ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

REVIEW ARTICLE

Available Online at www.ijarcs.info

XML Data Mining using XQuery and Improved Apriori Algorithm

Tabassum N. Mujawar*
Computer Engineering Department
Terna Engineering College
Nerul Navi Mumbai, India
tabsmaktum@gmail.com

Subhash K.Shinde
Computer Engineering Department
Lokmanya Tilak College of Engineering
Koparkhirne, Navi Mumbai, India
skshinde@rediffmail.com

Varunakshi Bhojane
Computer Engineering Department
Pillai's Institute of Information Technology,
New Panvel, Navi Mumbai, India
varunakshi_k@yahoo.co.in

Abstract: The growing usage of XML technology for data storage has naturally resulted in the pressing need for more suitable tools and techniques to perform mining on XML data. One of the simple methods to mine XML data is probably to transform the data from XML to relations. However, the drawback of this method is: the transformation itself is usually complex and time-consuming. Therefore an alternative approach which can mine XML data without any preprocessing and postprocessing is required. XQuery which is query language for XML satisfies the demand for intelligently querying XML data and hence can be used to mine XML data .The paper presents a framework for mining XML data using XQuery and Association Rule Mining Technique. To improve the shortcomings of classic Apriori algorithm, an improved Apriori algorithm with itemset pruning and transactions trimming is put forward.

Keywords: XML, XQuery, Apriori, association rule mining, transaction reduction, items pruning.

I. INTRODUCTION

Most recently, Extensible Markup Language (XML), proposed by the World Wide Web Consortium has become a widely accepted standard for exchanging information between various applications. This results in huge amount of data available in XML form. Following the increasing use of XML technology for data storage and data exchange between applications, the subject of mining XML data has become important research topic. The process of knowledge discovery for well-structured data such as relational databases and object-oriented databases has gain more attention and provides successful performance, but the world of mining semi-structured data such as XML still remains at a preliminary stage. As, XML is inherently flexible, in both structure and semantics, the topic of mining XML data is confronted with more challenges. The aim of XML mining is to integrate the emerging XML technology into data mining technology. Thus more suitable tools and techniques for XML mining are starting to emerge.

To discover knowledge, it is necessary to intelligently query XML documents and extract contents from XML tags to prepare data for mining. This demand is satisfied by XQuery [1] which is standard query language, defined by W3C, for extracting contents from XML documents. XQuery for XML is like SQL for relational databases. Also, XQuery is supported by all the major database engines like IBM, Oracle, and Microsoft, etc.Thus XQuery provides a simple and flexible way to process XML data in its original form without any conversion. Therefore, XQuery can be used to mine XML data without any preprocessing and postprocessing.

Association rule mining is one of the widely used and well researched data mining techniques which find associations between items in a database. Mining association rules from huge amount of data can help in many business decision making processes. Association rule mining has a wide range of applicability. One of the important applications is consumer market analysis; this will analyze the buying behaviour of customers for determining items that customers buy together. Based on this analysis, associations rules of the form "Customer who bought item A also bought item B ", can be extracted. Such rules will help to make decisions about discounts, which products to be put next to each other, how to improve profits etc. This paper presents an approach of mining XML data using XQuery and well known association rule mining technique.

The outline of this paper is as follows: Section 2 presents the relate work, Section 3 describes the basic Apriori algorithm and its limitations. An improved Apriori algorithm is presented in Section 4 and experimental results are shown in section 5.Finally paper concludes in section 6.

II. RELATED WORK

Several techniques for mining XML data have been proposed. Recently a number of researchers have developed algorithms able to detect frequently occurring substructures from structure of XML data. These include TreeFinder by Termier [2] and TreeMiner by Zaki [3] which extract the frequent tree patterns from the structure of XML data. The XMINE operator [4] has been introduced for extracting association rules from XML documents, where mapping the XML data to a relational structure is required before mining is performed. However, the recent works reported that the association rules can be mined directly from XML data without converting it into relational form. Thus, an approach

for mining association rules from XML data using XQuery without conversion has been presented by Jacky W. W. Wan.[5].Another approach is to use programs written in high level programming language such as Java, for mining association rules from XML data [6]. However XQuery implementation of mining association rules from XML data has some limitations. Therefore, an improved framework for mining XML data using XQuery and .NET based implementation of Apriori algorithm has been proposed [7].

