Maintenance of Generalized Association Rules for Record Deletion Based on the Pre-Large Concept

TZUNG-PEI HONG[†], TZU-JUNG HUANG[†]
[†]Department of Electrical Engineering
[‡]Library and Information Center
National University of Kaohsiung
No.700, Kaohsiung University Road, Kaohsiung 811, TAIWAN

Abstract: - In the past, we proposed an incremental mining algorithm for maintenance of generalized association rules as new transactions were inserted. Deletion of records in databases is, however, commonly seen in real-world applications. In this paper, we thus attempt to extend our previous approach to solve this issue. The proposed algorithm maintains generalized association rules based on the concept of pre-large itemsets for deleted data. The concept of pre-large itemsets is used to reduce the need for rescanning original databases and to save maintenance costs. The proposed algorithm doesn't need to rescan the original database until a number of records have been deleted. It can thus save much maintenance time.

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

1 Introduction

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][20]. Many algorithms for mining association rules from transactions were then 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.* 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 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 Srikant and Agrawal's mining approach to efficiently and effectively maintain generalized association rules for deleted data.

2 Mining Generalized Association Rules

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

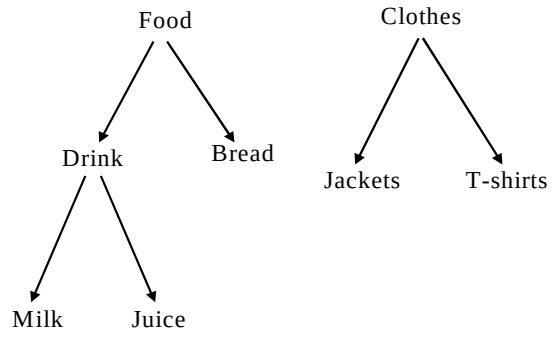

Fig. 1: An example of predefined taxonomic structures

Srikant and Agrawal proposed a method for finding generalized association rules on multiple levels [21]. 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 Rule Maintenance of Record Deletion

When records are deleted from databases, the original association rules may become invalid, or new implicitly valid rules may appear in the resulting updated databases. For example, assume a database has eight records as shown in Table 1 and assume the minimum support is 50%. The large 1-itemsets mined out from the data are {(A), (B), (C), (E)}. If two records *TID*=200 and *TID*=300 are deleted from Table 1, the originally small itemset (D) will become large.

Table 1: An original database with *TID* and *Items*

TID	*Items*
100	ACD
200	BCE
300	ABC
400	ABE
500	ABE
600	ACD
700	BCD
800	BCE

In the past, we proposed an incremental maintenance algorithm for record insertion under item taxonomies. Processing record deletion is, however, different from processing record insertion. In this paper, we use the concept of pre-large itemsets for processing record deletion under item taxonomies. Considering an original database and deleted records, the following nine cases (illustrated in Fig. 2) by the concept of pre-large itemsets may arise.

Cases 2, 3, 4, 7 and 8 above will not affect the final association rules. Case 1 may remove existing association rules, and cases 5, 6 and 9 may add new association rules. If we retain all large and pre-large itemsets, cases 5 and 6 can be handled easily. Also, in the maintenance phase, the ratio of deleted records to old transactions is usually very small. This is more apparent when the database is growing larger.

