International Journal of Engineering & Technology, 7 (2.7) (2018) 636-641

International Journal of Engineering & Technology

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

Research Paper



Incremental Mining of Popular Patterns from Transactional Databases

G. Vijay Kumar[1]*, M. Sreedevi[2], K. Bhargav[3], P. Mohan Krishna[4]

[1,3,4]Department of ECSE, Koneru Lakshmiah Education Foundation, Guntur, Andhra Pradesh, India
[2]Department of CSE, Koneru Lakshmiah Education Foundation, Guntur, Andhra Pradesh, India
**Email: gvijay_73@kluniversity.in*

Abstract

From the day the mining of frequent pattern problem has been introduced the researchers have extended the frequent patterns to various helpful patterns like cyclic, periodic, regular patterns in emerging databases. In this paper, we get to know about popular pattern which gives the Popularity of every items between the incremental databases. The method that used for the mining of popular patterns is known as Incrpop-growth algorithm. Incrpop-tree structure is been applied in this algorithm. In incremental databases the event recurrence and the event conduct of the example changes at whatever point a little arrangement of new exchanges are added to the database. In this way proposes another calculation called Incrpop-tree to mine mainstream designs in incremental value-based database utilizing Incrpop-tree structure. At long last analyses have been done and comes about are indicated which gives data about conservativeness, time proficient and space productive.

Keywords: *Frequent Patterns; Popularity,; Incrpop Tree; Popular Patterns; Incremental Database.*

1. Introduction

So many works have been proposed by the introduction of the research problem of frequent pattern mining. These proposed works have been ordered into two classes. In the first category it is mainly focused on algorithm efficiency [1]. Consider a case, to evade the candidate generation and the test method of the Apriori algorithm [2], a tree-structured algorithm called FP-growth [3] was proposed to gather the data of the transactional database so that frequent patterns will also be mined recursively by using this FP-tree structure.

In the second class it is mostly attentive on the extending work of frequent patterns [4] to other interesting or useful patterns which are further used in the study. The following mining patterns are based on support calculations in the given data. However, these measurements are valuable, the support-based frequent pattern are not adequate to discover numerous interesting knowledge in transactional database among patterns. Due to this problem these arose some interestingness measures and their corresponding correlated patterns, regular patterns.

Many users want to find popular patterns in real-life situations. Now let us consider an example, a person wants to find other people within large group of people in social network [5] as these people are most popular in the group. Similarly people want to get connected easily to a popular person by searching them quickly. This search technique can be done simply if the popular people are gathered and located in one popular pattern. These can be done by more usage of these websites or networks.

In incremental databases [6, 7, 8, 9] new transactions will add continuously to the transaction database. So the Popular patterns will change whenever database is updated. In order to make best decisions, users may be interested on getting the latest popular patterns from the updated database. Many tree based applications [10] have been introduced to mine popular patterns in incremental databases.

The remaining paper is prepared as follows. In unit 2 we deliberate about the associated work. In unit 3 we will outline the problem of popular patterns in transactional database. In unit 4 we define the procedure of mining popular patterns by Incrpop-tree algorithm. In unit 5 we describe how to illustrate the Incrpop-tree construction of the mining for popular patterns. In unit 6 the popular patterns of incremental database have been calculated. In unit 7 our experimental results have been shown. Finally, the paper is been concluded in unit 8.

2. Related Work

The topic in this paper is related with the methods in finding the popular patterns by using Incrpop-tree algorithm. This mainly helps in finding the different types of patterns using different methods and algorithms. In this we also know about the incremental database and mine the popular patterns from that transactional database. R. Agarwal [2] has proposed association mining rules which are helpful in finding the item sets of the patterns. This gives information about the lowest support threshold to generate association rule from the item sets.

