Discovering Weighted Calendar-Based Temporal Relationship Rules using Frequent Pattern Tree

Gupta Pankaj and B. B. Sagar

Department of Computer Science and Engineering, BIT, Mesra:Ranchi, off-Campus: Noida, A-7, Sector 1, Noida – 201301, Uttar Pradesh, India. pgupta@bitmesra.ac.in, drbbsagar@gmail.com

Abstract

The advent of data mining approach has brought many fascinating situations and several challenges to database community. The objective of data mining is to explore the unseen patterns in data, which are valid, novel, potentially subsidiary and ultimately understandable. The authorize and real-time transactional databases often show temporal feature and time validity life-span. Utilizing temporal relationship rule mining one may determine unusual relationship rules regarding different time-intervals. Some relationship rules may hold, through some intervals while not others and this may lead to subsidiary information. Using calendar mined patterns has already been projected by researchers to confine the time-validity relationships. However, when we consider the weight factor like utility of item in transactions and if we incorporate this weight factor in our model to mine then fascinating results of relationships come on time-variant data. This manuscript propose a narrative procedure to find relationship rule on time-variant-weighted data utilizing frequent pattern tree-hierarchical structures which give us a consequential benefit in expressions of time and memory-recollection utilization though including time and weight factor.

Keywords: Data-Mining, Temporal Association Rule-Mining, Temporal Data-Mining, Time-Weight-Carry Mining, Temporal-Weighted Relationship Rules, Weight-Carrying Transaction

1. Introduction

Well-organized algorithm¹ for discovery recurrent patterns is one of the input successes of data-mining process. Apriori's algorithm^{2,3} is predicated on support & confidence skeleton. It has two stages: (1) Candidate-set of items generated in all probable ways. (2) The record scanned and numbering of all transactions for every itemset. The procedure remains in loop till recurrent item set is existing. Here may be a few issues which can be raised in relation to data-mining of-

- Relationship among every pattern's of a definite kind during a categorical time gap.
- Relationship among every pattern's of a definite kind with a categorical periodicity.
- Relationship among every pattern's of definite kind with a concrete periodicity during a concrete time gap.

More over if we integrate weight i.e. utility of an itemset during its life cycle in above process of mining then what will happen?

All these verbalizations designate that data is not independent of time and its utility weight⁴. Time and utility shows a principal role in authentic dataset. In miscellaneous business, time and utility is most principal dimension for an itemset. In subsisting algorithms resembling Apriori¹ and its variant when time and weight³ aspect is mixed up in the dataset, which gives supplementary infor-mation for industry, although it also increments the time and space difficulty since now it is necessary to scrutinize the database for each legal categorical period with its minimum utility.

Our anticipated approach is to propose a wellorganized procedure for mining Relationship ruling in time-weight predicated dataset by utilizing resourceful data space device i.e. frequent pattern tree.

*Author for correspondence

Rest in this paper is as per the following. Cognate work-on at relationship mining, temporal information mining, weighted mining in databases, is given in Part 2. The temporal-weighted mining model is in Part 3. Part 4 verbalize about how temporal-weighted patterns mined utilizing FP tree. Part 5 discuss about performance studies and comparison of proposed algorithm lastly, in Part 6 we finish up and conclude the work.

2. Correlated Work

The perception of involvement & relationship ruling was given as Apriori's algorithm¹. Later on it trailed by upgrades²⁻⁵. Algorithm performance was amended by deploy recurrent pattern magnification approach⁶. In manuscript⁴ the oversight of the time dimension in relationship ruling was mention. A temporal feature of relationship-ruling was proposed by¹². As per this transactions which belong to records be time imprinted and time gap is designated by the utilizer toward split the data into disjoint fragments, similar to years, months and days. More-over repeated relationship-ruling was given via Ozden¹⁹ by means of given least support and most confidence factors.

