Maintenance of Generalized Association Rules Based on Pre-large Concepts

TZUNG-PEI HONG †, TZU-JUNG HUANG ‡ and CHAO-SHENG CHANG ‡
†Department of Electrical Engineering, National University of Kaohsiung
‡Department of Information Management, I-Shou University
Kaohsiung , Taiwan, R.O.C.

Abstract: - Due to the increasing use of very large databases and data warehouses, mining useful information and helpful knowledge from transactions is evolving into an important research area. In the past, researchers usually assumed databases were static and items were on a single level to simplify data mining problems. Thus, most of algorithms proposed focused on a single level, and did not utilize previously mined information in incrementally growing databases. Items in real world applications are, however, commonly with taxonomy. This paper thus proposes a maintenance algorithm for generalized association rules with taxonomy based on the concept of pre-large itemset. A pre-large itemset is not truly large, but promises to be large in the future. A lower and an upper support threshold are used to realize this concept. The two user-specified upper and lower support thresholds make the pre-large itemsets act as a gap to avoid small itemsets becoming large in the updated database when new transactions are inserted. The proposed algorithm doesn't need to rescan the original database until a number of transactions have been newly inserted. If the database has grown larger, then the number of new transactions allowed will be larger too.

Key-Words: - data mining, generalized association rule, taxonomy, large itemset, pre-large itemset.

1 Introduction

Deriving association rules from transaction databases is most commonly seen in data mining [1][2][8][9][10][11][12][14][15]. It discovers relationships among items such that the presence of certain items in a transaction tends to imply the presence of certain other items. In the past, Agrawal and his co-workers proposed several mining algorithms for finding association rules in transaction data based on the concept of large itemsets [1][2][15]. Then, many algorithms for mining association rules from transactions were proposed, most of which were executed in level-wise processes.

Cheung and his co-workers proposed an incremental mining algorithm, called FUP (Fast UPdate algorithm) [5], for incrementally maintaining association rules mined. Hong *et. al.* thus proposed a new mining algorithm based on two support thresholds to further reduce the need for rescanning original databases [13]. It uses a lower and an upper support threshold to reduce the need for rescanning original databases and to save maintenance costs.

Most mining algorithms focused on finding association rules based on a single-concept level in which the items considered had no hierarchical relationships. Items in real-world applications are, however, usually organized in some hierarchies and can be represented using hierarchy trees. Mining multiple-concept-level rules may lead to discovery of more general and important knowledge from data. In this paper, we adopt Hong et al's pre-large itemsets and Srilant and Agrawal's mining approaches to efficiently and effectively maintain generalized association rules on a taxonomy. The concept of pre-large itemsets is used to reduce the number for rescanning original databases and to save maintenance cost [13].

2 Mining Generalized Association Rules

Previous studies on data mining focused on finding association rules on a single-concept level. However, mining generalized association rules on multiple levels may lead to discovery of more generalized knowledge from data. Relevant item taxonomies are usually predefined in real-world applications and can be represented by hierarchy trees. Terminal nodes on the trees represent actual

items appearing in transactions; internal nodes represent classes or concepts formed by lower-level nodes. A simple example is given in Fig.1

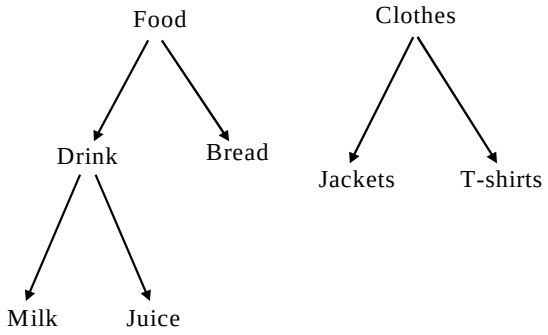

Fig.1: An example of predefined taxonomic structures

In this example, the food falls into two classes: drink and bread. Drink can be further classified into milk and juice. Similarly, assume clothes are divided into jackets and T-shirts. Only the terminal items (milk, juice, bread, jacket, and T-shirt) can appear in transactions.

