ICEB 2004 Proceedings

International Conference on Electronic Business (ICEB)

Winter 12-5-2004

A Three-phased Online Association Rule Mining Approach for Diverse Mining Requests

Chingyao Wang

Shianshyong Tseng

Tzungpei Hong

Yianshu Chu

A Three-phased Online Association Rule Mining Approach for Diverse Mining Requests

Chingyao Wang[1], Shianshyong Tseng[1], Tzungpei Hong[2], Yianshu Chu[1]

[1] Institute of Computer and Information Science, National Chiao-Tung University, Hsinchu, 300, Taiwan, China
[2] Department of Electrical Engineering, National University of Kaohsiung, Kaohsiung, 811, Taiwan, China
tphong@nuk.edu.tw, {cywang, sstseng, yschu}@cis.nctu.edu.tw

ABSTRACT

In the past, most incremental mining and online mining algorithms considered finding the set of association rules or patterns consistent with the entire set of data inserted so far. Users can not easily obtain the results from their only interested portion of data. For providing ad-hoc, query-driven and online mining supports, we first propose a relation called *multidimensional pattern relation* to structurally and systematically store the context information and the mining information for later analysis. Each tuple in the relation comes from an inserted dataset in the database. This concept is similar to the construction of a data warehouse for OLAP. However, unlike the summarized information of fact attributes in a data warehouse, the mined patterns in the multidimensional pattern relation can not be directly aggregated to satisfy users' mining requests. We then develop an online mining approach called *Three-phased Online Association Rule Mining* (TOARM) based on the proposed multidimensional pattern relation to support online generation of association rules under multidimensional considerations. Experiments for both homogeneous and heterogeneous datasets are made, with results showing the effectiveness of the proposed approach.

Keywords: association rule, incremental mining, multidimensional mining, constraint-based mining, data warehouse

1. INTRODUCTION

Data mining technology has become increasingly important in the field of large databases and data warehouses. This technology helps discover non-trivial, implicit, previously unknown and potentially useful knowledge [3][9][16], thus being able to aid managers in making good decision. Among various types of databases and mined knowledge, mining association rules from transaction databases is the most interesting and popular. Previous works on mining association rules could be classified into batch mining approaches [2][4][5][7][20][22][24] and incremental mining approaches [10][11][13][18][23][25] according to the processing ways. Most of them have focused on finding association rules or patterns in a specified part of a database [15]. Some contexts (circumstance information) such as region, time and branch have usually been ignored in mining requests. Users can not easily obtain association rules or patterns from their only interested portion of data. However, decision-makers usually diversely consider problems at different aspects [14][15][16]. They may need to analyze market demands, customer preferences, localities, and short-term/long-term trends. They may also want to understand the change of discovered patterns or rules in different dimensions. This may decrease the usage of mining in online decision support for multidimensional data.

In this paper, we attempt to extend the concept of *effectively utilizing previously discovered patterns* in incremental data mining to support online generation of association rules under multidimensional considerations. We first propose the *multidimensional pattern relation* to structurally and systematically store the additional context information and mining information for each inserted dataset. It is conceptually similar to the construction of a data warehouse for OLAP [8][19][26]. Both of them preprocess the underlying data in advance, integrate related information, and store the results in a centralized structural repository for later use and analysis. However, unlike the summarized information of fact attributes in a data warehouse, the mined patterns in the multidimensional pattern relation can not be directly aggregated to satisfy users' mining requests. We then develop a *Three-phased Online Association Rule Mining* (TOARM) approach to effectively and efficiently satisfy diverse mining requests. It mainly consists of three phases, *generation of candidate itemsets*, *reduction of candidate itemsets*, and *generation of association rules*. The phase for *generation of candidate itemsets* selects the tuples satisfying the context constraints in a mining request and generates the candidate itemsets from the matched tuples. After that, the phase for *reduction of candidate itemsets* calculates the upper-bound supports of the candidate itemsets and adopts two pruning strategies to reduce the number of candidate itemsets. Finally, the phase for *generation of association rules* finds the final large itemsets and then derives the association rules from them. Experimental results also show the effectiveness of the proposed TOARM approach.

