ISSN 2347 - 3983
Volume 7, No. 9 September 2019
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter12792019.pdf
https://doi.org/10.30534/ijeter/2019/12792019

On mining Incremental Databases for Regular and Frequent Patterns

NVS Pavan Kumar[1], Dr.JKR Sastry[2], Dr. K Raja Sekhara Rao[3]
[1]Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India, nvspavankumar@kluniversity,in
[2]Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh, India, drsastry@kluniversity,in
3Usharama institute of Engineering and Technology, Andhra Pradesh, India, krr_it@yahoo.co.in

ABSTRACT

Implementation of Incremental databases undertaken for dealing with different data phases and accumulation at regular intervals. The transactions contained in the fixed database and increment for the same could differ a lot based on the time and season during which the increment created, and the original database accumulated. Many algorithms have been in existence for incremental mining databases to derive frequent patterns that yield positive associations. Many applications exist that deal with regular and frequent patterns having negative associations mined from incremental databases. Negative associations are equally important to discover the counter effect of one pattern over the other.

In this paper, an algorithm and implementation of the same using IBM supplied transactional data converted to an incremental database presented that helps to mine regular and frequent patterns having negative associations.

Keywords: Incremental databases, Frequent, and Regular Data Mining, Negative Associations

1. INTRODUCTION

A database stores the data related to a set of transactions that happen over some time. A database mined at a specific point in time. The knowledge mined at that time is valid, considering the data stored in the database at that time. When data gets added to the database, the patterns are no more valid as the size of the data is changed. New items might get added, and some items withdrawn by the seller for selling. The very nature of the data might get changed, and as a result, the patterns that mined at the earlier instance may not be valid as on date. Processing the entire data again is a time-consuming and costly proposition.

A method has been presented [36] using which mining of fixed databases for identifying the negative patterns presented. Another method presented [35] mining of distributed fixed databases has presented for finding negative patterns. In both cases the database is fixed static, meaning the nature of the database has been fixed constant. But in general, no database is fixed. It keeps growing and gets updated as time progresses. The patterns that are valid for some time may not be valid after sometimes due to the changes occurring in the database.

A database can be accumulated using data increment, which is valid at a point in time.

An item set that happens within several transactions is called patterns. The patterns as such will be huge in number processing of which will take lots of time. The number of patterns also keeps increasing as the database increases. The frequent patterns are more important — frequent patterns found by selecting the patterns that meet the minimum threshold value dictated by the user. The support value is an interesting measure selected by the user. The frequency of a pattern is the number of times that pattern appears within the database. There is no period fixed in this case. Frequent patterns may be sporadic meaning there may not be any regularity that defines the occurrence behavior of patterns.

Thus there is a problem that frequent patterns may not be regular, and regular patterns may not be frequent. Sometimes, there is a necessity to consider both of them. The association between the patterns mined is equally important. Many times, a positive association among the patterns considered concerning either regular, regular-frequent, regular-frequent-maximal.

However, some of the patterns may have a negative association which means one pattern contradicts the other which is the case when one deals with either drug having a different chemical composition or weather forecasting where one whether pattern contradicts another leading to wrong conclusions and decisions. Negative associations are some times more important than the positive associations and therefore need clear investigations of the same. In the paper, the issue of finding the negative associations considering a fixed database and incremental database situated at a specific location considered. A method that considers the data as a sliding window presented in this paper.

2. PROBLEM DEFINITION

Thus the problem is to find regular, frequent, maximal item sets that are negatively associated when the transactional

data is available as a fixed database and also as an increment located at the same sites on the same server.

3. RELATED WORK

Rakesh Agarwal et al., [1] described mining association rules between sets of items in large databases. They developed an algorithm on a transaction database of traditional market basket analysis. This algorithm uses the pruning concept within the Apriori algorithm. To process, the queries user-specified functions required. They have enhanced their work. With the help of this system, one could answer the queries with consequent and antecedent. For this, they ignored the confidence factor of the earlier support confidence framework. This kind of finding is also useful in shelf management in supermarkets easily. If the total set of rules that are generated for an itemset x in Y with k item, then all the subsets of 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 the support of x and should satisfy the given threshold.

Agarwal, R et al., [2] presented two algorithms, namely Apriori and AprioriTiD, for discovering association rules that exist among the item sets in a large database. These algorithms differ in the way candidate itemsets are counted and generated. They have presented the features of an algorithm achieved by combining the two algorithms that they have presented. They have shown that the hybrid algorithm scales–up linearly concerning the number of transactions contained in the database.