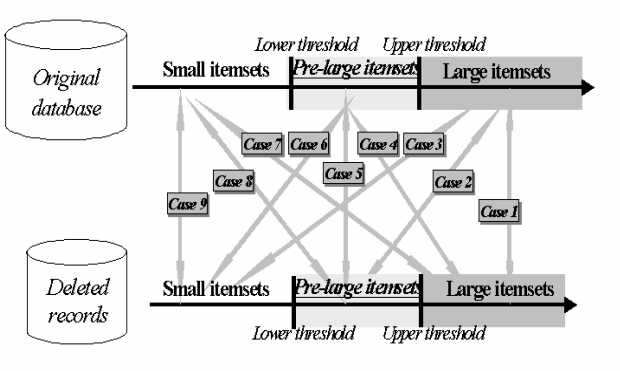

Fig. 2: Nine cases arise from deleting records for existing databases

Let S_l and S_u be respectively the lower and the upper support thresholds, d and t be respectively the numbers of the original and deleted records, and r denote the ratio of deleted records t to old transactions d. If $r \leq \frac{S_u - S_l}{S_u}$, then an itemset that is small (neither large nor pre-large) in both the original database and the deleted records is not large for the entire updated database. In this paper, we will generalize Hong *et al*'s pre-large concept to maintain the association rules with item taxonomies.

4 Notation

The notation used in this paper is defined below.

D : the original database;
T : the set of deleted records;
U : the entire updated database, i.e., D - T;
d : the number of transactions in D;
t : the number of records in T;
S_l *: the lower support threshold for pre-large itemsets;*
S_u *: the upper support threshold for large itemsets,* $S_u > S_l$*;*
L_k^D *: the set of large k-itemsets from D;*
L_k^U *: the set of large k-itemsets from U;*
P_k^D *: the set of pre-large k-itemsets from D;*
P_k^U *: the set of pre-large k-itemsets from U;*
C_k *: the set of all candidate k-itemsets;*
R_k^T *: the set of all k-itemsets in T which exist in* $(L_k^D \cup P_k^D)$*;*

I : an itemset;
$S^D(I)$: the count of I in D;
$S^T(I)$: the count of I in T;
$S^U(I)$: the count of I in U.

5 The Proposed Algorithm

Assume *d* is the number of transactions in the original database. A variable, *c*, is used to record the number of deleted 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 for record deletion:

INPUT: A set of large and pre-large itemsets in the original database consisting of $(d - c)$ transactions, a set of *t* deleted records, 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 deleted records 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 deleted records.

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

STEP 4: Find, from the deleted records, all the *k*-itemsets R_k^T and their counts that exist in the large itemsets L_k^D or in the pre-large itemsets P_k^D of the original database.

STEP 5: For each itemset *I* in the originally large itemset L_k^D, do the following substeps:

Substep 5-1: Set the new count $S^U(I) = S^D(I) - S^T(I)$.

Substep 5-2: If $S^U(I)/(d\text{-}t\text{-}c) \geq S_u$, then assign *I* as a large itemset, set $S^D(I) = S^U(I)$ and keep *I* with $S^D(I)$;
otherwise, if $S^U(I)/(d\text{-}t\text{-}c) \geq S_l$, 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 6: For each itemset *I* in the originally pre-large itemset P_k^D, do the following substeps:

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

Substep 6-2: If $S^U(I)/(d\text{-}t\text{-}c) \geq S_u$, then assign *I* as a large itemset, set $S^D(I) = S^U(I)$ and keep *I* with $S^D(I)$;
otherwise, if $S^U(I)/(d\text{-}t\text{-}c) \geq S_l$, 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 7: 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 7-1: If *I* is in the large itemsets L_k^T or pre-large itemsets P_k^T from the deleted expanded transactions, then do nothing.

Substep 7-2: If *I* is small for the deleted expanded transactions, then put it in the rescan-set *R*, which is used when rescanning in STEP 8 is necessary..

STEP 8: If $t+c \leq f$ or *R* is null, then do nothing; otherwise, rescan the original database to determine large or pre-large itemsets.

STEP 9: Form candidate *(k+1)*-itemsets C_{k+1} from finally large and pre-large *k*-itemsets ($L_k^U \bigcup P_k^U$). Each 2-itemset in C_2 must not include items with ancestor or descendant relation in the taxonomy.

STEP 10: Set $k = k+1$.

STEP 11: Repeat STEPs 3 to 10 until no new large or pre-large itemsets are found.

STEP 12: 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 13: Output the generalized association rules which have no ancestor rules found.