An excellent bibliography of temporal data-mining can be originated in the Roddick text²¹. As per Roddick²¹ sense, the system foremost looks for the relationship than it is utilized to add in temporal feature. It can be utilized in position predicated & gap predicated replica of time concurrently. Recurrent pattern move toward for taking out the time responsive data be given in²⁰ where the pattern occurrence past below a time window structure is utilized to reply time responsive query. A group of item patterns along by means of their occurrence pasts are compacted & stored utilizing a tree-hierarchical arrangement identical to Frequent Pattern tree are rationalized incrementally with arriving transactions¹³. Li deal with the calendar predicated relationship ruling situation¹⁷, the outcome prove that temporal Apriori is 4 to 21 times more quick than straight Apriori's, and the finishing time prodigiously reduces with reverence to exact match.

Then the objective of utility mining is to find out all the item-sets whose utility values are beyond a user particular threshold in a transaction database.

Our methodology is consideration items' time of substantial life or lifespan and this may be the period between the earliest run through appearance and the final time the item appearance in related transactions in the database. We calculate the support of an item set characterized by temporal support and then create FP tree to mine frequent patters and their relationship. The new input of weight is that it can efficiently make out the temporal high usefulness item-sets with less candidate-item-sets such that the finishing time can be minimized competently.

3. Temporal-Weighted Relationship Model

3.1 Temporal Relationship Ruling⁴

Description: The occurrence of an item-set in excess of a time-period 'T' is the total figure of transactions in which it arise separated by whole figure of transaction in excess of that time-period. Within the identical manner, confidence of an item's set with other item's set is operation of together these objects over the time-era separated by initial item-set of that time-era.

Support (X) = Occurrence of X in particular time-gap / Total nos. of transactions in that particular time-gap.

Confidence (X => Y[T_start,T_end]) = Support_ count(X U Y) over Time gap / occurrence of X in gap.

Where, T_start points out the legal begin time and T_ end point out legal end time according to temporal-data.

3.2 Simple Schedule Mined Patterns¹⁷

While temporal sequence is pertained in provisions of years, months & days then they structure the phrase schedule representation. It can be bringing in temporal-data-mining.

A schedule representation is relational representation (in context of relational-databases) R = (F : D, F : D, F : D, F : D) together with a valid restriction. Each attribute F is a calendar unit name like year, month & day etc. Each domain D is a finite subset of the positive integers.

Within schedule model, a schedule pattern <d , d , dd > covers another pattern <d', d' , d' $\overset{n}{\dots} \overset{n-1}{\dots} \overset{n-1}{\dots} d^{n-2}$ if and only if for each I, $1 <= I <= n^{n}$ either $d^{n-2} \times r d = d^{n-1}$. Now it is our job is to extract recurrent pattern over arbitrary time-gap in provisions of schedule pattern representation.

3.3 Weight or Utility Factor

The importance of an item in itemset and a transaction in set of transactions give their utility factor. The objective of usefulness mining is to find out every the item-sets whose usefulness values are ahead of a user specific threshold in a transactional database. In^{23} , the objective of usefulness mining is defined as discovery of all high usefulness itemsets. An item-set I is a high utility itemset if u(I)=thresh-hold, where I is subset of the Itemsets and this minimum utility threshold; otherwise, it is a low utility itemset.

For instance, $u(A_{2}) = 3$, u(A, C) = u(A) + u(C) = 3 + 1 = 4, which include utility(u) of both item.

4. Proposed Algorithm

In this part, the projected algorithm will be described in details. The main purpose of the projected algorithm is to keep temporal frequent item-sets utilizing FP tree²⁰ and then mine the patterns as per utilizer thresh hold utility factor.

The algorithm employs the skeleton for design the Temporal FP-Tree for frequent temporal itemsets predicated on calendar schema and then mines the novel and valid patters with reference to utilizer utility thresh-hold factor.

Step I: // Collect the transaction either in ascending or descending order as per calendar schema. Follow the calendar schema pattern to sort out these set of transactions

Say, T=Number of sorted (on basis of calendar schema) Transaction

Step II: // Construction of Temporal FP-tree for frequent itemsets which satisfy minimum support factor within calendar schema pattern.

Let the sorted frequent –item list in the transaction be n [N] where, n is the veryfirst element and N is the left behind list.