III. APRIORI ALGORITHM

A. Basic Algorithm:

One of the first algorithms to evolve for frequent itemset generation and Association rule mining was Apriori[8].Apriori is used to find all frequent itemsets in a given database by making multiple passes over database. Each pass of Apriori algorithm involve three steps, for generation of frequent itemsets: the scan step, the join step and prune step.

In the first pass the algorithm simply counts item occurrences to determine the frequent 1-itemsets. These frequent 1-itemsets become seed for the next pass. In each subsequent pass this seed set is used to calculate candidate itemset by joining previous level frequent itemsets with itself. Then the supports of candidate itemsets during the pass over the data are found and the next level frequent itemsets are generated by eliminating those candidates whose support value is less than user-specified minimum support. Then the prune step helps in filtering out candidate item-sets whose subsets (prior level) are not frequent. This is based on the anti-monotonic property as a result of which every subset of a frequent itemset is also frequent. Thus a candidate itemset which is composed of one or more infrequent item sets of a prior level is filtered (pruned) from the process of frequent itemset generation. The algorithm terminates when the set of all frequent itemsets becomes empty. Finally the association rules can be discovered based on these frequent itemsets.

B. Limitations:

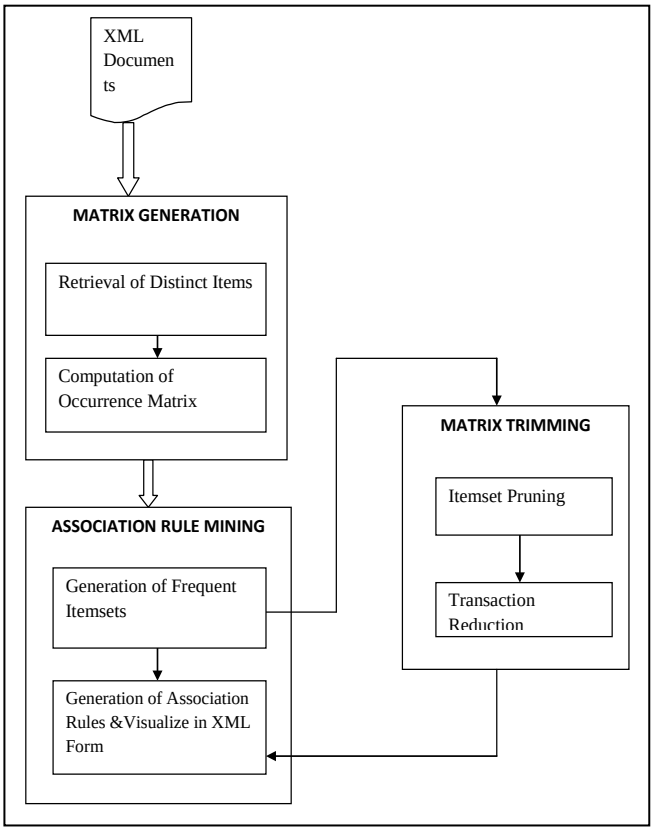
The two major issues which affect the performance of Apriori algorithm are as follows:

Frequent itemset generation is the most important task of association rule mining process and it is crucial to performance. The candidate itemsets generated during early iteration is generally, in orders of magnitude, larger than the frequent itemsets it really contains. Thus the processing in initial iterations in fact dominates the total execution cost .Therefore the initial candidate itemset generation, especially for the large 2-itemsets is the key issue to improve the performance of algorithm.

Another performance related issue is, requirement to scan database many times during frequent itemset discovery. A straightforward implementation would require one pass over the database of all transactions for all iterations. Thus to improve efficiency of algorithm, the number of database scans must be reduced and the number of items in each transactions must be trimmed.

IV. IMPROVED ALGORITHM

In order to overcome limitations of classic Apriori algorithm this section presents the improved approach to mine association rules from XML data. The improvement technique used is the combination of "Transaction reduction and Pruning Optimization" strategy [9] and "Boolean matrix based Apriori algorithm" [10]. Consequently, the main aim is to optimize the process of finding association rules from XML data which should be more efficient and scalable. The framework used for generating association rules from XML data is presented in fig.1.