STEP 14: For each remaining rule *x*, find its close ancestor rule *y* and calculate the support interest measure $I_{support}(x)$ of *x* as:

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

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

$$I_{confidence}(x) = \frac{confidence_x}{\frac{count_{x_{r+1}}}{count_{y_{r+1}}} \times confidence_y} \quad (2)$$

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

STEP 15: 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.

STEP 16: If $t + c > f$, then set $d = d - t - c$ and set $c = 0$; otherwise, set $c = t + c$.

After Step 16, the final generalized association rules for the updated database have been determined.

6 An Example

An example is given below to illustrate the proposed maintenance algorithm for generalized association rules. Assume the original database includes 10 transactions as shown in Table 2.

Table 2. The original database in this example

TID	ITEMS
100	A
200	A, E
300	B, E
400	A, B, D
500	D
600	A, B
700	A, C, E
800	B, D
900	C, D, E
1000	A, D, E

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.

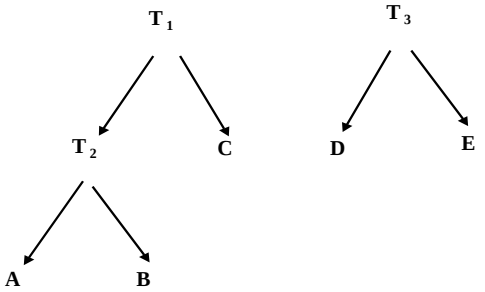

Fig. 3: The predefined taxonomy in this example

For S_l = 40% and S_u = 60%, the sets of large and pre-large itemsets for the given original transaction database are then kept for later maintenance. Assume now the two deleted transactions are shown in Table 3.

Table 3. Two deleted transactions

TID	Items
900	C, D, E
1000	A, D, E

The variable c is initially set at 0. The safety number f for new transactions is calculated as:

$$f = \left\lfloor \frac{(S_u - S_l)d}{1 - S_u} \right\rfloor = \left\lceil \frac{(0.6 - 0.4)10}{1 - 0.6} \right\rceil = 5.$$

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

Table 4. The new expanded transactions

TID	Items
900	C, D, E, T_1, T_3
1000	A, D, E, T_3, T_2, T_1

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

7 Conclusion

In this paper, we adopt Srikant and Agrawal's approach to maintain generalized association rules for deletion of records. The proposed algorithm can efficiently and effectively maintain association rules

with a taxonomy based on the pre-large concept and to further reduce the need for rescanning original databases. The proposed algorithm does not require rescanning of the original databases until a number of deleted records have been processed.

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] M. Y. Lin and S. Y. Lee, "Incremental update on sequential patterns in large databases," *The Tenth IEEE International Conference on Tools with Artificial Intelligence*, pp. 24-31, 1998.

[16] H. Mannila, H. Toivonen, and A. I. Verkamo, "Efficient algorithm for discovering association rules," *The AAAI Workshop on Knowledge Discovery in Databases*, pp. 181-192, 1994.

[17] J. S. Park, M. S. Chen, P. S. Yu, "Using a hash-based method with transaction trimming for mining association rules," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9, No. 5, pp. 812-825, 1997.

[18] W. Pedrycz, "Data mining and fuzzy modeling," *New Frontiers in Fuzzy Logic and Soft Computing Biennial Conference of the North American Fuzzy Information Processing Society*, 1996, pp. 263-267.

[19] N. L. Sarda and N. V. Srinivas, "An adaptive algorithm for incremental mining of association rules," *The Ninth International Workshop on Database and Expert Systems*, pp. 240-245, 1998.

[20] 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.

[21] 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.

[22] R. Srikant and R. Agrawal, "Mining quantitative association rules in large relational tables," *The 1996 ACM SIGMOD International Conference on Management of Data*, pp. 1-12, Montreal, Canada, 1996.

[23] S. Zhang, "Aggregation and maintenance for database mining," *Intelligent Data Analysis*, Vol. 3, No. 6, pp. 475-490, 1999.