FP-Growth [3] it is an algorithm planned to construct the FP-tree to show fillings of transactional databases. Tanbeer [11] has intro-

Copyright © 2018Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

duced the problem in mining of regular patterns in transactional databases with high compact tree structure which helps us in knowing the mining methods to find the popular patterns from transactional database using FP-tree structure. In this paper the method of mining from the incremental databases have been introduced which is the main role in this present paper. With this mining of popular patterns in incremental database have been proposed and have been implemented. In the paper [13, 14] downward closure property has been discussed which is applicable for mining Popular patterns from the database. We have to check the downward closure property for every pattern provided in the database. After checking the downward closure property then only it is easy to mine the popular patterns.
In the paper [5] Cameron gives the method of finding the strong group of friends among the friends in a social network. Now a days many of the people use social media to get linked to each other. To find out the most important persons; data mining methods have been put on to social networks. In the paper [1] another new technique namely time-fading model has been implemented to find out the recurrent patterns from streams of undefined data. In the paper [15] mining of constrained frequent patterns have been shown. With this a new method called anti-monotone constraints have been projected which helps in finding minute portions of mined frequent patterns. There are several tree patterns to get the different patterns from the databases. A suffix tree algorithm has been proposed in the paper [10] to notice symbol, order and section periodicity in time series. In the paper [14] a new techniquecalled PFRP-methods are proposed to determineregular patterns [16] in large databases. This helps us to find the regularity [17] of each pattern in the database.

3. Problem Definition

Here, we fundamentally portray issue definitions identified with well-known patterns. Let item={x1 , x2 , x3,……x4} be an arrangement of m domain item. An arrangement of n transactions in transactional database are {t1, t2, t3,…..tn} where each transaction bethe subset of every item in the TBD. |tj| is used to speak to the measure of transaction tj .Let X={x1, x2,…..xk} belongs to item be an example comprising of k items where |x|=k≤m. the anticipated database of "X" meant as DBx is an arrangement of Transactional Data Base transaction that contain X. Here maxTL(x) and sumTL(x) are utilized to speak to most extreme length and aggregate size of every transaction in DBx.

3.1. Statement I

The transaction fame Incrpop(X, tj) of an example "X" in transaction tj that counts the enrollment level of "X" in tj. We processed the enrollment degree in light of the contrast among the transaction length |tj| and pattern length |x|.
Incrpop(X, tj) = $|t_j|$ - |X|

3.2. Statement II

The lengthy transactions Popularity Incrpop(X, tmaxTL(x)) of the pattern "X" in the transaction tmaxTL(x) measure the enrollment level of "X" in tmaxTL, where tmaxTL(x) is the transaction that have the greatest length in DBx.
Incrpop(X, $tmaxTL_{(x)}$) = $(\max_{tj \in DBx} |tj|)$ - |X|

3.3. Statement III

The Popularity Incrpop(X) of an example "X" in the Transactional database amount of an accumulated participation level of X in every transactions in the Transactional data base. It is characterized by normal of all the transactiona lPopularities of "X".
Incrpop(X) =1/$|DB_x|$ $\sum_{tj \in DBx}$ Incrpop(X, tj)

3.4. Statement IV

A client determined least popular threshold min_popp is given, an example X is viewed as prominent if the Popularity is as a minimum min_popp (i.eIncrpop(X) ≥ min_popp).

3.5. Statement V

(Popularity of Popular patterns X in incremental database)
Let us consider that new transactions have been added to the current database.
Here UBD is referred as updated data base. Let I be the num of truncations further added to the database. Initially the first transaction i.e. null transaction (tf) will be zero then whenever I transactions are added then the initial transaction will be tf+i at the beginning of the database and last transaction (tl) will be tl+i. In this way the transactions have been added and the newly defined database will be considered and the Incrpop tree algorithm will be implemented and popular patterns have been mined.

4. Methodology

Incrpop-Growth: Popular pattern mining with a Incrpop-tree

The process toward mining the regular example, the estimation of recurrence fulfills the descending closure property i.e on the off chance that one item is taken by means of rare formerly the superset of it is additionally takenrarely which helps in diminishing the hunt space by erasing the rare examples and in this manner the way toward mining is quicker. Now and again perceptive notice that the Popularity does not fulfill descending closure property. Consider an example; if {c} is Unpopular but {b, c} can be Popular. Thus the readers take the mining of popular patterns as a difficult step.