Srilant and Agrawal proposed a method for finding generalized association rules on multiple levels [16]. Their mining process can be divided into four phases. In the first phase, ancestors of items in each given transaction are added according to the predefined taxonomy. In the second phase, candidate itemsets are generated and counted by scanning the expanded transaction data. In the third phase, all possible generalized association rules are induced from the large itemsets found in the second phase. The rules with calculated confidence values larger than a predefined threshold (called the minimum confidence) are kept. In the fourth phase, uninteresting association rules are pruned away and interesting rules are output according to the following three interest requirements:

1. a rule has no ancestor rules (by replacing the items in a rule with their ancestors in the taxonomy) mined out;
2. the support value of a rule is *R*-time larger than the expected support values of its ancestor rules;
3. the confidence value of a rule is *R*-time larger than the expected confidence values of its ancestor rules.

3 Related Maintenance Algorithms

In real-world applications, transaction databases grow over time and the association rules mined from them must be re-evaluated because new association rules may be generated and old association rules may become invalid when the new entire databases are considered. Designing efficient maintenance algorithms is thus important.

In 1996, Cheung proposed a new incremental mining algorithm, called FUP (Fast Update algorithm) [5,7] for solving the above problem. Using FUP, large itemsets with their counts in preceding runs are recorded for later use in maintenance. Assume there exist an original database and several newly inserted transactions. FUP divides the mining process into the following four cases (Table 1):

Table 1. Four cases and their FUP results

Cases: Original – New	Results
Case 1: Large – Large	Always large
Case 2: Large – Small	Determined from existing information
Case 3: Small – Large	Determined by rescanning original database
Case 4: Small – Small	Always small

FUP thus focuses on the newly inserted transactions and can save some processing time in rule maintenance. But FUP still has to scan an original database for managing Case 3 in which a candidate itemset is large in newly inserted transactions but is small in the original database. This situation may often occur when the number of newly inserted transactions is small. For example, suppose only one transaction is inserted into a database. In this situation, each itemset in the transaction is large. Case 3 thus needs to be processed in a more efficient way.

Hong *et. al.* thus proposed a mining algorithm based on pre-large itemsets to further reduce the need for rescanning original databases [13]. A pre-large itemset is not truly large, but promises to be large in the future. A lower support threshold and an upper support threshold are used to realize this concept. The upper support threshold is the same as that used in the conventional mining

algorithms. The support ratio of an itemset must be larger than the upper support threshold in order to be considered large. On the other hand, the lower support threshold defines the lowest support ratio for an itemset to be treated as pre-large. An itemset with its support ratio below the lower threshold is thought of as a small itemset. Pre-large itemsets act like buffers in the incremental mining process and are used to reduce the movements of itemsets directly from large to small and vice-versa.

Considering an original database and transactions newly inserted using the two support thresholds, itemsets may fall into one of the following nine cases illustrated in Fig.2.

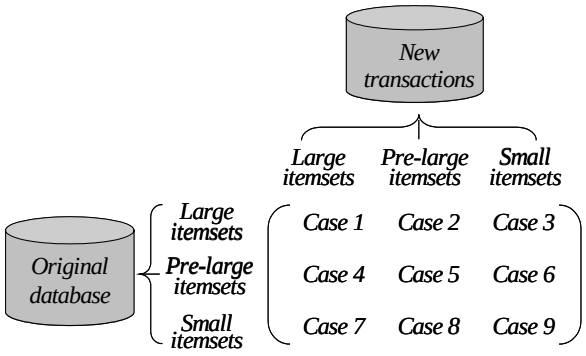

Fig.2: Nine cases arising from adding new transactions to existing databases

Cases 1, 5, 6, 8 and 9 above will not affect the final association rules according to the weighted average of the counts. Cases 2 and 3 may remove existing association rules, and cases 4 and 7 may add new association rules. If we retain all large and pre-large itemsets with their counts after each pass, then cases 2, 3 and 4 can be handled easily. Also, in the maintenance phase, the ratio of new transactions to old transactions is usually very small. This is more apparent when the database is growing larger. An itemset in case 7 cannot possibly be large for the entire updated database as long as the number of transactions is small when compared to the number of transactions in the original database. Let S_l and S_u be respectively the lower and the upper support thresholds, and let d and t be respectively the numbers of the original and new transactions. They showed that if