2. RELATED WORK

An association rule indicates a relationship among items such that the occurrence of certain items in a transaction would imply the occurrence of some other items in the same transaction. The process of mining association rules can roughly be decomposed into two tasks [4]: *finding large itemsets* and *generating interesting association rules*. The first task discovers the itemsets that satisfy a user-specified *minimum support* from a given database. It is used to obtain the statistically significant patterns. The second task finds the association rules that satisfy a user-specified *minimum confidence* from the large itemsets. Since this process is rather costly and time-consuming, some famous mining algorithms, such as Apriori [4], DIC [7], DHP [22], Partition [24], Sampling [20] and GSP [5], were proposed to achieve this purpose. Among them, the Apriori algorithm, which is the best well-known, utilizes a level-wise candidate generation approach to reduce its search space, such that only the large itemsets found in the previous level are treated as seeds for generating the candidate itemsets in the current level. This level-by-level property can greatly reduce the number of itemsets to be considered in a mining process. Many following algorithms were then based on this property and attempted to further reduce candidate itemsets and I/O costs. Comprehensive overviews can be referred to in [9][16].

Most of the mining algorithms process data in a batch way and must re-process the entire database whenever either the data stored in a database or the thresholds (i.e. the minimum support or the minimum confidence) set by users are changed. They do not utilize previously mined patterns for later maintenance, and may require considerable computation time to obtain the updated set of association rules or patterns [10]. Recently, some researchers have developed incremental mining algorithms to maintain association rules without re-processing the entire database whenever the database is updated. Examples include the FUP-based algorithms proposed by Cheung et al. [10][11], the adaptive algorithm proposed by Sarda and Srinivas [23], the incremental mining algorithm based on the concept of *pre-large itemsets* proposed by Hong et al. [18], and the incremental updating technique based on the concept of *negative border* proposed by Thomas et al. [25] and Feldman et al. [13]. The common idea of the above researches lies in that the previously mined patterns are stored in advance for later usage. When new transactions are inserted or old records are deleted, a large part of the final results can be obtained by comparing the patterns mined from the newly inserted transactions or deleted records with the pre-stored mined knowledge. Only a small portion of patterns needs to be re-processed against the entire database. Much computation time can thus be saved in this way. Among the above approaches, the FUP-based algorithms [10][11] store the previously mined large itemsets for later maintenance. Some other approaches utilize the *pre-large itemsets* [18] and the *negative border* [13][25] to enlarge the amount of pre-stored mined information for further improving the maintenance performance at the expense of storage spaces.

3. THE MULTIDIMENSIONAL PATTERN RELATION

A multidimensional pattern relation schema *MPR* is a special relation schema for storing mining information. An *MPR* consists of three types of attributes, *identification* (*ID*), *context*, and *content*. There is only one identification attribute for an *MPR*. It is used to uniquely label the tuples. Context attributes describe the contexts (circumstance information) of an individual block of data which are gathered together from a specific business viewpoint. Examples of context attributes are region, time and branch. Content attributes describe available mining information which is discovered from each individual block of data by a batch mining algorithm. Examples of content attributes include the number of transactions, the number of mined patterns, and the set of previously mined large itemsets with their supports.

The set of all previously mined patterns with their supports for an individual block of data is called a *pattern set* (*ps*) in this paper. Assume the minimum support is s and there are l large itemsets discovered from an individual block of data. A pattern set can be represented as $ps = \{(x_i, s_i) \mid s_i \geq s \text{ and } 1 \leq i \leq l\}$, where x_i is a large itemset and s_i is its support. The pattern set is thus a principal content attribute for an inserted block of data.

A multidimensional pattern relation schema *MPR* with n_1 context attributes and n_2 content attributes can be represented as $MPR(ID,\ CX_1,\ CX_2,\ \ldots,\ CX_{n_1},\ CN_1,\ CN_2,\ \ldots,\ CN_{n_2})$, where *ID* is an identification attribute, CX_i, $1 \leq i \leq n_1$, is a context attribute, and CN_i, $1 \leq i \leq n_2$, is a content attribute. Assume a multidimensional pattern relation *mpr*, which is an instance of the given *MPR*, includes tuples $\{t_1, t_2, \ldots, t_m\}$. Each tuple $t_i = (id_i,\ cx_{i1},\ cx_{i2},\ \ldots,\ cx_{in_1},\ cn_{i1},\ cn_{i2},\ \ldots,\ cn_{in_2})$ in *mpr* indicates that for the block of data under the contexts of cx_{i1}, cx_{i2}, …, and cx_{in_1}, the mining information contains cn_{i1}, cn_{i2}, …, and cn_{in_2}.