Jiawei Han et al., [3] in their earlier research stated about candidate set that it is expensive to generate candidate sets when the patterns are prolific in other words losing patterns. Then they proposed FP tree refers to a frequent pattern tree, which is an extension of a prefix tree. This kind of tree follows the prefix tree approach to accommodate crucial compressed information about frequent patterns. The tree is useful in finding complete frequent patterns by fragmenting them. Based on the divide and conquer technique, they developed a strategy to split the big problem into small divisions. The tree structure avoids candidate generation at each level by fragmenting the database. 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 kind of a concept described in other words as confined search in the conditional database. This tree-based approach certainly works efficiently than the precious expensive algorithms and remained a landmark in the frequent patterns as well as regular itemset mining

Zaki at al [2003] introduced the vertical format of the database, which is the transpose orientation of the traditional transaction databases. In the transaction database, the tuples are of the form transaction number and itemset as two attributes whereas in the vertical format the table is of two columns, itemset, and the corresponding transactions,

One needs to consider both positive and negative associations among the exiting patterns to get a meaningful picture of the regularly occurring patterns. An extension to the existing traditional association rule that considers the extension of rules such as $A \Rightarrow \neg B$, $\neg A \Rightarrow B$, and $\neg A \Rightarrow \neg B$ *by* W. Xindong et al., [5]. The method proposed by them considers an appropriate pruning strategy and a suitable measure for computing the interestingness of the patterns.

Data gets generated periodically. Predicting the way the data generated some times can be achieved through varying periodic mining. Fixing periodicity for a time series is challenging. Most of the algorithms take the periodicity as input from the users which is a clear limitation as the periodic provided by the user some may either go wrong or just not suitable to predict the trends. A novel method has proposed by M.G et al., [6] for detecting the periodicity rate on a specific time series.

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. The number of frequent itemsets is much higher than the number of equivalence classes.

Many algorithms from a different angle presented in the literature that can discover positive and negative association rules, These algorithms suffer from many angles that include the need for heavy memory, too many CPU cycles, etc. C. Cornell et al., [8] have critically examined all the research articles and cataloged the algorithms based on the criteria used by those algorithms. An Apriori-based algorithm presented for mining both positive and negative associations and which uses a support confidence framework.

Lin Zhou et al., [9] weblogs contain gigabytes of data every day about the dynamics of the web. These webpages accessed in sequences. Weblogs are mined to discover these patterns and named as path traversal patterns. Different users have different habits of accessing webpages. Hence we need a measure to help the webserver for decision making. Depending on how interesting and helpful the webpage, the measured utility is defined. The two-phase utility mining method is best suitable to find high utility path traversal patterns. The result proved that these paths are well ahead of traditional frequent pattern mining in helping decision making easy to users. Weblog contains the information about the IP address, timestamp 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 considered in this algorithm. Usefulness is the measure of utility. In other words, the measures like how much time a user spent on a webpage can be a measure of interestingness.

Yue-shi Lee et al., [10] proposed TWU (Transaction Weighted Utilization) to maintain Apriori downward closure property. Each level of candidate set generation of TWU, previous level elements used. The idea of interactive mining is about modifying the min support required as and when weblogs 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 weblogs. Based on timestamps, some of the records 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 the track down the activities of users. The knowledge generated with these algorithms is helpful for the developers to make necessary changes for various actions and options on a website.

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

Earlier researchers developed few algorithms namely Apriori All, GSP by Agarwal and Srikanth. Later, Wang developed the suffix tree algorithm. Zaki in 2000 developed an SPADE, algorithm. Pie et al. also developed Free Span and Prefix Span. They also developed a WAP-tree algorithm to work on a single element set sequence. This database found in e-commerce websites where users click on a single item at any point in time. Also, there will be forward, and backward movement by the user to know different models and other details of models and add them to the cart. Along with the PLWAP-Tree approach, there are several other algorithms such as FS-miner, pattern growth tree also contributed have their glory in the area of finding association rules of the form $x=>y$ and $x \cap y=\{ø\}$.

Lie Chang et al., [11] proposed CSTrea construction 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 rescan 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 which lead to the improvement of efficiency of the algorithm. The pre-FUFP algorithm by Chu-Wei is to find frequent itemset. The former methods require identifying candidate sets in the former methods. Later FPtree is an efficient tree data structure to find the patterns successfully. For both approaches, the transaction database processed batch-wise. Whenever new transactions included, it is necessary to update the tree. For this purpose, a fast updated FP tree structure invented. In their research, they developed the Pre-FUFP algorithm. They maintain a pair of upper support threshold as well as the lower support threshold which does not require rescanning the database for newly occurred transactions. Hence this is one of the landmark algorithms in finding frequent patterns in incremental databases, which is far better than COFP-Tree (conditional FP-tree), QFP-growth and generalized FP-tree, and so on.

