Role of Segment Progressive Filter in Dynamic Data mining

Mohsin Naqvi[1], Kashif Hussain[1], Sohail Asghar[1], Simon Fong[2]
[1]Center of Research in Data Engineering (CORDE)
Mohammad Ali Jinnah University
Islamabad, Pakistan
sohail.asg@gmail.com

[2]Department of Computer and Information Science
Faculty of Science and Technology
University of Macau, Macau SAR
ccfong@umac.mo

Journal of Digital Information Management

ABSTRACT: *Association rule mining perhaps the most widely described technique among the minding paradigms. The temporal association rule mining in the association rule mining tries to find relations among items in datasets. The temporal association mining has strength in detecting the dynamic nature of databases. Unfortunately the current mining methods ignore the consideration of database content updates. In the current research we have introduced the Incremental Standing method for Segment Progressive Filter (ISPF). The proposed technique can support the database update and mine updated datasets. We prove that the proposed algorithm is an optimal way of mining. We have applied the scan reduction technique to generate all candidate k-item sets to form 2-candidate item sets directly. We also bring good experimentation to validate and document clearly the algorithm with good examples.*

Categories and Subject Descriptors
H.2.8 [Database Applications]: Data mining; **I.1.2[Algorithms]:** Analysis of algorithms

General Terms: Database mining, Association rule mining, Scan reduction technique

Keywords: Temporal association rules, Segment Progressive Filter, Mining algorithms

Received: 11 March 2011, Revised 29 April 2011, Accepted 16 May 2011

1. Introduction

As early in 1993 Agrawal has introduced the Association rule problem [2] which is capable of finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories. Association relationship is useful in selective marketing, decision analysis and market basket analysis fields [1]. Several techniques have been developed for mining association rules [17] such as FP-Growth algorithm [18], mining of generalized and multi-level rules [19], constraint based rule mining and mining multi- dimensional rules [20].

Temporal association rule mining is first introduced by Wang, Yang and Muntz in years 1999-2001 together with the introduction of the TAR (Temporal Association Rule) algorithm [3]. Temporal association rule mining has been introduced in order to solve the problem on handling time-series by including time expression into association rules [4]. Temporal association helps to find the valuable relationship among the different item sets, in temporal database. Temporal association rules are largely different from traditional association rules by the fact that temporal association rules attempt to model temporal relationships in the data. There are different types of temporal association rules defined in the literature such as inter-transaction rules, episode rules, trend dependencies, sequence association rules and calendric association rules.

There are many interesting papers such as [4, 5, 6] in which most of existing techniques are developed based on temporal content analysis. New TAR algorithms that have been presented for general temporal association rule mining in database are PPCI algorithm [11], SPF [9], and ITRAM [4]. Temporal association rules have various kinds like Calendric Association rule [12], Cyclic Association rule [13], Association rule based on partition [14], progressive weighted miner [10], incremental temporal association rule [4] and periodic temporal association rule [15].

Temporal databases are known to be continually being updated or appended. Temporal association rule mining must synchronize with these updated transactions, without any loss of time granularity. Existing temporal association rule mining techniques cannot deal with the upcoming transactions of database as they might operate in batches. New rules may get omitted, and we need to address this issue. Let n be the number of partitions and m be the number of updates of the database. We need to generate the temporal association rules without loss of time granularity. In order to solve this problem, the INCREMENTAL STANDING FOR SEGMENT PROGRESSIVE FILTER (ISPF) algorithms are proposed. ISPF effectively divide the database item set with their common start and end times. It is a common phenomenon that items in the real transition database have their dissimilar exhibition periods.

The rest of paper is organized as follows. Section 2 provides the review of some related works. Section 3 describes the proposed algorithm. Performance result is shown in section 4. And section 5 gives the conclusion of the paper.

2. Early Studies

Several algorithms have been proposed for mining the temporal association rules in temporal database. Among these algorithms, Tarek et al proposed the ITARM algorithm to discover the temporal frequent item set after the temporal transaction database has been updated [4]. The basic idea of ITARM algorithm depends on previously generated 2-candidate

item set with their supports. ITRAM works as it checks first the extension of the pervious partition and attempts to find 2-candidate item set from the new partition; if it succeeds then it merges the current partition with the pervious partition, and from there it finds the 2-candidate item set. This approach is basically introduced to facilitate incremental mining techniques over an ever updating transaction database.