$$t \le \frac{(S_u - S_l)d}{1 - S_u}$$

then an itemset that is small (neither large nor pre-large) in the original database but is large in newly inserted transactions is not large for the entire updated database [13]. In this paper, we will generalize Hong et al's approach to maintain the association rules with taxonomy.

4 The Proposed Algorithm

The proposed maintenance algorithm integrates Hong et al's pre-large concepts and Srikant and Agrawal's mining method to find cross-level interesting association rules. Assume d is the number of transactions in the original database. A variable, c, is used to record the number of new transactions since the last re-scan of the original database. Details of the proposed mining algorithm are given below.

The maintenance algorithm for generalized association rules:

INPUT: A set of large and pre-large itemsets in the original database consisting of (d + c) transactions, a set of t new transactions, a predefined taxonomy, a lower support threshold S_l, an upper support threshold S_u, a predefined confidence value λ, and a predefined interest threshold α.

OUTPUT: A set of final generalized association rules for the updated database.

STEP 1: Calculate the safety number f of new transactions as follows:

$$f = \left\lfloor \frac{(S_u - S_l)d}{1 - S_u} \right\rfloor$$

STEP 2: Add ancestors of items appearing in the new transactions.

STEP 3: Set k = 1, where k records the number of items in itemsets.

STEP 4: Find all the candidate k-itemsets C_k and their counts in the new expanded transactions.

STEP 5: Divide the candidate k-itemsets into three parts according to whether they are large, pre-large or small in the original database.

STEP 6: For each itemset I in the originally large k-itemsets L_k^D, do the following substeps:

Substep 6-1: Set the new count $S^U(I) = S^T(I) + S^D(I)$.

Substep 6-2: If $S^U(I)/(d+t+c) \ge S_u$, then assign I as a large itemset, set $S^D(I) = S^U(I)$ and keep I with $S^D(I)$;
otherwise, if $S^U(I)/(d+t+c) \ge S_l$, then assign I as a pre-large

itemset, set $S^D(I) = S^U(I)$ and keep I with $S^D(I)$;
otherwise, neglect I.

STEP 7: For each itemset I in the originally pre-large itemset P_k^D, do the following substeps:

Substep 7-1: Set the new count $S^U(I) = S^T(I) + S^D(I)$.

Substep 7-2: If $S^U(I)/(d+t+c) \geq S_u$, then assign I as a large itemset, set $S^D(I) = S^U(I)$ and keep I with $S^D(I)$;
otherwise, if $S^U(I)/(d+t+c) \geq S_l$, then assign I as a pre-large itemset, set $S^D(I) = S^U(I)$ and keep I with $S^D(I)$;
otherwise, neglect I.

STEP 8: For each itemset I in the candidate itemsets that is not in the originally large itemsets L_k^D or pre-large itemsets P_k^D, do the following substeps:

Substep 8-1: If I is in the large itemsets L_k^T or pre-large itemsets P_k^T from the new expanded transactions, then put it in the rescan-set R, which is used when rescanning in STEP 9 is necessary.

Substep 8-2: If I is small for the new expanded transactions, then do nothing.

STEP 9: If $t + c \leq f$ or R is null, then do nothing; otherwise, rescan the original database to determine whether the itemsets in the rescan-set R are large or pre-large.