Example 1: Table 1 shows a multidimensional pattern relation with the initial minimum support set at 5%. *ID* is an identification attribute, *Region*, *Branch* and *Time* are context attributes, and *No_Trans*, *No_Patterns* and *Pattern_Sets* are content attributes. The *Pattern_Sets* attribute records the sets of mined large itemsets from the previous data blocks. For example, the tuple with *ID* = 1 shows that seven large itemsets, {(A, 10%), (B, 11%), (C, 9%), (AB, 8%), (AC, 7%), (BC, 6%), (ABC, 6%)}, are discovered from 10000 transactions and under the contexts of *Region* = **CA**, *Branch* = **San Francisco** and *Time* = **2003/10**. The other tuples have similar meaning. ■

Table 1: A multidimensional pattern relation with minimum support = 5%

ID	*Region*	*Branch*	*Time*	*No_ Trans.*	*No_ Patterns*	*Pattern_Sets (Itemset, Support)*
1	CA	San Francisco	2003/10	10000	7	(A,10%),(B,11%),(C,9%),(AB,8%),(AC,7%),(BC,6%),(ABC,6%)
2	CA	San Francisco	2003/11	15000	3	(A,5%),(B,7%),(C,5%)
3	CA	San Francisco	2003/12	12000	2	(A,5%),(C,9%)
4	CA	Los Angeles	2003/10	20000	4	(A,8%),(B,6%),(C,7%),(AC,6%)
5	CA	Los Angeles	2003/11	25000	2	(A,5%),(C,6%)
6	CA	Los Angeles	2003/12	30000	4	(A,6%),(B,6%),(C,9%),(AB,6%)
7	NY	New York	2003/10	18000	3	(B,8%),(C,7%),(BC,6%)
8	NY	New York	2003/11	18500	2	(B,8%),(C,6%)
9	NY	New York	2003/12	19000	5	(A,5%),(B,9%),(C,8%),(D,6%),(BC,6%)

4. MULTIDIMENSIONAL ONLINE MINING FOR ASSOCIATION RULES

The goal of online mining is to find the association rules satisfying the constraints in a mining request on line. The types of mining requests allowed can grow up through the usage of the proposed multidimensional pattern relation. In this paper, an online mining approach called *Three-phased Online Association Rule Mining* (TOARM) is proposed to achieve the mining task from a multidimensional pattern relation. TOARM first selects the tuples from the relation satisfying the constraints in a mining request. It then integrates and outputs the mining information in these tuples to users. Before describing the TOARM approach, we first formally define the problem to be solved and some related terminology. Some lemmas are also derived (The detailed proofs are omitted here).

Assume $mpr = \{t_1, t_2, \ldots, t_m\}$ is a multidimensional pattern relation based on an initial minimum support s. Given a mining request q with a set of contexts cx_q, a new minimum support s_q ($s_q \geq s$), and a new minimum confidence $conf_q$, the proposed algorithm will effectively and efficiently derive the association rules satisfying s_q, $conf_q$ and cx_q. A tuple with cx_q in a multidimensional pattern relation is called a *matched tuple*. Let t_i denote the i-th tuple in a multidimensional pattern relation, $t_i.trans$ denote the number of transactions kept in t_i, $t_i.ps$ denote the pattern set in t_i, and $t_i.s_x$ denote the actual support of an itemset x in t_i.