Jigyasa Bisaria et al., [2009] [12] in their research article stated that in incremental databases, any infrequent pattern with a timestamp could become frequent after a delta time. With $t+\Delta t$ time stamp which requires a separate methodology to handle all these infrequent patterns. This approach maintains a limited number of patterns in the knowledge base. This incremental sequence extraction ISE method works based on rough sets. According to rough sets, the partition should satisfy certain conditions such as $yi \neq ø$, $Yi \cap yj=ø$, $V= 4$. Then the equivalence class forms a partition. 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 [2009] [13] described utility is one of the criteria of the interestingness of a sequence. As the utility is high, the sequence is more useful. One can observe that utility value is positive always. Hence finding high utility sequences associated with negative utility is also an important aspect in data mining. HUINIU-mine is introduced to find such sequences which will identify transaction weighted items with high utility values, which are few.

Apriori algorithm is one of the most classical algorithms invented for mining association rules. The transactional database scanned many times for mining the candidate sets, which are representatives of the Itemset patterns. Yi-ming et al., [14] have presented an algorithm that requires the scanning of the database and in the process constructs vertical table format, which is used to extract the association rules that exist among the mined itemsets. The algorithm requires less storage for computing the algorithm.

S K Tanveer et al., [2010] [15] proposed a tree-structured algorithm for regular patterns. Frequent patterns are a fundamental approach in data mining concepts. They are discovered based on the support count and correlation framework in transactional databases. These patterns are no longer satisfied with the requirement of finding meaningful patterns. Periodic patterns are a well-attempted technique to find patterns in temporal databases. These methods

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 used as the qualifying measure for any pattern to be regular. The maximum of the periods must be less than the regularity threshold.

Farhan Ahmed et al. [2009] [16] in their research, stated that the Real-world scenarios would not reflect with the binary occurrences approach of webpages. Hence consider the time spent on the webpage by the user as a utility measure. Apriori algorithm generates too many candidate sets and leads to several database scans to find the web traversal path. A novel algorithm EUWPTM proposed 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. WEB MINER is a web mining system useful in applying the data mining techniques on the World Wide Web (www). Mining language MINT is used in WUM (Web Utilization Miner) to investigate dynamically specified interesting web patterns. The algorithm is a pruning based algorithm to prune unwanted 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 the database. There are several other 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 a pruning strategy.

Many of the databases get developed increment by increment. The patterns existing in the database keep changing as and more increments added to the database. It is necessary to mine the database yet again every time an increment added to the database. Efforts are made to mine regular patterns from incremental databases by Vijay Kumar et al., [17] especially using the vertical formats considering the user-defined regularity threshold.

Eya Ben Ahmed et al., [2011] [18] focussed on the structure of the data warehouse. According to the mining is performed on the data from the data warehouse. While presenting the data from the transactional database to the 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.

Most of the research is focussed on frequent mining itemsets considering a threshold value proposed by the users. Rehualirty of the occurrence of a pattern is equally important in addition to the frequency. The regularity of a pattern exhibits the behavior of the occurrence of the pattern. The regularity and the frequency change every time an increment added to the database requiring the mining of the databases every time an increment to the database added. Vijay Kumar et al., [19] have proposed to mine frequent and regular patterns using vertical format with a view of generating positive associations**.**

Diana Martin et al., [2014] [20] developed a MOPAR algorithm. This algorithm is multi-objective and works on large databases, evolves a reduced set of positive and negative quantitative association rules. According to the 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 strict rules only extracted during execution. Comprehensibility is the property that an association rule should be easy to understand. Otherwise, the user will unlikely to use a complex rule which is not understood

Closed Patterns, compressed patterns, and so on comes under frequent compact 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 many times. Diana Martin et al. have presented an algorithm to find the negatively associated positive regular patterns. These patterns are also known as non-overlapping patterns or contradicting patterns. A vertical format of the database used with a sliding window with different sizes and with different thresholds

Incremental databases are the repositories of most emerging realistic data from e-commerce sites and other sources. They are typical as new transactions 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 a very complex process because of the search space and the size of the database. These no overlapping patterns play a vital role in decision making by extracting complex hidden knowledge from the transactional databases. Window Sliding progresses with time, leaving the old transactions from one end and keep on including new transactions from another end. The vertical format of the 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. The database need not be scanned several times in this approach

Many contributions presented in literature which aim at mining frequent items sets having positive associations [21] [22][23][[24][25][26][27][28][29][30][31][32][33]. Pawan et al. presented the way regular and frequent patterns that yield negative associations mined from the static database, data Streams, and distributed databases[34][35][36]. They have presented in this paper the way the mining of regular and frequent patterns mined when dealing with incremental databases.