STEP 10: Form candidate $(k+1)$-itemsets C_{k+1} from finally large and pre-large k-itemsets ($L_k^U \bigcup P_k^U$)that appear in the new expanded transactions. Each 2-itemset in C_2 must not include items with ancestor or descendant relation in the taxonomy.

STEP 11: Set $k = k + 1$.

STEP 12: Repeat STEPs 3 to 11 until no new large or pre-large itemsets are found.

STEP 13: Discover the modified association rules according to the modified large itemsets by checking whether their confidence values are larger than or equal to the predefined minimum confidence.

STEP 14: Output the association rules which have no ancestor rules found.

STEP 15: For each remaining rule x, find the close ancestor rule y and calculate the support interest measure $I_{support}(x)$ of x as:

$$I_{support}(x) = \frac{count_x}{\frac{\Pi_{k=1}^{r+1} count_{x_k}}{\Pi_{k=1}^{r+1} count_{y_k}} \times count_y}$$

and the confidence interest measure $I_{confidence}(x)$ of x as:

$$I_{confidence}(x) = \frac{confidence_x}{\frac{count_{x_{r+1}}}{count_{y_{r+1}}} \times confidence_y}$$

where $confidence_x$ and $confidence_y$ are respectively the confidence values of rules x and y.

STEP 16: Output the rules with their support interest measure or confidence interest measure larger than or equal to the predefined interest threshold α as interesting rules.

TEP 17: If $t + c > f$, then set $d = d + t + c$ and set $c = 0$; otherwise, set $c = t + c$.

After Step 17, the final generalized association rules for the updated database have been determined.

5 An Example

An example is given below to illustrate the proposed maintenance algorithm for generalized association rules. Assume the original database includes 6 transactions as shown in Table 2.

Table 2. The original database in this example

TID	ITEMS
100	C
200	A, E
300	B, E
400	A, B, D
500	D
600	A

Each transaction includes a transaction ID and some purchased items. For example, the fourth transaction consists of three items: A, B and D. Assume the predefined taxonomy is as shown in Fig.3.

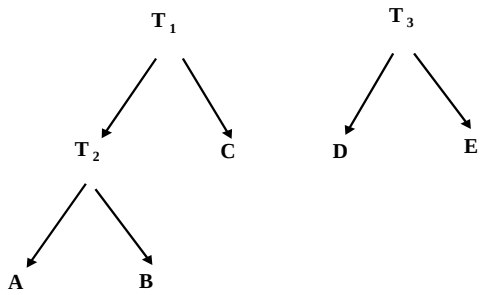

Fig.3 : The predefined taxonomy in this example

For S_l = 30% and S_u = 50%, the sets of large and pre-large itemsets for the given original transaction database are then kept for later maintenance. Assume now the two new transactions shown in Table 3 are inserted to the original database.

Table 3. Two new transactions

TID	Items
700	A, C
800	B, D

The proposed maintenance algorithm for generalized association rules proceeds as follows. The variable *c* is initially set at 0.

The safety number *f* for new transactions is calculated as:

$$f = \left\lfloor \frac{(Su - Sl)d}{1 - Su} \right\rfloor = \left\lfloor \frac{(0.5 - 0.3)6}{1 - 0.5} \right\rfloor = 2 .$$

The ancestors of items appearing in the new transactions are added. The new expanded transactions are thus shown in Table 4.

Table 4. The new expanded transactions

TID	Items
700	A, C, T_2, T_1
800	B, D, T_3, T_2, T_1

All candidate 1-itemsets C_1 and their counts from the new expanded transactions are found. All the candidate 1-itmesets are divided into three parts: {T_1}{T_2}{T_3}{A}, {B}{D}, and {C}, according to whether they are large, pre-large or small in the original database. STEPs 3 to 11 are repeated to find all large itemsets. Results are shown in Table 5.

Table 5. All the large itemsets for the updated database

1-itemset	2-itemset	3-itemset
{T_1}	{T_1, T_3}	None
{T_2}	{T_2, T_3}	
{T_3}		
{A}		

The association rules are then generated according to the modified large itemsets and the interest threshold.