Call to_Insert_Tree (n [N], Tree).

```
Method to_Insert_Tree (n [N], Tree).
```

```
+
```

- 1: If Tree has a child C such that
- 2: If (C.time = N.time) then // for checking interval phase I
- 3: If (C.itemname = N.itemname) then
- 4: C.count = C.count ++ // Increase counter by 1
- 5: Else generate a new node // Node shaped on the same-branch
- 6: Connect to its parent N. Counts=1 // Initialize the counter by1
- 7: Else generate a new division link from the root.

Call to_insert_tree (n, N) recursively

- } // End of Method
- Step III: // Mine the temporal frequent patterns from FP-tree.

Call Temporal Frequent Pattern growth (TFP-tree, null).

- Method Temporal Frequent Pattern growth (Tree, α)
 - {
 - 1: if Tree has a only prefix path
 - 2: next {
 - 3: let R is the only prefix-path part of Tree;
 - 4: let S be the multi-path path that contain all temporal relations of R tree
 - 5: for each combination (denoted as $\beta)$ of the nodes in the path R do
 - 6: generate pattern $\beta \cup \alpha$ with support = minimum support of nodes in β ;
 - 7: let freq pattern set(R) be the set of patterns so
 generated; }
 - 8: else let S be Tree;
 - 9: for each item 'ai' in S do { // Mining multipath FP-tree
 - 10: generate pattern β = ai $\cup \alpha$ with support = ai .support;
 - 11: construct β 's conditional pattern-base and then β 's conditional FP-tree Tree β ;
 - 12: if Tree $\beta = \emptyset$
 - 13: then call TFP-growth(Tree β , β);
 - 14: let freq pattern set(S) be the set of patterns so
 generated; }
 - 15: return(freq pattern set(R) U freq pattern set(S) U (freq pattern set(R) × freq pattern set(S)))
 - } // End of Method
- Step IV: // Calculate utility factor for frequent mined temporal patterns and check them with user thresh-hold.

Let 'FP' is the list of temporal frequent patterns mined in above step,

Do (FP < > Null)

.

If $FP \ge u(X)$ then

Novel Pattern, say NP= NP U FP //valid pat-

tern as per user requirement

Else Truncate FP & Next(FP)

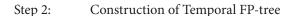
End

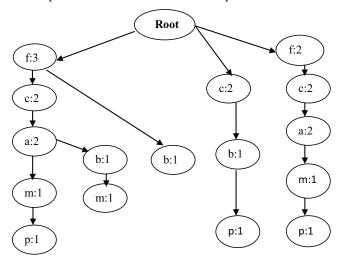
}

Example 1: Table 1 illustrates a transactional-database enclose the transaction data from March 2011 to June 2011. The quantity of transactions documented in calendar year is arrange in ascending order and then item are arranged in descending order of their frequency which support a minimum support of 2 occurrence (i.e min_support=2).

Table 1

T-id	Items Brought	Date	Schedule pattern	Items arrange in Descending Frequency
1001	f, a, c, d, g, i, m, p	14/03/2011	(14,03,11)	f, c, a, m, p
2001	a, b, c, f, l, m, o	15/03/2011	(15,03,11)	f, c, a, b, m
3001	b, f, h, j, o, w	15/03/2011	(15,03,11)	f, b
4001	b,c,k,s,p	04/06/2011	(04,06,11)	c, b, p
5001	a, f, c, e, l, p, m ,n	05/06/2011	(05,06,11)	f, c, a, m, p
6001	f, a, c, d, e	06/06/2011	(06,06,11)	f, c, a





Step 3: Mine the temporal frequent patterns from FP-tree.

Now consider, Table 3 which shows a utility-table²² for items for the above specified period, be the value connected with item 'ip' in the usefulness Table. This value is a sign of the magnitude of an item-set, which is independent of transactions.

Step 4: Calculation utility factor for frequent mined temporal patterns and check them with user thresh-hold.

For Calendar schema <*, 3, 11>

Utility Factor, for u(fb)=u(f)+u(b)=7+10=17, which shows as importance of this frequent temporal pattern (fb).