The framework consists of following phases:

A. Boolean matrix Generation:

During this phase the actual XML transaction data files are stored into transaction data base, in which each XML transaction data file is uniquely identified with transaction identifiers. The XML document structure is given in Fig. 2.

```
<?xml version=1.0 encoding="utf-8"?>
<transactions>
        <transaction>
                <items>
                        <item> I1 </item>
                        <item> I2 </item>
                        <item> I5 </item>
                </items>
        </transaction>
        <transaction>
                <items>
                        <item> I2 </item>
                        <item> I3 </item>
                </items>
        </transaction>
        ....................

</transactions>
```

Figure 2. Input XML Transaction File

Then XQuery expression is used to find distinct items from each transaction file, which will form the candidate-1

itemset (C1).These candidate itemsets become seed to next step of generating frequent itemsets. The next thing is to generate initial occurrence matrix. This Boolean matrix is a structure such that: Columns represent items found during the first traversal of transactional database, and rows represent various transactions. If item occurs in the transaction the corresponding columns is encoded as 1, otherwise encoded as 0 .Those items whose support value is greater than minimum support will be included in frequent-1 itemset (L1) and these itemsets will become seed to generate next level candidate and frequent itemsets. In this way, the transaction database will be scanned only once while creating the initial matrix.

B. Pruning The Boolean Matrix:

As number of transactions will increase the size of matrix will become large so the matrix is pruned that means deleting some rows and columns from it by applying following two strategies.

a. ***Itemset Pruning:*** When a K-1-dimensional frequent itemsets (L_{k-1}) are calculated, a temporary table is generated. This table consist of all K-dimensional subsets of all members of L_{k-1} and their frequency of occurrence in the L_{k-1}.If the frequency of any item I_j, is less than K-1 then this item is not used to generate the k-dimensional frequent itemsets (L_k). Hence we can delete all the frequent itemsets which contain such an item I_j.The matrix is pruned by deleting the column corresponding to the item I_j. By applying this strategy the scale of L_{k-1} is effectively reduced and generation of K-dimensional candidate itemset (C_k) becomes more efficient. Also the size of matrix is reduced as columns corresponding to the non frequent itemsets are removed..

b. ***Transaction Reduction:*** According to transaction reduction strategy, while scanning the database to generate K+1-dimensional frequent itemsets (L_{k+1}), calculate the length of all transactions and delete the transaction whose length is less than K+1 directly.

Thus, to prune the Boolean matrix, the sum of elements in each row is calculated and the row of matrix is deleted directly if the sum is less than K+1. By applying this transaction reduction strategy the scale of transaction database is reduced constantly and scanning of unnecessary transactions is avoided. In this way for every subsequent pass the unnecessary rows and columns are deleted from matrix and size of matrix becomes smaller. So it will help to improve the overall performance of algorithm.

C. *Association Rule Mining Phase:*

This phase consists of generation of all frequent itemsets and generation of association rules from these frequent itemsets which satisfies minimum confidence. The candidate-1 itemset (C1) and frequent-1 itemset (L1) are calculated in previous phases. To find next level candidate itemsets the join step of algorithm is applied i.e. to find C_k the frequent k-1 itemset (L_{k-1}) is joined by itself. The join $L_{k-1} \bowtie L_{k-1}$ is performed. Thus current level candidates are found by combining previous level frequent itemsets. From the C_k, the itemsets whose support value is less than minimum support are removed and frequent-k itemset (L_k) is generated. Similarly the next level candidate and frequent itemsets are calculated. Finally, from these frequent itemsets association rules which satisfy minimum confidence are formulated and represented in XML format as shown in Fig. 3.

```
<?xml version=1.0  encoding="utf-8"?>

<Association Rules>
        <rule>
        <items> I1  I2----->  I3 </items>
        <confidence>  0.76 </confidence>
        </rule>
        <rule>
        <items> I2  I3----->  I5 </items>
        <confidence>  0.5 </confidence>
        </rule>
        . . . . . .
        . . . . . .
</Association Rules>
```

Figure 3. Association Rules in XML Form

D. Example:

This section describes a sample execution of the algorithm. The XML transaction data files are stored into transaction data base, in which each XML transaction data file is uniquely identified with transaction identifiers. The distinct items from each transaction file are found and occurrence matrix is computed using XQuery, based on occurrence of items with respect to the total items as shown in Fig. 4.The minimum support value considered for the example is 2. The further procedure of generating frequent itemsets and association rules is shown in Fig. 5.