Statement VI

The Incrpop(X) of a pattern "X" in the Transactional Database counts the total participation level of X in the TDB. It is characterized as far as sumTL(x) =∑_ (tj∈DBx) 〖|tj|〗 follows.

$$\text{Incrpop(X)} = \frac{1}{|DBx|} \sum_{tj \in DBx} \text{Pop(X, tj)}$$

$$= \frac{1}{|DBx|} \sum_{tj \in DBx} (|\text{tj}| - |X|)$$

$$= \frac{sumTL(x)}{|DBx|} - |X|$$

Pop tree-algorithm
Input: DB, λ, K, N
Output: Popular patterns.
Procedure:
Let X_i subset (I) be a k-item set
For (i=1; i<=K, i++)
For each X_i
Update **IncrPop (Xi)**
Incrpop (Xi) = (Sum (Xi)/|DBxi|)-|Xi|
If Incrpop (X_i)>=λ
Popular
Else
$Incrpop^{Ub}(Xi^1)$ =max (xi)-|Xi|
For (i^1=i+1;i^1<=K;i^1++)

If ($Incrpop^{Ub}(X_i^l)$>=λ)
X_i is popular
Else
Delete X_i

Repeat
Find if any db+ exist
If dB+ exist

Repeat the procedure recursively

Else

Increase the *k* value using 'and operation' until no candidate is generated.

Let us consider an example for the above algorithm

Let us consider a table

Table 1: Transactions

Transaction ID	Transaction
t1	{b,c,d}
t2	{a,f,g,h}
t3	{b,d,e,f,g,h}
t4	{a,c,i,e}
t5	{b,e,g}
t6	{d,f,g,h}
t7	{a,c,i,f}

Consider the above table sumTL ({b, d}) =9, then Popular of pattern {b, d} is

$$\frac{\text{sumTL}([b,d])}{|\{t1,t3\}|} - |\{b,d\}|$$

$$= \frac{9}{2} - 2 = 2.5$$

Similarly Popularity of pattern {c} is $\frac{sumTL(\{c\})}{|\{t1,t4,t7\}|}$ - |{c}|

$$= \frac{11}{3} - 1 = 2.6$$

Here sumTL ({b, c}) <sumTL ({c}), so with this we can say that sumTL(X) confirms that it satisfies downward closure property.

5. Construction of Incrpop Tree

Mining the Popular patterns is proposed by Incrpop-tree algorithm that contains two techniques.

1 Incrpop tree construction

2 Mining of Popular patterns from Incrpop-tree

The Incrpop tree generally known as famous example tree has been worked by taking the required data from the transactional database by separating it for twice. Since Incrpop(X) cannot fulfill the downward closure property; the disagreeable thing should be kept in Incrpop tree for quite a while as their supersets might be well known. Yet, sumTL(x) fulfills the downward closure property so not at all compelling reason to keep the disliked examples in the tree.

Let us see the doublesituations:

Condition (i): the Popularity will be lower or identical to its lengthy transaction's Popularity i.e. Incrpop(x) ≤ Incrpop(X, t_{maxTL})

Condition (ii): For X ≤ X|,Incrpop (X|) will not surpass maxTL(x) - |X|

From the above equation is another condition which gives upper bound Popularity Incrpop(X')

$Incrpop^{UB}$(X') = maxTL(X) - |X'|

By the conditions mentioned above we eliminate the unpopular patterns that is known as super-pattern Popularity check. The Incrpop-tree also consists of a parent node and a child node. In constructing the Incrpop tree structures the foremost steps should be computed for every patterns are below

<x: support(x) ,maxTL(x) ,Incrpop(x) >.

The min_popp is set by the user. Let us take an assumed min_popp = 3.2

Now compute every pattern in the table above in order to find the pattern that is not popular

<a: 3,4,3> , <b: 3,6,3> , <c: 3,4,2.6> , <d: 3,6,3.3> , <e: 3,6,3.3> , <f:4,6,3.5>, <g: 4,6,3.2> , <h: 3,6,3.6> , <i:2,4,3>

By computing every pattern's we came to know the pattern's a,b,c, i, etc. not popular pattern hence their Incrpop(x) value is lower compared to min_popp. But we can't eliminate them without calculating Incrpop (X').