4. COMPARATIVE ANALYSIS

Algorithms used for negative mining associations considering regular, frequent itemsets to assess the adequacy of those algorithms and also considering incremental databases presented. **Table 1** shows the comparison. From the table, one can see that none of the existing algorithms are dealing with the most important aspects of the negative associations that include regularity, positive/negative associations, and frequency and interestingness measures considering the incremental databases.

5. INVESTIGATIONS AND FINDINGS

5.1 Architectural design of Experimental Incremental database

IBM supplied 100,000 records related to sales transactions. Out of these 90,000 (Ninety Thousand) records placed into a flat-file and 10,000 (ten thousand) records placed into another flat file. The original database and the increment of the same stored on the same server. It means that a static database of 90,000 records and an incremental database of 10,000 records considered. Here database increment is considered as 10,000 records. The processing of the transactional database that is updated once in a while with an Increment of 10,000 records is undertaken using an IBM database. However, to evaluate the algorithm proposed a sample database of 15 records and an incremental database of 6 records considered.

An Algorithm developed and implemented, which can process the negatively associated patterns considering both the data files. The architecture of the incremental database processing implemented for determining the overall patterns shown in Figure 1.

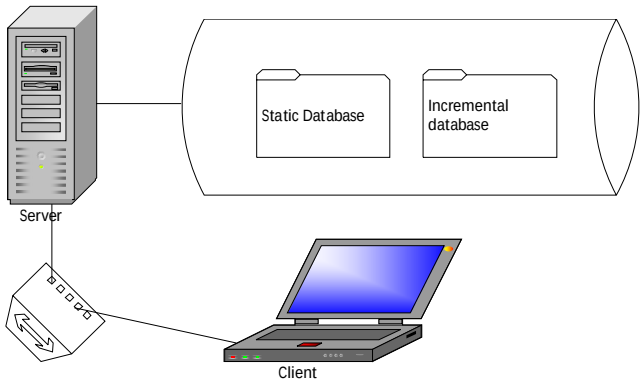

Figure 1: Incremental database and Mining Systems

5.2 Creating a Static and Incremental Database

Twenty-one records extracted from IBM supplied data. Records 1-15 considered static databases, and the records 16-21 have been added as an increment to the existing database, thus forming 21 records. Two windows extracted out of 21 records. Records from 1-15 are considered as window-1 and record 6-21 as windows two and then the pattern recognition and itemset considering both the windows carried. The original incremental data and the two windows created shown in Table 2.

5.3 Algorithmic Approaches for finding negative associations from incremental databases considering the regularity and frequency of the item sets

In a sample of data contained in IBM supplied data, two sets of records extracted that constitute records 1-15 and 6-21. An increment of 6 records added to the original. The records from 1-15 regarded as window-1 and the records from 6-21 considered as window-2.

Algorithm

1. Consider the static data and incremental data that stored in two flat files
2. Read the beginning and ending pointer for the sliding windows
3. Read the transactions contained within starting and ending pointers into an Array as shown in Table 3
4. Convert the data in table 3 into the vertical format as shown in Table 4
5. Prune the Initial Irregular and non-frequent Items
6. Repeat the following process

   Consider the current Item

   Select the next item and prune it if it is not regular or not frequent and go to the next item (self-Loop).

If the next item is regular and frequent, get the intersection of the transactions of the current item and next item
If the intersection is null, then enter the current and next items into a negative set array
If the intersection is not null, then get the common elements and see if the count of elements is > frequency and the displacement between the pattern < regularity threshold decided by the user
If the common elements satisfy the regularity constraint and frequency constraint, then add the common elements into a vertical table as a new row as they are regular and frequent.
If the common elements do not satisfy the regularity and frequency constraint, then ignore them
If all the elements in the vertical table are exhausted, then convert the next item next to the current item as the current item and then LOOP
If all the elements in the vertical table are not exhausted, then move to the next item and LOOP

5.4 Experimentation for Mining Negative patterns based on regularity and frequency considering Incremental Databases

Step-1

Consider the first 15 records of sample data (window-1) and add Transaction IDS. The details of records contained in window-1 shown in Table 3, which shows the list of transactions and the items contained in those transactions. These transactions stored as a Flat file at the server

Step-2

Convert the records in window-1 to Vertical format. Table 4 shows the vertical format data. In the vertical data format, for each of the Item, in the data repository, the transactions that contain the Items are found and mapped.

Step-3

Find the first regular item by pruning all the previous items whose regularity is > User given Maximum Regularity threshold (λ_{min_reg}). Here regularity implies the relative occurrence of the Item, computed as the distance between two successive transactions. Considering the (λ_{min_reg}) = 5. The First regular Item is called Previous-Item.