Lemma 1: For each itemset x satisfying s_q and cx_q in a mining request q, there exists at least a matched tuple t, such that $t.s_x$ satisfies s_q. ■

Lemma 2: For each itemset x satisfying s_q and cx_q in a mining request q, it must be among the candidate itemsets obtained by collecting the ones whose supports are larger than or equal to s_q in at least one matched tuple. ■

Lemma 3: If x is a candidate itemset, then $\forall x' \subset x$, x' is also a candidate itemset. ■

The *appearing count* $Count_x^{appearing}$ of a candidate itemset x is defined as the count of x calculated from the matched tuples in which x appears. Thus:

$$Count_x^{appearing} = \sum_{t_i \in matched\ tuples\ \&\ x \in t_i.ps} t_i.trans * t_i.s_x. \quad (1)$$

The *upper-bound count* $Count_{\bar{x}}^{UB}$ of a candidate itemset x is defined as the upper bound count of x calculated from the matched tuples in which x does not appear. Thus:

$$Count_{\bar{x}}^{UB} = \sum_{t_i \in matched\ tuples\ \&\ x \notin t_i.ps} (t_i.trans * s - 1) \cdot \quad (2)$$

Let *Match_Trans* denote the number of transactions in the matched tuples. Thus:

$$Match\_Trans = \sum_{t_i \in matched\ tuples} t_i.trans \cdot \quad (3)$$

The *upper-bound support* s_x^{UB} of a candidate itemset x is thus calculated as:

$$s_x^{UB} = \frac{Count_x^{appearing} + Count_{\bar{x}}^{UB}}{Match\_Trans}. \quad (4)$$

Lemma 4: If x is a candidate itemset and s_x is its actual support, then $s_x \leq s_x^{UB}$. ■

Lemma 5: If x is a candidate itemset, then $\forall x' \subset x$, $s_{x'}^{UB} \geq s_x^{UB}$. ■

Lemma 6: If a candidate itemset x is contained in all the matched tuples, then $s_x^{UB} = s_x$. ■

The Three-phased Online Association Rule Mining (TOARM) approach:

INPUT: A multidimensional pattern relation based on an initial minimum support s and a mining request q with a set of contexts cx_q, a minimum support s_q and a minimum confidence $conf_q$.

OUTPUT: A set of association rules satisfying the mining request q.

***Phase 1*: *Generation of candidate itemsets*:**

(a) Select the tuples satisfying cx_q from the multidimensional pattern relation.

(b) Gather the candidate itemsets appearing in the matched tuples.

(c) Calculate $Count_x^{appearing}$ and $Count_{\bar{x}}^{UB}$ for each candidate itemset x.

Phase 2*: *Reduction of candidate itemsets:

(a) Calculate the upper-bound support s_x^{UB} of each candidate itemset x by the formula:

$$s_x^{UB} = \frac{Count_x^{appearing} + Count_{\bar{x}}^{UB}}{Match\_Trans}.$$

(b) Discard the candidate itemset x and its proper supersets from the candidate set if $s_x^{UB} \leq s_q$.

(c) Put x into the set of large itemsets if $s_x^{UB} = \frac{Count_x^{appearing}}{Match\_Trans}$ and $s_x^{UB} \geq s_q$.

Phase 3*: *Generation of association rules:

(a) Check whether each remaining candidate itemset x is large by scanning the underlying blocks of data for the matched tuples in which x does not appear.

(b) Generate the association rules satisfying the minimum confidence $conf_q$ from the set of large itemsets.

The TOARM approach only considers the itemsets appearing in the matched tuples and satisfying the minimum support as the candidate ones. It also uses two pruning strategies to reduce the number of candidate itemsets. It therefore only needs to re-process the remaining candidate itemsets against the underlying blocks of data for the matched tuples in which they do not appear. Due to the above consideration, the cost of re-processing underlying blocks of data by the TOARM approach is less than that by typical batch mining or incremental mining approaches.

Example 2: For the multidimensional pattern relation given in Table 1, assume a mining request q is to get the patterns under the contexts cx_q of *Region* = **CA** and *Time* = **2003/11~2003/12** and satisfying the minimum support s_q = 5.5%. According to Lemma 2, the set of candidate itemsets is {{A}, {B}, {C}, {AB}}, which is the union of the itemsets appearing in the pattern sets and with their supports larger than 5.5%. Among these candidate itemsets, in Phase 2, the TOARM approach can remove the candidate itemsets {A} and {AB} according to Lemmas 4 and 5, and put the candidate itemset {C} into the set of large itemsets for q according to Lemma 6. Only the remaining candidate itemset {B} needs to be further processed in Phase 3.