Computing the Incrpop (X') values of the transactions that has lower value than min Incrpop

Incrpop (X') =maxTL(X) - |X'| (|X'|=|X|+1)

Incrpop(a')= 4-2=2 ; Incrpop(b')= 6-2=4 ; Incrpop(c')= 4-2=2 ; Incrpop(i')=4-2=2

By computing this we came to realize that b has Incrpop esteem which is more prominent than min_popp. So at long last all are mainstream designs separated from a, c, i. In the wake of computing the Incrpop(X') a, c and I are demonstrated as unpopular so they are not associated with Incrpop tree structure. The rest of the examples are submitted in plunging request besides the H-table will be set as <x: support(x), sumTL(x), maxTL(x)> and the Incrpop tree will be drawn by substituting every transaction one by one.

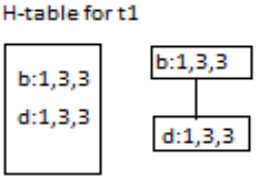


Fig. 1:H-table for t1

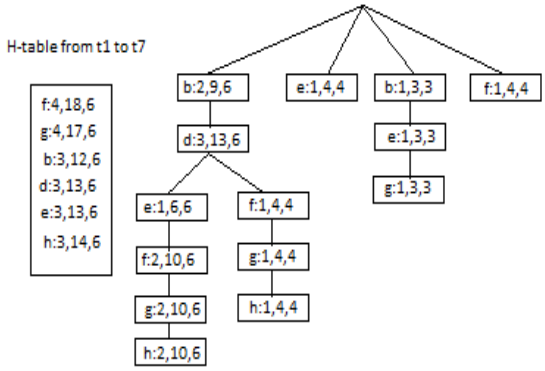


Fig. 2:H-table from t1 to t7

After drawing the Incrpop tree structure, a projected data base is been considered and then each pattern is compared with that projected data base and Popularity of each pattern has been calculated. The values of each pattern calculated with respect to the projected database which are less than the min_popp provided by the user those patterns will be pruned from the transactional data base and the tree is drawn for the second time with the help of the remaining popular patterns.

Let us consider {h} as a projected database. Now this {h} is compared to every popular pattern which has been considered and a tree is again drawn.

Incrpop values of the patterns related to projected database has been calculated

Incrpop (f, h) = 14/3 – 2 = 2.6

Incrpop (g, h) =14/3 – 2=2.6

Incrpop (b, h) =6/1 - 2= 4

Incrpop (d, h) =10/2 – 2=3

Incrpop (e, h) = 6/1 – 2 =4

By calculating the Incrpop values we compare this with the min_popp and remove the patterns which has less Incrpop value .the patterns which are removed are **f, g, d.** Incrpop tree has been drawn

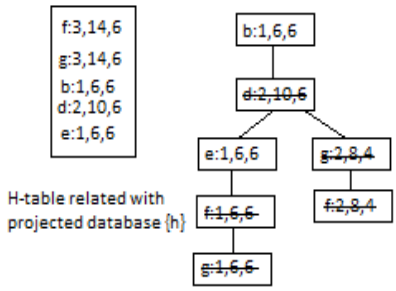


Fig. 3:H-table related with projected database{h}

So finally **b, e** are the Popular patterns.
In this way we can easily find out the popular patterns in the large transactional database and which also makes the user easy to discover the popular items sets in the entire data set given.

6. Incrementing the Data

The process is same even though so many transactions are been added. But the main thing is this is there is no need to consider the entire table i.e. from first transaction to the last transaction. First check how many transactions has been added. According to it the transactions has been considered. Suppose three transactions have been added at the end then we consider transactions from t3 but not t1.
For example, 3 transactions have been added

Table 2: New added transactions

Transaction id	Transaction
t8	{d,e,f,i}
t9	{a,c,e,h,i,b}
t10	{b,g,h}

So the table will be

Transaction id	Transaction
t1	{b,c,d}
t2	{a,f,g,h}
t3	{b,d,e,f,g,h}
t4	{a,c,i,e}
t5	{b,e,g}
t6	{d,f,g,h}
t7	{a,c,i,f}
t8	{d,e,f,i}
t9	{a,c,e,h,i,b}
t10	{b,g,h}