Step-4

Consider each item starting from Previous-item and repeat the following procedure.

1. Consider the next item and let that be current-item
2. Find if the current-item is regular. If the current-item is not regular prune it.
3. If the current- item is regular, find Intersection of the transactions of the current item with the previous-item.
4. If the intersection is null, then add the Item set into the negative item-set list.
5. If the intersection is not null, find the regularity considering the common elements.
6. If the regularity is < (λ_{min_reg}), then add the previous item and the current item set along with its related transaction as an additional record to the vertical database since they are positively associated.
7. If the next item is not the lost entry in the vertical table, Make the Current Item as the next Item and loop.
8. If the next item is the last in the Vertical table, then Previous Item = Previous Item +1 and then Loop.

After this step, the positively associated regular and frequent itemsets and the negatively associated regular and frequent itemsets are shown in Table 5 and Table 6 respectively

The records related to window-2 (records 6-21) and the vertical format of the same, shown in Table 7 and Table 8. The algorithm applied to the records contained in window-2 and the items sets that are positive, regular and frequent shown in Table 9 and similarly, the itemsets that are negative, regular and frequent itemsets shown in Table 10.

A comparison of negative associations between regular, frequent items sets derived out of records contained in window-1 and window-2 is shown in Table 11. From the table, it can be seen that item sets that yield negative association greatly differs from window to window as the additional increments get added into the system

Pseudo Code

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 λ VDB)  return;
  if (TrnIdIy∩ TrrnIdIj) = = {;} ) Result=Result ∪I_yj ;
  else
  return ;
}