Another approach proposed by C. H. Lee et al, is progressive partition miner (PPM) [6]. In PPM the database is first partitioned by the size of time granularity. The PPM algorithm is applying with a filtering threshold mechanism on each partition of the database to prune out those cumulatively infrequent 2-itemsets. PPM also employs database scanning reduction technique. However, the limitation of this technique is its ability to deal with problems of incremental mining.

♦Moreover J. M. Ale et al expands the notion of association rule incorporation of the time to their frequent item sets [7]. Thus it tries to extend the existing non-temporal mining model by introducing the concept of temporal support. Discovery of association rule is done in a two-phase process; it first finds the frequent item set according to the lifespan of the item set and secondly it uses these frequent item sets to generate the rules. These rules are checked based on the confidence. In this proposed technique it however does not consider the updates of the database.

M. Chen et al developed a temporal association rule model to be used in video database for video event detection [5]. In this approach it captures the characteristics of temporal patterns with respect to the event of interest. M. Chen et al proposed their framework based on feature extraction, hierarchical temporal association mining and multimodal data mining. Often traditional association rule mining approaches use a manually assigned threshold. The advantage of Chen's model is the use of an adaptive mechanism for determining the essential threshold.

Byon et al proposed an Exponential Smoothing (ES) filter for temporal association rule mining [8]. ES filter takes two steps; one is to partition the database and then feed them into a Progressive Weighted Miner (PWM). PWM has a weight function that gives greater weights to recent data than old data; each is divided by equal period [10].

Ru Miao et al presented the idea of Apriori-extended mining periodic temporal association rules (MPTAR) [15]. Previously techniques of TAR did not consider the individual item exhibition period. MPTAR solved this problem, by including the exhibition period of individual item. Again MPTAR is a two-step periodic rule mining mechanism. The first step is mining the trend of continues attribute through cycle curve and the second step is calculating the period of the attribute. MPTAR did not define the cumulative threshold, and it is short of embracing upcoming transaction entries in the association rule mining.

Edi Winarko et al invented a new algorithm called, ARMADA (mining richer temporal association rules from interval-based data) [16]. While reading the database into memory, it counts the support of each state and generates frequent 1-patterns. By using a recursive find-then-index strategy, the algorithm discovers all temporal patterns from the in-memory database.

3. Algorithm for ISPF

There are two major challenges in general temporal association rules methods which we will have to overcome. The first major challenge is to tackle the problem of updating the association rules while temporal database are continually being updated. The second challenge is the exhibition period of the item set in the database that should be allowed to differ from one to another. In the light of these challenges, we combine ISPF into TAR mining. ISPF consists of three major procedures; one is the database updating, the second is database segmentation and the third is candidate generation from the segments.

At each ending interval of the database update, the database is divided based on the imposed time granularity. The database is divided in the light of item set's common start and finish period. Then it checks the latest update of the database. By using this technique the number of segmentation is minimized and it is small when compared to the other previous methods. This feature provides the capability of filtering the candidate item set in either the forward or backward direction. After the segmentation it generates the 2-candidate item set in each sub database. When all sub databases are processed, all theses 2-candidate item set are merged in union. After this, scan reduction technique is utilized over these candidate item sets and it generates the k-item set. As the last step of the algorithm, when all the k-item sets are generated, TIS and SIS are computed, and it counts the support for each rule.

We present the proposed algorithm, ISPF that is to be used for mining incremental temporal association rule, in the form of pseudo code as below. The advantage of this technique is its ability to deal with the problem of incremental mining techniques in mining temporal association rules.