To apply Itemset pruning strategy a temporary table T1 is computed which will consist of frequencies of items all in L2'.From table T1 it is observed that frequency of item 1 and 4 is less than 2 so all itemsets from L2' containing items 1 and 4 will be deleted. Thus we have reduced the size of L2.To trim the matrix columns corresponding to items 1 and 4 are deleted from initial matrix and the new matrix will be as given below:

	2	3	5
T1	0	0	0
T2	1	1	1
T3	1	1	1
T4	1	0	0
T5	1	1	0

Now, to apply transaction reduction strategy, from this modified matrix rows having length less than 3 are deleted. So for calculation of L3 this trimmed matrix is used.

Transaction database

TID	Transaction.xml
T1	<transaction> <items> <item> 1 </item> <item> 4 </item> </items> </transaction>
T2	<transaction> <items> <item> 2 </item> <item> 3 </item> <item> 4 </item> <item> 5 </item> </items> </transaction>
T3	<transaction> <items> <item> 1 </item> <item> 2 </item> <item> 3 </item> <item> 5 </item> </items> </transaction>
T4	<transaction> <items> <item> 2 </item> <item> 4 </item> </items> </transaction>
T5	<transaction> <items> <item> 1 </item> <item> 2 </item> <item> 3 </item> </items> </transaction>