Algorithm INC_NPRISM()
{
   // To find the 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);
         }
    }
}
```

5.5 Data Analysis of IBM supplied data

IBM supplied 100,000 sales related transactions out of which 25,000 records selected. Two windows comprising records 1 to 20, 000 as window-1 and 5000 to 25000 records as window-2 have created. The algorithm relating to mining negative regular and frequent itemsets have been generated considering both the windows. Data Analysis of window-1 is done considering the Maximum regularity being 800 and varying Maximum frequency commencing from 400 to 750, generating the number of positive and negatively associated item sets. The details of such a generation shown in Table 12

Similarly, Data Analysis of window-2 is done considering the Maximum regularity being 800 and varying Maximum frequency commencing from 400 to 750, generating the number of positive and negatively associated item sets. The details of such a generation shown in Table 13

The positive and negative associations considering regularity and frequency of the itemsets that are related to window-1 and window-2 shown in Figure 2 and Figure 3. There are slight variations in the number of positively and negatively associated item sets even with a small addition of increment of 5000 Items.

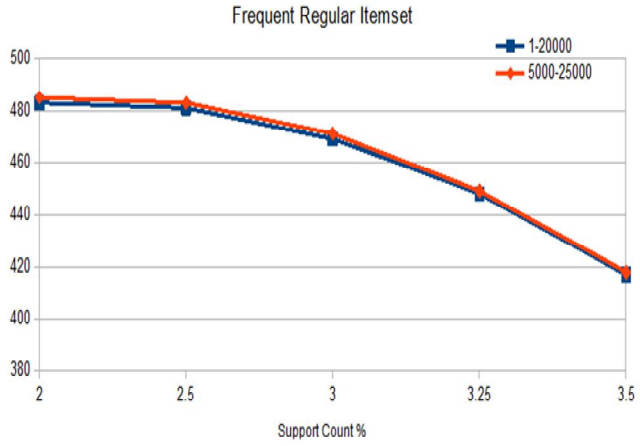

Figure 2: Frequent regular items sets for WINDOW-1 and WINDOW-2

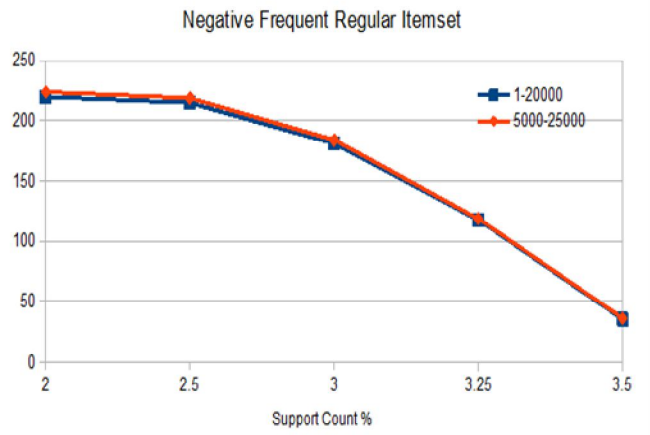

Figure 3: Frequent regular items sets for WINDOW-1 and WINDOW-2

6. CONCLUSION

Data is never static. The patterns that are generated using the static data will be no more valid when new data gets generated. The patterns are either re-adjusted are generated when new data get added to the database. The patterns generated must be up-to-date so that current day decisions effected. The method proposed in this chapter uses a single-window concept which allows for the definition of pointer using which a window is selected — the transactions contained in a window used for the generation of patterns.

Frequent patterns generated without much concern with the time during which the item set occurs. Frequency is just a count. The regularity of occurrence of a pattern is more important, which means the patterns that happen within a period are important. Item sets found concerning a fixed time frame and frequency. All the three dimensions of pattern finding ({regularity}, {regularity, Frequency}, {regularity, Frequency, Maximally}) investigated so that the patterns found from different dimensions can be investigated and used as per their applicability

REFERENCES

1. Agarwal R, Imielinski T Swamy, Mining Association Rules between Sets of Items in Large Databases, SIGMOD Conference on Management of Data, pp: 207-216, 1993
2. 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.
3. 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
4. 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.
5. 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
6. 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).
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. 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.
9. 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.
10. Yue - ShiLee Show-JaneYen "Incremental and interactive mining of web traversal patterns" Information Sciences Volume 178, Issue 2, 15 January 2008, Pages 287-306
11. 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.
12. 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
13. Chun-Jung, Chua Vincent, 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
14. Yi-ming, G., Zhi-jun, W. A Vertical format algorithm for frequent mining itemsets, IEEE Transactions, pp. 11-13, 2010
15. Tanveer, S.K., Ahmed, C.F., Jeong, B.S. Lee, and 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.
16. Chowdhury Farhan Ahmed, Syed Khairuzzaman Tanveer, Byeong-Soo Jeong, Young-Koo Lee, Efficient mining of Utility-Based web path Traversal Patterns, ISBN 978-89-5519-139-4, ICACT 2009.
17. 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

18. Eya Ben Ahmed, Ahlem Nabi 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.
19. 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.