Algorithm: ISPF

```
Input:  transaction database DB, minimum
support, time granularity, update of database
db.

Output:    frequent item sets.

Step#1:    Divide the database based on the
imposed time granularity.
  Step#2: Check the update of the database.
          If the database is updated, append
          this update with pervious database
          transactions.
  Step#3: Partition (in the light of
          exhibition period) the database
          based on either common star or end
          time
Step#4:    Generate the 2candidate item set
from each sub database
       Step#4.1:  Merge the new and pervious
                  partition frequent
                  2-candidate item set.
       Step#4.2:  Count the relative support
of each item set.
       Step#4.3:  Apply pruning.
       Step#4.4:  Proceed to next partition
       Step#4.5:  Go to step#4.1.
Step#5:    Generate the k-item set through scan
reduction technique.
Step#6:    Count the support and apply pruning.
Step#7:    Generate the sub candidate item.
Step#8:    Count the support.
Step#9:    Prune.
```

Figure 1. Proposed Algorithm ISPF

4. Experimental Results

The efficacy of algorithm ISPF is tested by a given case study. Consider the transaction database shown in the Table 1. A set of time series in the database indicate the transaction records dated from January to March. They are the archival records which have already existed. A new portion of transactions that represent the incremental update of the database is recorded in the month of April. These transactions are shown in the last part of the table. Minimum support 30% and minimum confidence 75% are set for the experiment. The scanning direction of partitions 1 and 2 are from left to right, whereas the direction of partitions 3 and 4 are from right to left.

	P1	Date	TID	Item set
Database		Jan 03	TID1	A F
			TID2	D C F
			TID3	A C F
			TID4	A D
	P2	FEB03	TID5	C D
			TID6	B C D F
			TID7	A B C
			TID8	B
	P3	MAR03	TID9	E F
			TID10	B C F
			TID11	A B
			TID12	A E
UPDATE DATABASE	P4	APR03	TID13	AB
			TID14	A B E
			TID15	E
			TID16	B

Table 1. Transaction database sample used in the experiment

Table 2 illustrates the start time and end time of the item set. By using this information, the database is partitioned based on either the common starting time or common ending time which could be optionally chosen by the user. The choice has little difference on the results when the sample size is large enough.

The results of the database partitioning are shown in Table 3. They include the partition 1-candidate item sets, their supports and the partition number of each candidate item set. The support values of the AD, AF, CF candidate item sets are equal to the defined threshold; AC and DF are pruned because their support values are lower than the defined threshold.

Item	Start	End
A	Jan-02	Apr-02
B	Feb-02	Apr-02
C	Jan-02	Mar-02
D	Jan-02	Feb-02
E	Mar-02	Apr-02
F	Jan-02	Mar-02

Table 2. The start and finish time of the item sets

P1		
C	**Start**	**Count**
AC	1	1
AD	1	2
AF	1	2
CF	1	2
DF	1	1

Table 3. Partition 1

Table 4 demonstrates the partition 1 frequent candidate item sets and partition 2 candidate item sets. Their support and partition values of the candidate item set are shown. The supports of the CF, BC, BD and CD candidate item sets are equal to or higher than the defined threshold; other item set are pruned because their support are less than the defined threshold.

P1+P2		
C	**Start**	**Count**
AD	1	2
AF	1	2
CF	1	3
AB	2	1
BC	2	2
BD	2	2
BF	2	1
CD	2	2

Table 4. Partition P2+P3

Table 5 illustrates the partition 4 frequent candidate item sets, as well as their support and partition values. AB is the only candidate item set that is qualified by the given minimum support; other item set are pruned because of their low support values.

P4		
C	**Start**	**Count**
AB	4	2
AE	4	1
BE	4	1

Table 5. Partition 4 candidate item set 1

Table 6 shows the partition frequent candidate item set of 4 & 3 candidate item sets, the support values and partition numbers of the candidate item set. AB and EF are the only candidate

P3+P4		
C	**Start**	**Count**
AB	4	3
AF	3	1
BF	3	1
CE	3	1
CF	3	1
EF	3	2

Table 6. Partition 4 & 3 candidate item sets

item sets that meet the defined threshold, other item sets are pruned away because their supports are less than the defined threshold.

After the scanning through all the sub databases, the resulting frequent candidate sets are AB BC BD CD CF and EF. Using scan reduction technique it generates k-item set. BCD and CDF are generated as a result.

