^{Iy}_{z-1}$;
    if(i_y_NextP >i_y_Regularity) then
     i_y_Regularity=i_y_NextP;
   }
  I_y_NextP=m- $TrnId^{Iy}_{last;}$
   if(i_y_NextP)>i_y_Regularity then
    i_y_Regularity=i_y_NextP;
  if (i_y_Regularity > λ_{max_reg})
   return FALSE
 return TRUE
}

Algorithm INC_PRUNE(I_y,I_j,n)
{
 I_{yj}= I_y∪I_j ;
 $TrnId^{Iyj}$ = $TrnId^{Iy}$∪$TrnId^{Ij}$;
  if (FindReg(I_{yj}, $TrnId^{Iyj}$, λ_{max_reg})= = FALSE)
   prune $I_{yj\,;}$
  else
   {
    VDB= VDB ∪ { I_{yj} , $TrnId^{Iyj}$ };
    n=n+1;
   }
}

Algorithm INC_Result(I_y,I_j)
{
 if (I_{yj} ∈ VDB)
  return;
 if (TrnIdIy∩ TrrnIdIj) = = {ϕ})
  Result=Result ∪I_{yj} ;
 else
 return ;
}

Algorithm INC_NPRISM()
{
// To find first regular item
 for all k in 1 to n
 { if (FindReg(I_k, $TrnId^{Ik}$, λ_{max_reg},m)= = FALSW)
   prune I_y
  else
   break;
 }
 // To find remaining regular items

```
        for  j= k+1 to n
        {
           if  ( FindReg( I_j, TrnId^Ij, λ_max_reg )= = FALSE)
              prune I_j;
           else
          {
               INC_PRUNE(I_k,I_j);
               INC_Result(i_k,i_j);
           }
          }
}
```

4. Results

Experiment Results proven the efficiency of the algorithm in all aspects compared to the previous algorithms.

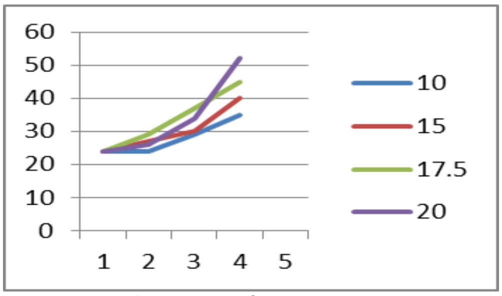

Fig. 3: Time taken in Sec

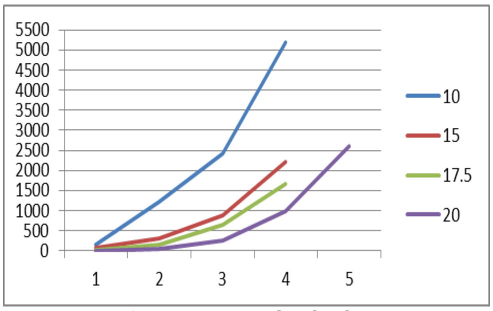

Fig.4: Memory utilized in bytes

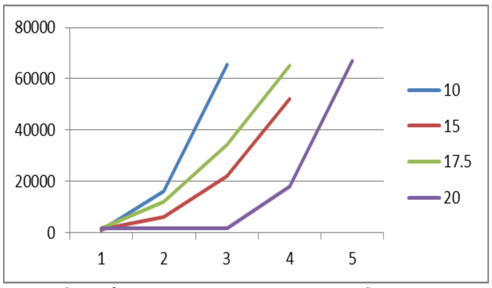

Fig. 5: Number of Negative itemsets (max regularity in percentage)

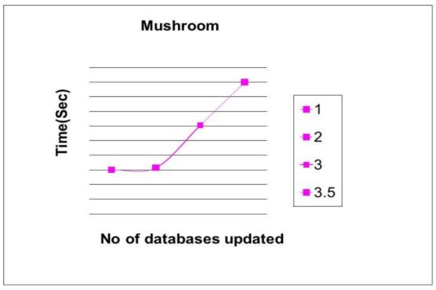

Fig. 6: Execution Time Over Mushroom

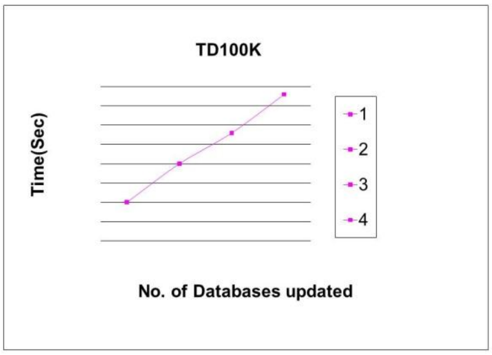

Fig. 7: Execution Time Over TD100K

5. Conclusions

Mining negative patterns is always challenging task as the search space is very high and the data is complex with multiple variables. Incremental databases are dynamic in nature as they grow with time. It is not advisable to construct a tree structure and modify it or re construct it as and when new transactions occur. Also scanning the entire database after each modification to the database with and addition of small amount of new transactions is not wise. Hence we proposed this algorithm without constructing tree structure. With sliding window approach transactions with oldest time stamp will be in effective as the window progresses. In our future work we will add more number of variables with different regularity thresholds. Our algorithm successfully finds all itemset with different size and identifies the relation between them.

Acknowledgement

I would like to express my sincere thanks to G. Vijay kumar for his continuous support in bringing this research successful. Also I extend my gratitude to my friend Prasanth P, technocrat for trouble shooting and improving of efficiency of code.