20. Diana Martín, Alejandro Rosete, Jess Alcalá-Fdez, Francisco Herrera, "A New Multi-objective 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
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. N V S Pavan Kumar, K Rajasekhara Rao "Mining Positive and Negative Regular Item-Sets using Vertical Databases" IJS-SST.a.17.32.33, pp. 232-248, 2016
23. Greeshma, L., Pradeepini, G., Mining Maximal Efficient Closed Itemsets Without Any Redundancy, Advances in Intelligent Systems and Computing, 433, pp. 339-347
24. Kyeongjoo Kim, Jihyun Song, and Minsoo Lee, Real-time Streaming Data Analysis using Spark, International Journal of Emerging Trends in Engineering Research, Volume 6, No.1, 2018, pp. 1-5
25. Changala, R., Rajeswara Rao, D., Evaluation and analysis of discovered patterns using pattern classification methods in text mining, ARPN Journal of Engineering and Applied Sciences, 13(11), pp. 3706-3717,2018
26. Kolli, S., Sreedevi, M., Prototype analysis of different data mining classification and clustering approaches, ARPN Journal of Engineering and Applied Sciences,13(9), pp. 3129-3135,2018
27. Deshpande, L., Rao, M. N., Concept drift identification using classifier ensemble approach, International Journal of Electrical and Computer Engineering,8(1), pp. 19-25,2018
28. Changala, R., Rajeswara Rao, D. T., A survey on the development of pattern evolving model for discovery of patterns in text mining using data mining techniques, Journal of Theoretical and Applied Information Technology,95(16), pp. 3974-3981,2017
29. Wagner, S.S., Rajarajeswari, P, Parallel frequent dataset mining and feature subset selection for high dimensional data on Hadoop using map-reduce, International Journal of Applied Engineering Research, 12(18), pp. 7783-7789,2017
30. Vijay Kumar, G., Krishna Chaitanya, T., Pratap, M., Mining popular patterns from the multidimensional database, Indian Journal of Science and Technology, 9(17),93106
31. Gangadhar, M. N. S., Sreedevi, M., Regular pattern mining on dynamic databases using vertical format on given user regularity threshold, Journal of Theoretical and Applied Information Technology, 86(3), pp. 360-364, 2016
32. Greeshma, L., Pradeepini, G., Input split frequent pattern tree using MapReduce paradigm in Hadoop, Journal of Theoretical and Applied Information Technology, 84(2), pp. 260-271
33. Dr. J. Sasi Bhanu, Dr. JKR Sastry, B. Sunitha Devi3, Dr. V Chandra Prakash, Career Guidance through TIC-TAC-TOE Game, International Journal of Emerging Trends in Engineering Research, Volume 7, No.6, 2019, pp. 25-31
34. NVS Pavan Kumar1 Dr.JKR Sastry, Dr. K Raja Sekhara Rao, Mining Distributed Databases for Negative Associations from Regular and Frequent Patterns, International Journal of Advanced Trends in Computer Science and Engineering, Volume 8, No.4, July – August 2019
35. NVS Pavan Kumar1 Dr.JKR Sastry, Dr. K Raja Sekhara Rao, Mining Negative Frequent regular Itemsets from Data Streams, International Journal of Emerging Trends in Engineering Research, Volume 7, No.8 August 2019
36. NVS Pavan Kumar1 Dr.JKR Sastry, Dr. K Raja Sekhara Rao, Mining Negative Associations between Regular and Frequent Patterns hidden in Static Databases, International Journal of Emerging Trends in Engineering Research, Volume 7, No.7 July 2019

Table 1: Comparative Analysis of Algorithms – Negative Associations – Regular and Frequent - using Incremental Databases

Algorithm Serial Number	Main Author	Interestingness measures					Occurrence Behaviour					Type of Associations		Mining technique
		Support	Confidence	Correlation	Multi support	Multi Correlation	Regularity	Irregularity/ Rare	Frequent	Maximal	Natural	Positive Associations	Negative Associations	
1	Agarwal R -1										√			
2	Agarwal R - 2										√			
3	Jiawei Han								√			√		FP Tree
4	Zaki	√							√			√		Vertical format
5	W. Xindong	√							√			√	√	
6	Elfeky, M.G						√					√		
7	Xie Zhi-Jun	√							√			√	√	FIET Tree
8	Lin Zhou										√			Two phased utility
9	Yue-shi Lee	√									√			Weighted Utilisation
10	Lie chang	√	√						√			√		Pre-FUFP
11	Jigyasa Bisaria	√							√					
12	Chun -Jung Chu	√									√			Utility based
13	Yi-ming	√									√			Vertical format
14	S K Tanbeer	√					√					√		