6 Conclusion

In this paper, we have proposed an efficiently and effectively maintenance algorithm based on Srilant and Agrawal's approach to maintain generalized association rules with a taxonomy. It adopts the concept of pre-large itemsets to further reduce the need of rescanning original databases. The proposed algorithm does not require rescanning of the original databases until a number of new transactions have been processed. If the size of the database grows larger, then the number of new transactions allowed before rescanning will be larger too. This characteristic is especially useful for real-world applications.

Acknowledgment

The authors would like to thank the anonymous referees for their very constructive comments. This research was supported by MOE Program for Promoting Academic Excellence of Universities under the grant number 89-E-FA04-1-4

References:

[1] R. Agrawal, T. Imielinksi and A. Swami, "Mining association rules between sets of items in large database," *The ACM SIGMOD Conference*, pp. 207-216, Washington DC, USA, 1993.

[2] R. Agrawal, T. Imielinksi and A. Swami, "Database mining: a performance perspective," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 5, No. 6, pp. 914-925, 1993.

[3] R. Agrawal and R. Srikant, "Fast algorithm for mining association rules," *The International Conference on Very Large Data Bases*, pp. 487-499, 1994.

[4] R. Agrawal and R. Srikant, "Mining sequential patterns," *The Eleventh IEEE International*

Conference on Data Engineering, pp. 3-14, 1995.

[5] D.W. Cheung, J. Han, V.T. Ng, and C.Y. Wong, "Maintenance of discovered association rules in large databases: An incremental updating approach," T*he Twelfth IEEE International Conference on Data Engineering*, pp. 106-114, 1996.

[6] D.W. Cheung, V.T. Ng, and B.W. Tam, "Maintenance of discovered knowledge: a case in multi-level association rules," *The Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 307-310, 1996.

[7] D.W. Cheung, S.D. Lee, and B. Kao, "A general incremental technique for maintaining discovered association rules," *In Proceedings of Database Systems for Advanced Applications*, pp. 185-194, Melbourne, Australia, 1997.

[8] M. S. Chen, J. Han and P. S. Yu, "Data mining: an overview from a database perspective," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 6, 1996.

[9] A. Famili, W. M. Shen, R. Weber and E. Simoudis, "Data preprocessing and intelligent data analysis," *Intelligent Data Analysis*, Vol. 1, No. 1, 1997.

[10] W. J. Frawley, G. Piatetsky-Shapiro and C. J. Matheus, "Knowledge discovery in databases: an overview," *The AAAI Workshop on Knowledge Discovery in Databases*, 1991, pp. 1-27.

[11] T. Fukuda, Y. Morimoto, S. Morishita and T. Tokuyama, "Mining optimized association rules for numeric attributes," *The ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pp. 182-191, 1996.

[12] J. Han and Y. Fu, "Discovery of multiple-level association rules from large database," *The Twenty-first International Conference on Very Large Data Bases*, pp. 420-431, Zurich, Switzerland, 1995.

[13] T. P. Hong, C. Y. Wang and Y. H. Tao, "A new incremental data mining algorithm using pre-large itemsets," *Intelligent Data Analysis*, Vol. 5, No. 2, 2001, pp. 111-129.

[14] T. P. Hong, C. S. Kuo and S. C. Chi, "A data mining algorithm for transaction data with quantitative values," *Intelligent Data Analysis*, Vol. 3, No. 5, 1999, pp. 363-376.

[15] R. Agrawal, R. Srikant and Q. Vu, "Mining association rules with item constraints," *The Third International Conference on Knowledge Discovery in Databases and Data Mining*, pp. 67-73, Newport Beach, California, 1997.

[16] R. Srikant and R. Agrawal, "Mining generalized association rules," *The Twenty-first International Conference on Very Large Data Bases*, pp. 407-419, Zurich, Switzerland, 1995.