Similarly, For u(fm)=u(f)+u(m)=7+9=16, u(cm)=u(c)+u(m)=1+9=10, u(am)=u(a)+u(m)=3+9=12, u (f c m) = u (f) + u (c) + u (m) = 7 + 1 + 9 = 1 6, u (f a m) = u (f) + u (a) + u (m) = 7 + 3 + 9 = 1 9, u(cam)=u(c)+u(a)+u(m)=1+3+9=13, u(fcam)=u(f)+u(c))+u(a)+u(m)=7+1+3+9=20

Similarly, For u(fa)=u(f)+u(a)=7+3=10, u(ca) = u(c) + u(a) = 1 + 3 = 4, u(fca) = u(f) + u(c) + u(a) = 7 + 1 + 3 = 11, u(fc)=u(f)+u(c)=7+1=10,

For Calendar schema <*, 6, 11>

Utility Factor, for u(cp)=u(c)+u(p)=1+7=8, which shows as importance of this frequent temporal pattern (cp).

Now the list of Frequent pattern in different interval if user threshold of an item is more than 10.

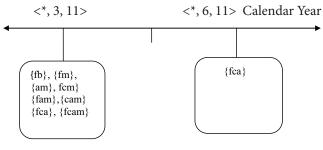


Figure 2.

5. Relative Performance Studies

Our testing was carrying out on a personal desktop Pentium 5, 3.6 MHz with 2GB RAM and we apply the projected algorithm using java's programming language.

To assess the projected algorithm we carry out a number of trials by means of artificial data produce same in a way to figure 2. This data enclose 10,339 transactions acquired in a time period of one year.

We as well evaluate the amount of recurrent patterns produced by our algorithm against TFP-Apriori, as can be

Table-2

Item	Time	Conditional	Conditional	Frequent
	Interval	Pattern base	FP-tree	item
Р	<*, 3, 11>	{(fcam:1}	Ø	Ø
Р	<*, 6, 11>	{(cb:1), (fcam:1)}	{(c:2)} p	{cp}
М	<*, 3, 11>	{(fcab:1), (fca:1)}	{(f:2, c:2, a:2)} m	{fm, cm, am, fcm, fam, cam, fcam}
М	<*, 6, 11>	{(fca:1)}	Ø	Ø
В	<*, 3, 11>	{(fca:1), (f:1)}	{(f:2)} b	{fb}
В	<*, 6, 11>	{(c:1)}	Ø	Ø
А	<*, 3, 11>	{(fc:2)}	{(f:2, c:2)} a	{fa, ca, fca}
А	<*, 6, 11>	{(fc:2)}	{(f:2, c:2)} a	{fa, ca, fca}
С	<*, 3, 11>	{(f:2)}	{(f:2)} c	{fc}
С	<*, 6, 11>	{(f:2)}	{(f:2)} c	{fc}
F	<*, *, 11>	Ø	Ø	Ø

Table 3

Items	Item per unit profit	Items	Item per unit profit	Items	Item per unit profit
a	3	g	5	m	9
b	10	h	3	n	2
c	1	Ι	4	0	4
d	6	j	8	р	7
e	5	k	3		
f	7	1	6		

seen in Figure 3 which shows less no. pattern generated as compare to existing and thus reduces the of memory-intake for execution.

For our last test, we estimate the increase performance of the projected algorithm. In this test we used a diverse transactional datasets, where 'Dx' indicate the volume of the data-set considered; we also altered the support threshold value from 5% to 20%. While seeing in Figure 4 & Figure 5, the finishing time somewhat increases as the volume of the data increase, however our proposed algorithm shows good scalability over TFP-Apriori.

6. Conclusions and Future Work

In this manuscript, we have projected an incipient mix approach for data mining procedure. With the develop-

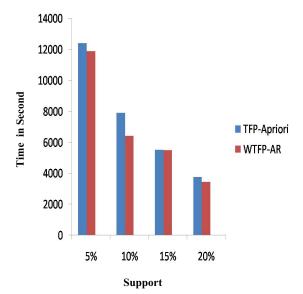


Figure 3. Frequent Patterns Generated.

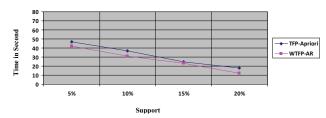


Figure 4. Comparison for Dx=11500.