6. EXPERIMENTS

The experiments were implemented in Java on a workstation with dual XEON 2.8GHz processors and 2048MB main memory, running RedHat 9.0 operation system. The datasets were generated by a generator similar to that used in [4]. The generator first generated L maximal potentially large itemsets, each with an average size of I items. The items in a potentially large itemset were randomly chosen from the total N items according to its actual size. The generator then generated D transactions, each with an average size of T items. The items in a transaction were generated according to the L maximal potentially large itemsets in a probabilistic way.

The two groups of datasets generated in the above way and used in our experiments are listed in Table 2, where the datasets in the same group had the same D, T and I values but different L or N values. Each dataset was treated as a block of data in the database. Among the two groups, Group 2 could be thought of as heterogeneous because of its varied N values. This group of datasets was used to show the effect of heterogeneous blocks of data on our approach.

Table 2: The two groups of datasets generated for the experiments

Group	*Size*	*Datasets*	*D*	*T*	*I*	*L*	*N*
1	10	*T10I8D10KL*[1] *to* *T10I8D10KL*[10]	10000	10	8	200 *to* 245	100
2	10	*T10I8D10KN*[1] *to* *T10I8D10KN*[10]	10000	10	8	200	100 *to* 145

The TOARM and the Apriori algorithms were then run for Groups 1 and 2 along with different minimum supports ranging from 0.022 to 0.04 in the mining requests. The execution times spent by the two algorithms for each group are respectively shown in Figures 1 and 2. From Figures 1, it is easily seen that the execution time by the TOARM algorithm on Groups 1 was always much less than that by the Apriori algorithm. This is because the datasets in this group was homogeneous, meaning they used the same set of items in each group. In this situation, the number of candidate itemsets considered by the TOARM algorithm was much closer to the number of the final large itemsets than that by the Apriori algorithm. The former thus had a more compact candidate set than the latter.

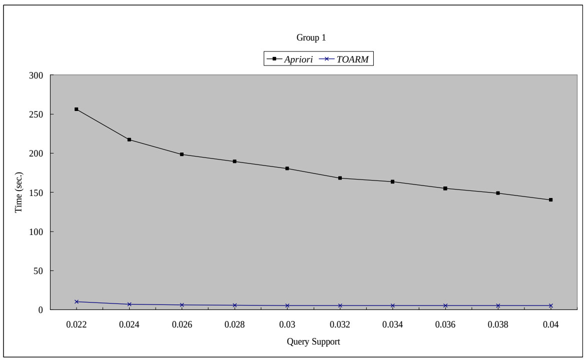

Figure 1: The execution time spent by the two algorithms for Group 1

On the contrary, the datasets in Group 2 were heterogeneous, meaning they used different sets of items. In this situation, the number of candidate itemsets considered by the TOARM algorithm was much larger than the number of the final large itemsets since most of the candidate itemsets appeared in only one or few tuples in the multidimensional pattern relation. But, since the TOARM algorithm adopted two pruning strategies in Phase 2 and only re-processed the remaining candidate itemsets in Phase 3 against the underlying datasets in which they do not appear, the execution time spent by the TOARM algorithm was usually still less than that spent by the Apriori algorithm. This is also consistent with the results shown in Figure 2.

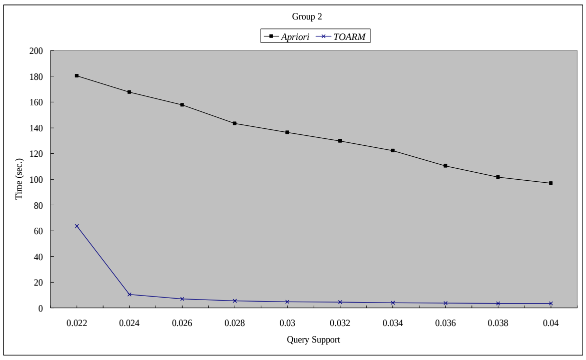

Figure 2: The execution time spent by the two algorithms for Group 2

7. CONCLUSION

In this paper, we have extended the concept of effectively utilizing previously discovered patterns in incremental mining to online decision support under multidimensional considerations. By structurally and systematically storing the additional context information and mining information in the multidimensional pattern relation, our proposed TOARM approach can easily and efficiently derive the association rules satisfying diverse user-concerned constraints. From the experimental results, the proposed TOARM approach is more efficient than the well-known Apriori approach especially for homogeneous datasets.