Now totally there are 10 transactions. But there is no need to calculate the entire 10 transactions. We need to consider the transactions from t4 to t10 and calculate their Popularity. After calculating the Popularity of all the items in the transactions then the Incrpop tree structure is been drawn and popular patterns has been calculated.
Now 3 transactions have been newly added then the initial transaction value will be tf+i (0+3=t3) and the final transaction value will be tl+i (7+3=t10). Now the newly considered database consists of transactions from t4 to t10.
Now the considered table is

Table 3: Transactions from t4 to t10

Transaction id	Transaction
t4	{a,c,i,e}
t5	{b,e,g}
t6	{d,f,g,h}
t7	{a,c,i,f}
t8	{d,e,f,i}
t9	{a,c,e,h,i,b}
t10	{b,g,h}

Now Incrpop values have been calculated for each pattern by using the formula mentioned above
Incrpop (a) =14/3 -1=3.6,
Incrpop (b) = 12/3 – 1 =3,
Incrpop(c) = 14/3 – 1 =3.6,
Incrpop (d) =8/2 – 1=3,
Incrpop (e) = 17/4 – 1=3.2,
Incrpop (f) = 12/3 – 1=3
Incrpop (g) = 17/4 -1 =2.3,
Incrpop (h) =13/3 -1=3.3,
Incrpop (i) =18/4 – 1 =3.5
Here also consider min_popp = 3.2
So the patterns which has less than min_popp value have been eliminated. So the patterns which has less Incrpop value are b, d, f, and g.
We should not eliminate them without calculating the Incrpop (X') value. Incrpop(b')= 6-2 = 4, Incrpop(d')=4-2 = 2, Incrpop(f')= 4-2=2, Incrpop(g')= 4-2 =2. By calculating this we can observe that b value is more than the min_popp so b is also a popular pattern. Remaining **d, f, g** are not Popular.
Now with the Popular patterns the format has been calculated which has to be inserted in H-table
<a:3,14,6> , <b:3,12,6> , <c:3,14,6>, <e:4,17,6> , <h:3,13,6> , <i: 4,18,6>
Now these are considered in H-table in descending order of their support value. H-tables have been generated and tree has been drawn. Now consider a projected database here {h} is considered as projected database. With the help of this projected database the Incrpop value of each item is calculated and finally popular patterns have been found.

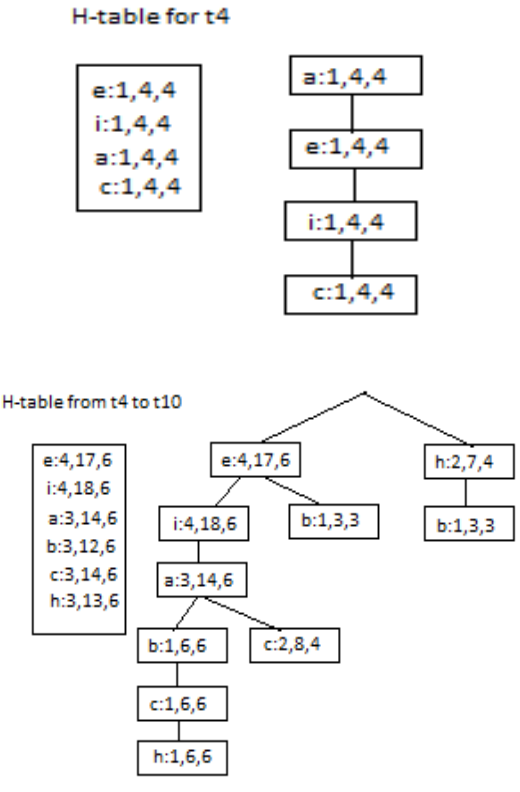


Fig. 4:H-table from t4 to t10

Now Incrpop values of each patterns related to projected database is calculated
Incrpop(e,h) = 6/1 – 2=4 , Incrpop(i,h) = 6/1 – 2=4 , Incrpop(a,h)= 6/1 – 2=4
Incrpop (b, h) =9/2 – 2=2.5, Incrpop(c, h) =6/1 – 2 = 4
Now by computing this we know that b is not popular since its value is less than the min_popp value. So this is eliminated
Now a tree has been drawn.