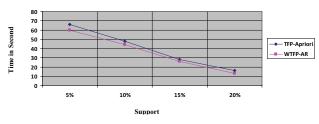


Figure 5. Comparison for Dx=17000.

ment of technology, the arduousness to obtain data is decrementing and the arduousness to analyze the astronomically immense volume of data is incrementing. The projected algorithm's provides an competent time responsive approach for mining recurrent items in dataset. Temporal FP-Tree with utility of an itemset as weight proposed in this work discovers frequent patterns during the time gaps designated by schedule schemas.

Projected system is the improvement of Temporal-Frequent-Pattern (TFP) method with weighted constraint of item's utility in the transaction. In tribulation result it is demonstrated that the a-few issues can be prosperously settled with the assistance of Proposed Algorithm. We likewise had demonstrated the completing time examination by charts with Apriori's Algorithm, which demonstrates proposed algorithm, takes a lesser amount of time to Apriori's.

By this with the assistance of conception of time and other weighted constraint, we consider the frequent patterns that have enough support in their lifespan period with the required their utility in transactional database.

7. References

- Agrawal R, Mannila H, Srikant R, Toivonen H, Verkano I. MIT Press: Fast Discovery of Association Rules. 1996.
- Srikent and Agrawal. Almaden: International Business Machine: Fast Algorithms for Mining Association Rules. 1994.
- 3. Agrawal and Srikant R. Mining Sequential Patterns. Proceeding IEEE International Conference on Database Engineering. 1995.
- 4. Juan M Ale and Gustavo Rossi. An approach to discovering temporal Association Rules. Association for Computing Machinery symposium on Applied Computing. 2000.
- Book::Jiawei Han, Kamber and Jian Pei. The Morgan Kaufmann Publishers: Data Mining: Concepts and Techniques, 3rd Edition. 2011.
- 6. Lee Chang, Cheng-Ru Lin and Ming-Syan Chen. Slidingwindow filtering: An Efficient algorithm for Incremental Mining. Proceeding of Association for Computing Machinery International Conference on Information and Knowledge Management. 2001.
- 7. Xiaoxin and Jiawei. San Francisco: Classification based on predictive association rule. 2003.
- 8. Thabtah Abdelijaber, Peter C and Yonghong. Comparison of classification techniques for a personnel scheduling problem. International Business Information Management Conference. 2004.
- 9. Tunc and Dag, Generating classification association rules with modified Apriori. Spain: International Conference on Artificial Intelligence. 2006.

- Book:: Ian H Witten, Eibe Frank and Mark A Hall. Morgan Kaufmann Publishers: Data mining: Practical Machine learning tools and techniques. 2000.
- Wei Wang, Jiong Yang and Philip S Yu. Efficient mining of weighted association rules. Proceeding of 7th SIGKDD International Conference on Knowledge Discovery and Data Mining. 2000.
- Coenen Frans and Leng Paul. An Evaluation of Approaches to Classification Rule Selection. Proceeding of 4th IEEE International Conference on Data Mining, 2004.
- 13. Ayad Ahmed, El-Makky Nagwa and Taha Yousry. Incremental Mining of Constrained Association Rules. Proceeding of SIAM Conference on Data Mining. 2001.
- 14. Weka:: Data Mining Software. Available from: http://www.cs.waikato.ac.nz/ml/weka.
- Rymon R. Search Through Systematic Set Enumeration. Proceeding 3rd International Conference Principles of Knowledge and Reasoning. 1992.
- Book:: Chen X, Petrounian I. Knowledge Discovery and Data Mining. Chapter 5-A Development Framework of Temporal data Mining. 2001.
- 17. Ozden B, Ramaswamy S and Silberschatz A. Cyclic Association Rule. Proceeding of 14th International conference on Data Engineering. 1998.
- Jiawei Han, Jian Pei, Yiwen Yin and Runying Mao. Mining Frequent Pattern without Candidate Generation. Proceeding ACM-SIGMOD International Conference Management of Data. 2000.
- 19. Roddick JF, Hornsby K, Spiliopoulou M. An Updated Bibliography of Temporal, Spatial, and Spatio-temporal Data Mining Research, TSDM. 2000.
- 20. Hong Yao, Howard J Hamilton and Cory J Butz. A foundational Approach to Mining Item-set Utilities from Databases. Florida, USA: Proceedings of the 4th SIAM International Conference on Data Mining. 2004.