The problem of mining temporal association rules basically consists of two steps. Firstly it generates all frequent maximal temporal item sets called TIS, and the corresponding temporal sub-item sets namely SIS. SIS are generated based on these TIS. Both TIS and SIS item sets carry relative supports that would have to be greater than the predefined minimum value. The subsequent step is to derive all the frequent general temporal association rules that are frequent enough to meet the minimum required confidence value. Generating the frequent general temporal association rules is simple when the frequent TIS and SIS and their corresponding support values are known by scanning the whole database once.

In our experiment, a list of SIS and TIS candidate item sets are generated and their support values are shown in Table 7. We can observe that SIS are subset of the frequent item sets TIS. The qualified SIS candidates are A(2,4), B(2,3), B(2,2), B(2,4), C(1,2), C(1,3), C(2,2), D(2,2), C(2,3), D(1,2) and TIS candidate item sets are AB(2,4), BC(2,3), BD(2,2), CD(2,4), CF(1,3), EF(3,3), BCD(2,2).

Candidate item set		count
SIS	A(2,4)	4
	B(2,3)	5
	B(2,2)	4
	B(2,4)	6
	C(1,2)	5
	C(1,3)	6
	C(2,2)	3
	C(2,3)	4
	D(1,2)	4
	D(2,2)	2
TIS	AB(2,4)	3
	BC(2,3)	2
	BD(2,2)	2
	CD(2,4)	2
	CF(1,3)	4
	EF(3,3)	1
	BCD(2,2)	2

Table 7. SIS and TIS

The following table shows the information about the final frequent candidate item sets. The support values of both SIS and TIS, and their start and end partition information are shown. They are the ingredients for temporal association rule mining.

Item set	S	E	S	E	TI'S
AB	A		B		
	1	4	2	4	AB(2,4)
	B		C		
BC	2	4	1	3	BC(2,3)
	B		D		
BD	2	4	1	2	BD(2,2)
	C		D		
CD	1	3	1	2	CD(2,4)
	C		F		
CF	1	3	1	3	CF(1,3)

	E		F		
EF	3	4	1	3	EF(3,3)
	B		C	D	
BCD	2	4	1	3	1 2 BCD(2,2)

Table 8. Temporal item sets

Temporal association rule mining techniques are generating the rules based on the temporal information of transactions. The process is the same as the existing technique that was already published in previous research papers. In our case, the database is continuously being updated; the updates result in a number of useful new rules that can potentially be extracted, but they are neglected by the existing techniques of temporal association rules. To alleviate this issue, our proposed ISPF algorithm is used. The results from our experiment demonstrate the significance of the proposed work. For instance, in the case study, P4 contains the updated transactions of the database. The existing techniques would have generated the rules based on the database partitions P1, P2 and P3. These techniques however do not cater for the P4 (which holds the updated transactions of database). When the latest part of the database transactions is omitted, intuitively the resulting rules would miss out the elements of the latest information; therefore it is losing their timeliness and appeal in the knowledge discovery process. Our proposed algorithm generates rules covering all parts of the database, from P1 to P4. The tables above show that the proposed algorithm generated the most updated rules. These updated rules may contribute to effective and complete decision-making.

5. Summary and Future Work

Temporal association rules mining is a technique that incorporates the temporal characteristics in the association rule mining process over the frequent item sets. Temporal databases are known to be continually updated in reality. Existing temporal association rules mining techniques have not covered the most recently updated part of the data and hence the temporal association rules miss out the latest information elements. In this paper we explored the problem of incremental mining problem in general. In particular, we proposed the INCREMENTAL STANDING FOR SEGMENT PROGRESSIVE FILTER (ISPF) algorithm in order to align the updating of the database and the temporal association rules mining. ISPF first divides the database according to the common start and

end times of the item sets, and it considers the updates of the temporal association rules. The case study results show the significance of the ISPF which performs better than the existing techniques and overcome the existing problem.