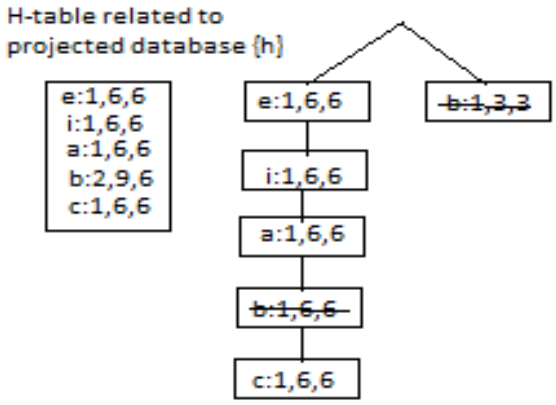


Fig. 5:H-table related to projected database{H}

Now finally **e, i, a, c**are the Popular patterns. So finally, we found out the popular patterns from the incremental database.

7. Experimental Results

In order to do trials, we frequently use data sets that are ordinarily utilized as a part of regular example mining tests as the attributes of these incremental datasets are notable. Counting these datasets we likewise utilize IBM engineered datasets from www.almaden.ibm.com/cs/journeyBy utilizing this information we got steady outcomes. With the assistance of room limitation, we denote to get the exploratory outcomes on a subset of these datasets.

Table 4: Dataset

Dataset	Trans-action	Items	Max TL	avgTL	Data density
Γ10I4D100K	100000	870	29	10.10	parse
Γ20I4D100K	99996	871	42	19.81	parse
nushroom	8124	119	23	23.00	ense

Generally, every program was wrote in C and keep running in UNIX by1.3 GHz of Quad-Core processor. Projects that portrays the runtime shows the aggregate implementation time. The outcomes that are accounted for are relied upon normal numerous keep running for each situation. In the beneath tests the Incrpop-trees were developed utilizing slipping request of item tallies of occurrence.
This Incrpop-tree is the main way to deal with mine popular patterns since incremental database construct situated in light of our insight. The execution of Incrpop-tree structure and Incrpop-development calculation while changing the parameters of mining, for example, threshold Popularity and dataset qualities have been introduced.

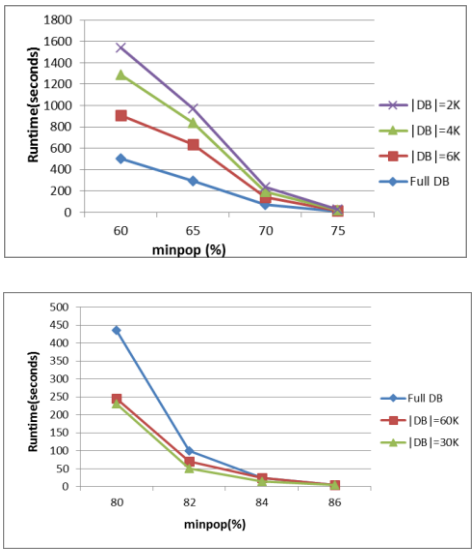


Fig. 6: Runtime

Runtime

The implementation time for the Incrpop growth required for the mining of popular patterns in datasets of various sorts and the changes in min_popp has been accounted for. The compiling time incorporates every one of the means of the H-table development, the Incrpop tree building and the comparing mining. From above picture we display the outputs on single short dataset and the other output on a dense dataset.
The impact of mining the differences in length of such datasets can be seen through execution popular patterns mining at the same time expanding the extent of both the datasets.

Reduction on the number of patterns when changing min_popp

We also inspected amount of patterns created using the Incrpop-growth which is similar to our previous experiment by varying the datasets size and min_popp. The graph shows that as the min_popp values is been increasing there is a decrease in quantity of patterns shown in percentage. Min_popp values has been taken on x-axes whereas percentage is changed in number of patterns are taken on y-axes.