Initial Occurrence Matrix

	1	2	3	4	5
T1	1	0	0	1	0
T2	0	1	1	1	1
T3	1	1	1	0	1
T4	0	1	0	1	0
T5	1	1	1	0	0

Figure 4. XML Transaction Files and Initial Occurrence matrix

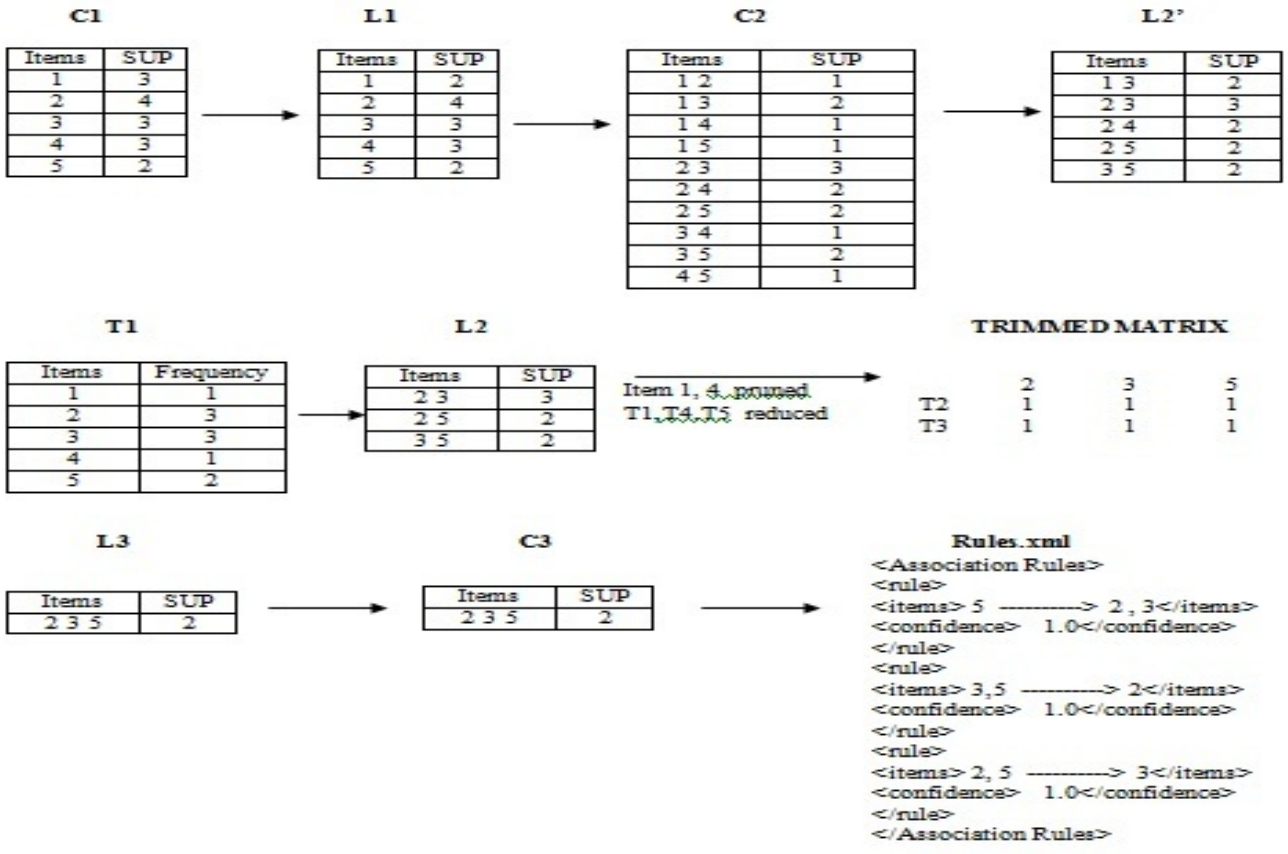


Figure 5. Complete Execution of Algorithm

Thus we can observe form fig.5 that transactions T1, T4 and T5 are with length less than 3 so we can delete the corresponding rows from the matrix. Thus we have reduced the scale of transaction database and size of occurrence matrix.

V. PERFORMANCE ANALYSIS

In order to appraise the performance of the improved matrix based Apriori algorithm with pruning and trimming strategy (MTPAPRIORI), experiments are conducted using the matrix based Apriori (MAPRIORI) algorithm and the MPTAPRIORI algorithm. The algorithms are tested on three synthetic data sets which will consist of supermarket customers tansactions.The datasets are generated randomly which have 100,500 and 1000 transactions. Maximum transaction size for each dataset is 20 and total distinct items are 50. The synthetic datasets are generated using some pre-defined tables which hold information about number of items in each transaction and the actual transactions. All the dataset creation programs are source coded in Java using the Collections Framework. The database used to store XML transaction data in its native form is DB2 9.1[11], which provides support for 'XML' data type. The algorithms are implemented using Java and DataDirectXquery, which is the fastest, most reliable, and most scalable XQuery engine installed on an Intel Core i3, 2.53GHz system running Windows 7 with RAM-3G.

The minimum confidence value considered is 0.5.The comparison of number of frequent itemsets generated for different support values for Dataset3 with 1000 transactions is shown in Fig. 6. From the graph we can observe that the number of frequent itemsets generated by MTPARIORI algorithm is less than that of the MAPRIORI algorithm.

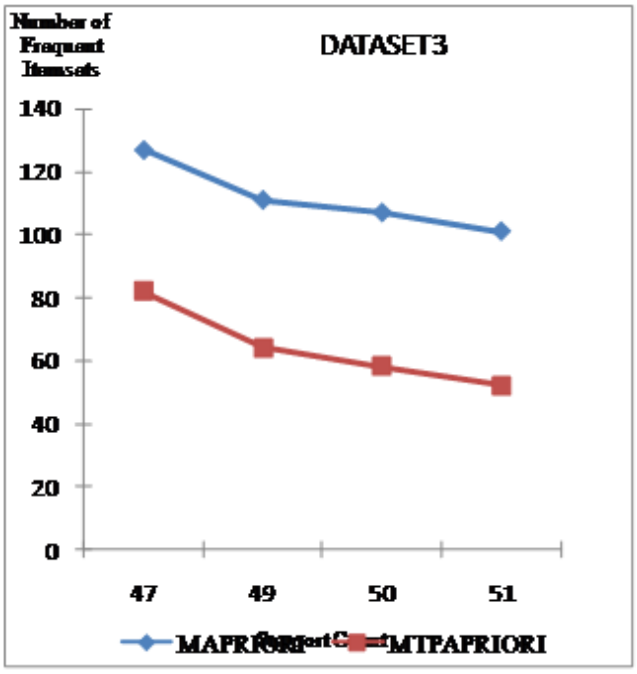


Figure 6. Comparison of Number of Frequent Itemsets for Dataset 3

The comparison of the running time of two algorithms for Dataset 3 is shown in fig7.