Our new method called ISPF theoretically should work with other variants of temporal association mining, as an add-on process rather than a revolutionary replacement. More complex examples would be tested in the future, as well as investigating the possibility of integrating ISPF into other mining algorithms.

As a future work, we opt to automate the proposed algorithm for real world applications domains, such as finance, marketing, medical, and security monitoring where real-time information streaming is typical and results of temporal association rules are critical. The other direction is to enhance the current user-interface of the temporal association mining program which facilitates the end-user to obtain temporal patterns and rules from relational temporal databases easily with a press of button. The temporal elements of the rules should be automatically evaluated and visualized for easy referencing.

References

[1] Tan, Pang-Ning., Steinbach, Michael., Vipin Kumar (2006). Introduction to Data Mining, Pearson Addison.

[2] Agrawal, R., Imielinkski, R.T., Swami, A (1993). Mining Association Rule Between sets of items in large database, In: Proceeding of ACM SIGMOD, p. 207-216.

[3] Wang. W., Yang. Y., Muntz. R.(1999). Temporal Association Rules with Numerical Attributes. NCLA CSD Technical Report 990011.

[4] Gharib, Tarek F., Nassar, Hamed Taha., Mohamed., Abraham, Ajith. (2010). An efficient algorithm for incremental mining of temporal association rules, Data & Knowledge Engineering Elsevier, p. 800-815.

[5] Chen, Min., Chen, Shu-Ching., Shyu, Mei-Ling. Hierarchical Temporal Association Mining for Video Event Detection in Video Databases.

[6] Lee, Chang-Hung., Lin, Cheng-Ru., Chen, Ming-Syan (2002). On Mining General Temporal Association Rule in Publication of database, ICDM, 2002.

[7] Ale, Juan M., Rossi, Gustavo H. (2000). An Approach to Discovering Temporal Rules, ACM March 2000.

[8] Byon, Lu-Na., Han, Jeong-Hye (2005). Fast for Temporal Association Rule in a Large Database, key Engineering Materials, p. 287-279.

[9] Chang, Cheng-Yue., Chen, Ming-Syan., Lee, Chang-Hung (2002). Mining General Temporal Association Rule for item with different exhibition period, IEEE.

[10] Lee, Chang-Hung., Ou Jian Chih., Chen, Ming-Syan (2003). Progressive Weighted Miner: An efficient method for time constraints mining, In: Proceedings of the Advance in Knowledge discovery and data mining: 7th Pacific –Asia conference, PAKDD, Seoul, Korea.

[11] Pandey, Anjana., Pardasani, K. R (2009). PPCI algorithm for mining temporal association rules in large database, International Journal of information and Knowledge.

[12].lin, Y., Ning, P (2001). Discovering Calendric based temporal association rule, In: Proceeding of the 8th international symposium on temporal and reasoning.

[13] Ozden, B., Ramaswamy, S., Silberschatz, A (1998). Cyclic Association rule, In: Proceeding of international conference on data engineering, p. 412-421.

[14] Chen, X., Petrounias, I (1998). A framework for temporal data mining, In proceeding 9th international conference on database and expert system application, DEXA.

[15] Miao, Ru., Shen, Xia-Jiong (2010). Construction of Periodic Temporal Association Rules in data mining, Seventh International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010) IEEE.

[16] Winarko, Edi., Roddick, John F. (2006). ARMADA – An algorithm for discovering richer relative temporal association rules from interval based data, Data & Knowledge Engineering Elsevier, p. 76–90.

[17] Agrwal, R., Srikant, R. (1994). Fast algorithm for mining association rules in large database, In: Proceeding of 20th international conference on very large databases, p. 478-499.

[18] Han, J., Pei, J., Vin, V (2000). Mining frequent pattern without candidate generation, In: Proceedings of 2000 ACM SIGMOD int. conference on management of data, p. 486-493.

[19] Han, J., Fu, V. (1995). Discovery of multiple level association rule from large database, In: Proceedings of the 21th international conference on very large databases, p. 420-431.

[20] Tung, A. H., Han, J., Lakshmanan, L. S., R. Ng, R (2001). Constraints based clustering in large databases, In: Proceeding of 2001 International conference on databases theory.