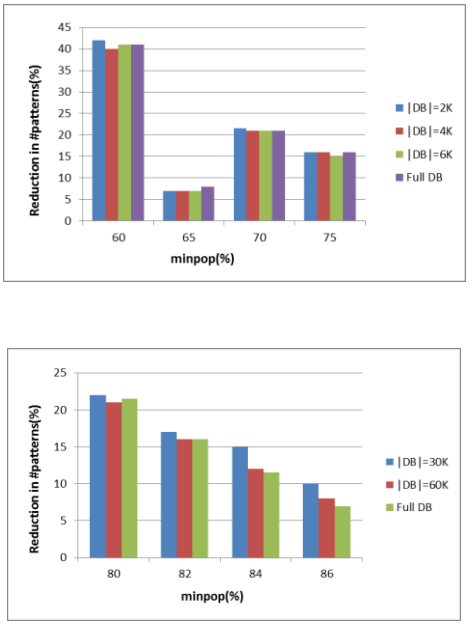


Fig. 7: reduction of number of patterns when changing min_popp

Consider the first graph, from this we can observe that when the min_popp value has been increased to 65% to 70% there is a drastic decrease in the quantity of patterns. While considering the second graph as the min_popp value increases from 86% to 88% then there is a sudden decrease in the percentage of number of patterns.

Incrpop tree compactness

In this segment we note the minimization of Incrpop-tree as far as quantity of Incrpop-tree nodes. Since the mushroom dataset have defined amount of transaction size, maximum length of transaction used for each conceivable example is constantly same. Since each thing experiences languid pruning stage so everything tree size is same with shifting the min_popp. By the expansion of min_popp the extent of tree structure slowly minimized in T20I4D100K. By growth in size of database, amount of the node will be improved into a dense and sparse datasets.

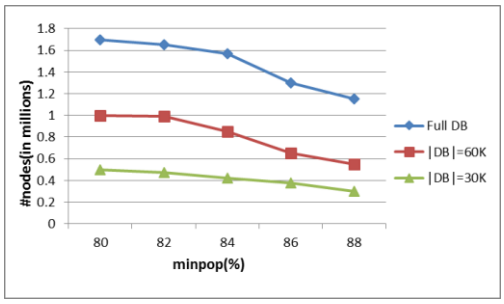
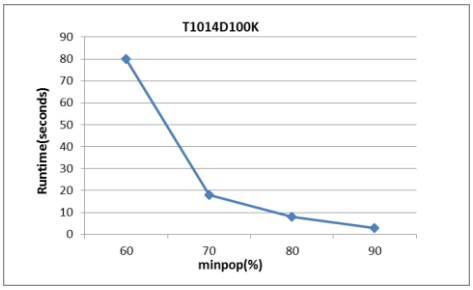


Fig. 8: compactness of the Incrpoptree node count on T20I4D100K



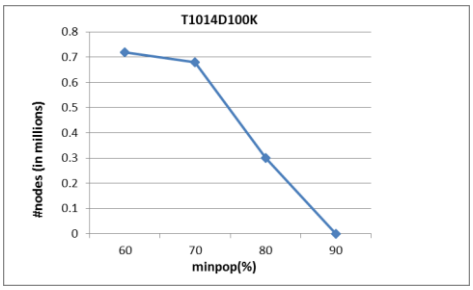


Fig. 9: scalability of Incrpop-growth

Scalability of Incrpop-growth

For understanding the Incrpop-growth mining method, we additionally implemented an algorithm on T10I4D100K. The below graph represents the results on scalability test on varying min_popp and number of nodes. We can observe that as the min_popp reduces complete tree structure, mining time, essential memory increases. The result show that Incrpop-tree can be capable to mine arrangement of Popular patterns on these type of datasets for sensible little estimation of the
Threshold Popularity through specific implementation period.