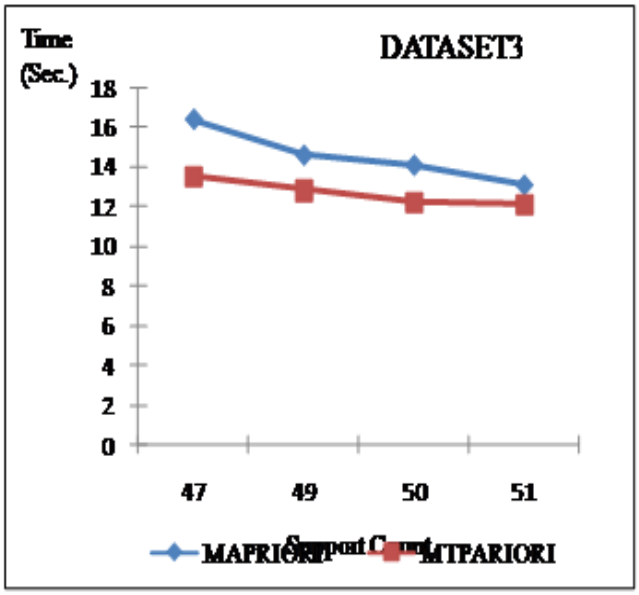


Figure 7. Comparison of Running Time for Dataset 3

The graph which takes Time Vs Support, describes how time factor is varying based on the support count of items. According to the graph, it can be seen that the running time of the improved Apriori with pruning and trimming strategy (MTPAPRIORI) is less than the running time of the pure matrix based Apriori (MAPRIORI) under different support counts.

VI. CONCLUSION

This paper presents an approach for mining association rules from XML data using XQuery and Apriori algorithm without any preprocessing and postprocessing, A matrix based Apriori algorithm (MAPRIORI) and an improved matrix based Apriori algorithm with pruning optimization and transaction reduction strategy (MTPAPRIORI) are implemented. The results show that the database is scanned only once because of matrix based approach used for both of these algorithms. If we compare the results obtained from the two algorithms it is observed that the number of frequent itemsets generated by MTPAPRIORI algorithm and the running time for this algorithm is less than the MAPRIORI algorithm for different support levels. Also the algorithm reduces the scale of the transaction database to be scanned. Thus the improved algorithm decreases the system overhead and improves the overall efficiency.

VII. REFERENCES

[1] World Wide Web Consortium, XQuery 1.0: An XML Query Language ,Available at http://www.w3.org/TR/xquery,2007.

[2] Termier, A., Rousset, M. and Sebag, M, " Treefinder: A First Step towards XML Data Mining,"Proc. IEEE International Conference on Data.Mining, 2002 ,pp.450-458

[3] Zaki, M. and Aggarwal, C., "XRULES: An Effective Structural Classifier for XML Data," Proc. 9th International Conference on Knowledge Discovery and Data Mining, 2003, pp. 137-170.

[4] Braga, A. Campi, M. Klemettinen, and P. L. Lanzi ,"Mining association rules from xml data," Proc. 4th International Conference on Data Warehousing and Knowledge Discovery , Aixen-Provence,France, 2002.

[5] Jacky W.W. Wan Gillian Dobbie ,"Mining Association Rules from XML Data using XQuery," Proc. second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation,2004, Volume 32,pp. 169-172.

[6] Qin Ding and Gnanasekaran Sundarraj,"Association Rule Mining from XML data ," Proc. International Conference on Data Mining, Las Vegas, Nevada, USA 2008,,pp. 144-151.

[7] R. Porkodi,V.Bhuvaneshwari, R. rajesh, t. Amudha, "An Improved Association Mining Technique for XML Data using XQuery and Apriori Algorithm,"Proc. IEEE International Advanced Computing Conference Patiala,India , 2009,pp. 1510-1514.

[8] R. Agrawal, and R. Srikant, "Fast Algorithms for Mining Association Rules," Proc. 20th International Conference on Very Large Databases, Santiago, Chile, 1994.

[9] Zhuang ChenShibang Cai, Qiulin Song and Chonglai Zhu , "An Improved Apriori Algorithm Based on Pruning Optimization and Transaction Reduction," Proc. 2nd International conference on Artificial Intelligence, Management Science and Electronic Commerce, Deng Leng 2011 , pp. 1908-1911.

[10] Hanbing Liu ,Baisheng Wang, "An Association Rule Mining Algorithm Based On A Boolean Matrix in Data Science Journal," Volume 6, Supplement, 9 September 2007.

[11] IBM DB2 XML support, Available at http:// www.oxygenxml.com/doc/IBM_DB2_XML_support.pdf.