References

[1] Agarwal, R., Imielinski, T., Swamy, A.N.: Mining Association Rules between sets of Items in Large Databases, ACM, SIGMOD Conference of Management of Data, pp. 207-216 (1993).
[2] Farhan Ahmed, Tanbeer, Byeong-Soo Jeong " A frame work for mining high utility web access sequence " IETE Technical Review 28(1) · January 2011
[3] Agarwal, R., Srikanth, R. Fast algorithms for mining association rules, In Proc. 1994 International Conference on very large databases (VLDBA'94), Santiago, Chile, pp. 487-499, Sept. 1994.
[4] Elfeky, M.G., Aref, W.G., Elmagarmid, A.K. Periodicity detection in time series databases, IEEE Transactions on Knowledge and Data Engineering 17(7), pp. 875-887 (2005).
[5] Han, J., Pei, J., Yin, Y. Mining frequent patterns without candidate generation, In Proc. ACM, SIGMOD, International Conference on Management of Data, 2000, pp. 1-12.
[6] N V S Pavan Kumar, K Rajasekhara Rao "Mining Positive and Negative Regular Item-Sets using Vertical Databases" IJSSST.a.17.32.33, pp. 232-248, 2016.
[7] Xie Zhi-jun Chen Hong Cuiping Li An Efficient Algorithm for Frequent Itemset Mining on Data Streams ICDM 2006: Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining pp 474-491
[8] Diana Martín, Alejandro Rosete , Jess Alcalá-Fdez , Francisco Herrera, "A New Multiobjective Evolutionary Algorithm for Mining a Reduced Set of Interesting Positive and Negative Quantitative Association Rules" EEE Transactions on Evolutionary Computation , Volume: 18, Issue: 1, Feb. 2014
[9] Yue-ShiLeeShow-JaneYen "Incremental and interactive mining of web traversal patterns" Information Sciences Volume 178, Issue 2, 15 January 2008, Pages 287-306

[10] Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanbeer, Byeong-Soo Jeong, and Young-koo Lee “Efficient mining of Utility-Based web path Traversal Patterns”, ISBN 978-89-5519-139-4,Feb.15-18,2009 ICACT 2009.

[11] L. Zhou, Y. Liu, J. Wang, and Y. Shi. "Utility-based Web Path Traversal Pattern Mining", in: Proceedings of the 7th IEEE International conference on Data Mining Workshops, pp. 373-8, 2007.

[12] LeiChang, TengjiaoWang, DongqingYang, HuaLuanc,, Shi-weiTangad, “Efficient algorithms for incremental maintenance of closed sequential patterns in large databases” ,Data & Knowledge Engineering Volume 68, Issue 1, January 2009, Pages 68-106.

[13] Jigyasa Bisaria, Namita Shrivastava, K.R. Pardasan “A Rough Sets Partitioning Model for Mining Sequential Patterns with Time Constraint”, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 2, No. 1, 2009

[14] Chun-Jung, ChuaVincent, S.Tsengb, TyneLianga, “An efficient algorithm for mining high utility itemsets with negative item values in large databases Applied Mathematics and Computation”, Volume 215, Issue 2, 15 September 2009, Pages 767-778

[15] Eya Ben Ahmed, Ahlem Nabli and Fäıez Gargouri “Cyclic Association Rules:Coupling Multiple Levels and Parallel Dimension Hierarchies”, Proceedings of the International Conference on Information and Knowledge Engineering (IKE); Athens 2011.

[16] Zaki, M.J., Karam, G. “Fast Vertical Mining using Diffsets”, SIGKDD’03, August 24-27, 2003, Copyright 2003 ACM 1-58113-737-0/03/0008.

[17] Tanbeer, S.K., Ahmed, C.F., Jeong, B.S. Lee, Y-K, “Mining Regular Patterns in Incremental Transactional Databases, 12th International Asia-Pacific web conference, 2010 IEEE, DOI 10.1109/APWeb.2010.68, pp.375-377.

[18] Yi-ming, G., Zhi-jun, W. A Vertical format algorithm for mining frequent itemsets, IEEE Transactions, pp. 11-13, 2010.

[19] Ahmed, C. F., Tanbeer, S. K., Jeong, B. S., & Lee, Y. K. “An efficient algorithm for sliding window-based weighted frequent pattern mining over data streams”.IEICE Transactions, Vol. 92-D(7),pp. 1369–1381, 2009.

[20] Chih-Hsiang Lin, Ding-Ying Chiu, Yi-Hung Wu, Arbee L. P. Chen, “Mining Frequent Itemsets from Data Streams with a Time-Sensitive Sliding Window”, in Society for industrial and applied mathematics, 2005.

[21] Vijay Kumar, G., Sreedevi, M., Pavan Kumar NVS. Mining Regular Patterns in Transactional Databases using vertical Format, IJARCS, Sep-Oct 2011, pp. 581-583.

[22] Vijay Kumar, G., Sreedevi, M., Pavan Kumar NVS A Vertical Format to Mine Regular Patterns in Incremental Transactional Databases”, Journal Of Computing, Volume 3, Issue 11, November 2011, ISSN 2151-9617

[23] Vijay Kumar, G., Valli Kumari, V., “Incremental Mining for Regular Frequent Patterns in Vertical Format” International Journal of Engineering and Technology, Vol 5 No 2, 2013.

[24] Vijay Kumar, G., Valli Kumari, V., “Sliding Window Technique to Mine Regular Frequent Patterns in Data Streams using Vertical Format”, 2012 IEEE International Conference on Computational Intelligence and Computing Research, 2012.

[25] C. Cornells, Y. Peng, Z. Xing, and C. Guoqing, "Mining Positive and Negative Association Rules from Large Databases," in IEEE Conference on Cybernetics and Intelligent Systems, 2006, pp. 1-6.

[26] W. Xindong, Z. Chengqi, and Z. Shichao, "Efficient mining of both positive and negative association rules," ACM Transactions on Information Systems (TOIS), vol. 22, pp. 381-405, 2004.