ACKNOWLEDGEMENT

This research was supported by the National Science Council of Taiwan, China under Grand No. NSC93-2752-E-009-006-PAE.

REFERENCES

[1] C. C. Aggarwal and P. S. Yu, “A New Approach to Online Generation of Association Rules,” IEEE Transactions on Knowledge and Data Engineering, Vol. 13, No. 4, pp. 527-540, 2001.

[2] R. Agrawal, T. Imielinksi and A. Swami, “Mining Association Rules between Sets of Items in Large Database,“ ACM SIGMOD Conference, pp. 207-216, Washington DC, USA, 1993.

[3] 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.

[4] R. Agrawal and R. Srikant, “Fast Algorithm for Mining Association Rules,” ACM International Conference on Very Large Data Bases, pp. 487-499, 1994.

[5] R. Agrawal and R. Srikant, ”Mining Sequential Patterns,” IEEE International Conference on Data Engineering, pp. 3-14, 1995.

[6] K. Beyer and R. Ramakrishnan, “Bottom-Up Computation of Sparse and Iceberg CUBEs,” ACM SIGMOD Conference, pp. 359-370, 1999.

[7] S. Brin, R. Motwani, J. D. Ullman and S. Tsur, “Dynamic Itemset Counting and Implication Rules for Market Basket Data,” ACM SIGMOD Conference, pp. 255-264, Tucson, Arizona, USA, 1997.

[8] S. Chaudhuri and U. Dayal, “An Overview of Data Warehousing and OLAP Technology,” ACM SIGMOD Record, 26:65-74, 1997.

[9] 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, pp. 866-883, 1996.

[10] D. W. Cheung, J. Han, V. T. Ng and C. Y. Wong, “Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Approach,” IEEE International Conference on Data Engineering, pp. 106-114, 1996.

[11] 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.

[12] M. Fang, N. Shivakumar, H, Garcia-Molina, R. Motwani, J. D. Ullman, “Computing Iceberg Queries Efficiently,” ACM International Conference on Very Large Data Bases, pp. 299-310, 1998.

[13] R. Feldman, Y. Aumann, A. Amir, and H. Mannila, “Efficient Algorithms for Discovering Frequent Sets in Incremental Databases,” ACM SIGMOD Workshop on DMKD, pp. 59-66, USA, 1997.

[14] G. Grahne, L. V. S. Lakshmanan, X. Wang and M. H. Xie, “On Dual Mining: From Patterns to Circumstances, and Back,” IEEE International Conference on Data Engineering, pp. 195-204, 2001.

[15] J. Han, L. V. S. Lakshmanan and R. Ng, “ Constraint-based, Multidimensional Data Mining,” IEEE Computer Magazine, pp.2-6, 1999.

[16] J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann Publishers, 2001.

[17] C. Hidber, “Online Association Rule Mining,” ACM SIGMOD Conference, pp. 145-156, USA, 1999.

[18] T. P. Hong, C. Y. Wang and Y. H. Tao, “A New Incremental Data Mining Algorithm Using Pre-large Itemsets,” International Journal on Intelligent Data Analysis, 2001.

[19] W. H. Immon, Building the Data Warehouse, Wiley Computer Publishing, 1996.

[20] 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.

[21] H. Mannila and H. Toivonen, “On an Algorithm for Finding all Interesting Sentences,” The European Meeting on Cybernetics and Systems Research, Vol. II, 1996.

[22] J. S. Park, M. S. Chen and 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.

[23] N. L. Sarda and N. V. Srinivas, “An Adaptive Algorithm for Incremental Mining of Association Rules,” IEEE International Workshop on Database and Expert Systems, pp. 240-245, 1998.

[24] A. Savasere, E. Omiecinski and S. Navathe, “An Efficient Algorithm for Mining Association Rules in Large Database,” ACM International Conference on Very Large Data Bases, pp. 432-444, 1995.

[25] S. Thomas, S. Bodagala, K. Alsabti and S. Ranka “An Efficient Algorithm for the Incremental Update of Association Rules in Large Databases,” The International Conference on Knowledge Discovery and Data Mining, pp. 263-266, 1997.

[26] J. Widom, "Research Problems in Data Warehousing," ACM International Conference on Information and Knowledge Management, 1995.