8. Conclusion

In this paper, an innovative type of pattern called popular pattern was introduced by us. The Incrpop-growth and the Incrpop-tree algorithm was also proposed by us. Even though the concept of Popularity doesn't suit for the downward closure property, we manage to tackle this problem by means of the total transaction size (sumTL) combined with predictable database allowing lazy pruning. The Incrpop-tree were showed dense, scalable and space effective for dense and sparse datasets in experimental outcomes. The structure of mining of the Popular pattern's and the Incrpop tree were shown time effectual based on the results.

References

[1] Leung C.K.-S., Jiang, F.: Frequent Pattern mining from Time-Fading Streams of Uncertain Data. In: Cuzzocrea, A., Dayal, U. (eds.) DaWaK 2011. LNCS, vol. 6862, pp. 252-264. Springer, Heidelberg.

[2] Agarwal, R., Srikant, R.: Fast algorithms for mining association rules In: VLDB 1994, pp. 487-499 (1994)

[3] Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD 2000, pp. 1-12 (2000).

[4] R. Agarwal, T. Imielinski, A. Swamy. Mining association rules between sets of items in large databases. In ACM SIGMOD Int. Conference on Management of Data, pp. 207- 216(1993).

[5] Cameron, J.J., Leung, C.K.-S., Tanbeer, S.K.: Finding strong groups of friends among friends in social networks. In: IEEE DASC 2011, pp. 824-831(2011).

[6] F. A. Anour et al., IMTAR: Incremental Mining of General Temporal Association Rules. Journal of Information Processing System. Vol 6 no.2 pp. 163-176. (2010)

[7] N. Y. Eltabakh et al. Incremental Mining for Frequent Patterns in Evolving Time Series Databases. Prudue University, prudue-e-pubs, Computer Science Technical Reports. Pp-1-37 (2008).

[8] G.Vijay Kumar, ValliKumari, Incremental Mining for Regular Frequent Patterns in Vertical Format. International Journal of Engineering and Technology, Vol 5, pp.1506-1511.

[9] G. Vijay Kumar, ValliKumari, IncMaRFI: Mining Maximal Regular Frequent Item set in Incremental Databases. International Journal of Engineering Science and Technology, Vol 5, No. 8, 2013

[10] Rasheed, F., Alshalalfa, M., Alhajj, R.: Efficient periodicity mining in time series databases using suffix trees. IEEE TKDE 23(1), 79-94 (2011)

[11] S. K. Tanbeer et al., Mining Regular Patterns in Incremental Transaction Databases. 12th International Asia-Pacific web conference, (2010) IEEE, DOI 10.1109/APWeb. 2010.68, pp.375-377.

[12] G.Vijay Kumar et al, Mining of Popular Patterns from Transactional Database.

[13] G.Vijay Kumar, M.Sreedevi, NVS.Pavan Kumar. Mining Regular Patterns in Transactional Databases using Vertical Format, International Journal of Advanced Research in Computer Science, Vol 2, pp.581-583.

[14] G.Vijay Kumar, V.ValliKumari, Parallel and distributed frequent-regular pattern mining using vertical format in large databases, IEEE Xplore, IET-(2012), pp-110-114.

[15] Leung, C.K.-S., Sun, L.: A new class of constraints for constrained frequent pattern mining. In: ACM SAC 2012, pp. 199-204 (2012).

[16] G.Vijay Kumar, V.ValliKumari, MaRFI: Maximal Regular Frequent Item set Mining using a pair of Transaction-id's, International Journal of Computer Science & Engineering Technology, Volume (4), Issue (7)-2013.

[17] G.Vijay Kumar, V.ValliKumari, Incremental Mining for regular-frequent patterns using vertical format, International Journal of Engineering and Technology, Volume 4, Issue 7, pp-1506-1511, 2013.

[18] Dong Woo Kim, Tae Gu Kang, Guozhong Li, SeongTaek Park: Analysis of User's Behaviors and Growth Factors of Shopping Mall using Big data, Volume 8, Issue 25, October 2015.

[19] G.Vijay Kumar, V.ValliKumari, Incremental Mining for regular-frequent patterns using vertical format, International Journal of Engineering and Technology, Volume 4, Issue 7, pp-1506-1511, 2013.