15	Farhan Ahmed	√									√			EUWPTM
16	Vijay Kumar-1	√					√					√		VDRP
17	Farhan Ahmed	√					√					√		
18	Eya Ben Ahmed	√					V							
19	Diana Martin	√					√					√	√	MOPONOR
20	Pavan NVS	√	√				√	√	√	√	√	√	√	Veridical Tab

Table 2: Database, database increment, and Window identification

Trid	Itemset	Type of Database	Window-1	Window-2
1	1 2 3 4 5 9 10 14	Fixed Database	Window-1	
2	4 5 6 7 10 15			
3	2 3 7 13 14 15			
4	8 10 11 15			
5	1 3 6 9 13			
6	4 5 6 15			Window-2
7	2 3 7 9 11 12 13			
8	5 8 11 12 14 15			
9	1 3 7 8 9 13			
10	4 5 6 8 9 10 15			
11	2 4 5 7 13 14			
12	5 8 11 15			
13	1 3 4 9 11			
14	4 5 6 11 13 14 15			
15	2 3 6 7 12 13			
16	5 8 11 12 14 15	Database Increment		
17	1 3 5 6 9 10			
18	4 5 6 12 14 15			
19	2 3 4 7 13			
20	5 8 11 12 15			
21	1 3 5 9 14			

Table 3: Transaction Table (window-1) sample data

Trid	Itemset
1	1 2 3 4 5 9 10 14
2	4 5 6 7 10 15
3	2 3 7 13 14 15
4	8 10 11 15
5	1 3 6 9 13
6	4 5 6 15
7	2 3 7 9 11 12 13
8	5 8 11 12 14 15
9	1 3 7 8 9 13
10	4 5 6 8 9 10 15
11	2 4 5 7 13 14
12	5 8 11 15
13	1 3 4 9 11
14	4 5 6 11 13 14 15
15	2 3 6 7 12 13

Table 4: Transaction Data (window-1) in vertical format

Itemset	**Trids**
1	5 9 13
2	3 7 11 15
3	3 5 7 9 13 15
4	2 6 10 11 13 14
5	2 6 8 10 11 12 14
6	2 5 6 10 14 15
7	2 3 7 9 11 15
8	4 8 9 10 12
9	5 7 9 10 13
10	2 4 10
11	4 7 8 12 13 14
12	7 8 15
13	3 5 7 9 11 14 15
14	3 8 11 14
15	2 3 4 6 8 10 12 14

Table 5: Positively associated regular and frequent Itemsets (window-1) with MR=4 and MF = 5

#	Itemset	Trids	Periods	Max Regularity	Support Count
1	3	3 5 7 9 13 15	2 2 2 4 2	4	6
2	4	2 6 10 11 13 14	4 4 1 2 1	4	6
3	5	2 6 8 10 11 12 14	4 2 2 1 1 2	4	7
4	6	2 5 6 10 14 15	3 1 4 4 1	4	6
5	7	2 3 7 9 11 15	1 4 2 2 4	4	6
6	8	4 8 9 10 12	4 1 1 2	4	5
7	9	5 7 9 10 13	2 2 1 3	3	5
8	11	4 7 8 12 13 14	3 1 4 1 1	4	6
9	13	3 5 7 9 11 14 15	2 2 2 2 3 1	3	7
10	15	2 3 4 6 8 10 12 14	1 1 2 2 2 2 2	2	8
11	5 15	2 6 8 10 12 14	4 2 2 2 2	4	6

Table 6: Negatively associated regular and frequent Itemsets (window-1)

#	**Item Set -1**	**Item Set-2**
1	3	5

Table 7: Itemset (window-2) with transactions Ids

Serial #	Transaction ID	Item Numbers
1	6	4 5 6 15
2	7	2 3 7 9 11 12 13
3	8	5 8 11 12 14 15
4	9	1 3 7 8 9 13
5	10	4 5 6 8 9 10 15
6	11	2 4 5 7 13 14
7	12	5 8 11 15
8	13	1 3 4 9 11
9	14	4 5 6 11 13 14 15
10	15	2 3 6 7 12 13
11	16	5 8 11 12 14 15
12	17	1 3 5 6 9 10
13	18	4 5 6 12 14 15
14	19	2 3 4 7 13
15	20	5 8 11 12 15
16	21	1 3 5 9 14

Table 8: Inverted Itemsets (window-2) with Transaction

#	**Itemset**	**Trids**
1	1	9 13 17 21
2	2	7 11 15 19
3	3	7 9 13 15 17 19 21
4	4	6 10 11 13 14 18 19
5	5	6 8 10 11 12 14 16 17 18 20 21

#	Itemset	Trids
1	1	9 13 17 21
6	6	6 10 14 15 17 18
7	7	7 9 11 15 19
8	8	8 9 10 12 16 20
9	9	7 9 10 13 17 21
10	10	10 17
11	11	7 8 12 13 14 16 20
12	12	7 8 15 16 18 20
13	13	7 9 11 14 15 19
14	14	8 11 14 16 18 21
15	15	6 8 10 12 14 16 18 20

Table 9: Positive regular, frequent items sets (window-2)

#	Itemset	Trids	Periods	Max Regularity	Support Count
1	3	7 9 13 15 17 19 21	2 4 2 2 2 2	4	7
2	4	6 10 11 13 14 18 19	4 1 2 1 4 1	4	7
3	5	6 8 10 11 12 14 16 17 18 20 21	2 2 1 1 2 2 1 1 2 1	2	11
4	6	6 10 14 15 17 18	4 4 1 2 1	4	6
5	7	7 9 11 15 19	2 2 4 4	4	5
6	8	8 9 10 12 16 20	1 1 2 4 4	4	6
7	9	7 9 10 13 17 21	2 1 3 4 4	4	6
8	11	7 8 12 13 14 16 20	1 4 1 1 2 4	4	7
9	13	7 9 11 14 15 19	2 2 3 1 4	4	6
10	14	8 11 14 16 18 21	3 3 2 2 3	3	6
11	15	6 8 10 12 14 16 18 20	2 2 2 2 2 2 2	2	8
12	5 14	8 11 14 16 18 215-21	3 3 2 2 3	3	6
13	5 15	6 8 10 12 14 16 18 20	2 2 2 2 2 2 2	2	8

Table 10: Negative regular frequent itemsets (window-2)

#	Item Set-1	Item Set-2
1	3	15
2	7	15

Table 11: Comparison of Negative itemsets considering window-1 and window-2

#	Item Set-1	Item Set-2	Item Set-1	Item Set-2
1			3	15
2			7	15
3	3	5		

Table 12: Generation of positively associated and negatively associated items sets MR = varying MF – Window-1

	MR	MF	FR	NFR	MF%
1-20000	800	400	483	220	2
	800	500	481	215	2.5
	800	600	469	181	3
	800	650	448	118	3.25
	800	700	417	35	3.5

Table 13: Generation of positively associated and negatively associated items sets MR = varying MF Window-2

	MR	MF	FR	NFR	MF%
5000-25000	800	400	485	224	2
	800	500	483	219	2.5
	800	600	471	184	3
	800	650	449	119	3.25
	800	700	418	36	3.5