An Efficient Association Rule Mining For XML Data

A.Bharathi[1] *K.AnandaKumar*[2]

Abstract

XML association rule mining is an important problem in data mining domain. Currently, the problem of association rule mining on XML data has not been well studied. In this paper, we proposed an efficient association rule mining for large amount of XML data. The set of data is viewed as a binary table. The value of the itemset is one, if the corresponding XML data exist in the dataset, zero for otherwise. The frequent itemset is generated along with the candidate key. The closed itemset for the given data set is also generated. The closed itemset don't have any superset. For both frequent itemset and closed itemset generation we use Apriori algorithm. The possible association rules are generated for the XML data. Then the generated Association rule is converted into XML format. Our proposed system EARM may reduce the memory storage size and it returns association rules with short response time.

Keywords: Association rules, Data Mining, Apriori Algorithm.

1. Introduction

Data Mining is the technique that is used to store and retrieve all types of data. During the mid- to late 1990s, commercial vendors began exploring the feasibility of applying traditional statistical and artificial intelligence analysis techniques to large databases for the purpose of discovering hidden data attributes, trends, and patterns. This exploration evolved into formal data-mining toolsets based on a wide collection of statistical analysis techniques. Data-mining techniques can generally be grouped into one of three categories: clustering, classifying, and predictive. The proliferation of database management systems has also contributed to recent massive gathering of all sorts of information. Today, we have far more information than we can handle: from business transactions and scientific data, to satellite pictures, text reports and military intelligence. Information retrieval is simply not enough anymore for decision-making. Confronted with huge collections of data, we have now created new needs to help us make better managerial choices. These needs are automatic summarization of data, extraction of the "essence" of information stored, and the discovery of patterns in raw data.

The goal of the project is to generate the association rules for XML data. This will reduce the memory storage size and it returns the association rules with short response time. The input XML document is given from which the frequent itemset and closed itemset is found using the apriori algorithm. The association rule is generated for frequent itemset and it is converted into XML format.

[1] Bannari Amman Institute of Technology. Email : bharathi_aa@rediffmail.com

[2] Dr.S.N.S.Rajalakshmi College of Arts and Science. Email : anandhsns@yahoo.co.in

2. Literature Review

Data mining is a task of discovering interesting patterns from large amount of data where the data can be stored in databases, data warehouses or other information repositories [1]. Data mining is the principle of sorting through large amounts of data and picking out relevant information. It is usually used by business intelligence organizations, and financial analysts, but it is increasingly used in the sciences to extract information from the enormous data sets generated by modern experimental and observational methods. It is a young field, drawing work from areas including database technology, artificial intelligence, neural networking, statistics, pattern recovery, knowledge acquisition and many other fields such as business, economics and bioinformatics.

2.1 Data Mining Categories

Clustering techniques group information based on a set of input patterns using an unsupervised or undirected algorithm. One example of clustering could be the analysis of business consumers for unknown attribute groupings.

Classifying techniques group or assign objects to predetermined groupings based on well-defined attributes. The groupings are often clusters discovered using the above techniques. An example would be assigning a consumer to a particular sales cluster based on their income level.

Predictive techniques take as input known attributes regarding a particular object or category and apply those attributes to another similar group to identify expected behavior or outcomes. For example, if a group of individuals wearing helmets and shoulder pads is known to be a football team, we can expect another group of individuals with helmets and pads to be a football team as well [2].

2.2 Data Mining Techniques

The following list describes many data-mining techniques in use today. Each of these techniques exists in several variations and can be applied to one or more of the categories above.

Regression Modeling : This technique applies standard statistics to data to prove or disprove a hypothesis. One example of this is linear regression, in which variables are measured against a standard or target variable path over time. A second example is logistic regression, where the probability of an event is predicted based on known values in correlation with the occurrence of prior similar events.

Visualization : This technique builds multidimensional graphs to allow a data analyst to decipher trends, patterns, or relationships.

Correlation : This technique identifies relationships between two or more variables in a data group.

Variance Analysis : This is a statistical technique to identify differences in mean values between a target or known variable and nondependent variables or variable groups.

Discriminate Analysis : This is a classification technique used to identify or "discriminate" the factors leading to membership within a grouping.

Forecasting : Forecasting techniques predict variable outcomes based on the known outcomes of past events.

Cluster Analysis : This technique reduces data instances to cluster groupings and then analyzes the attributes displayed by each group.

Decision Trees : Decision trees separate data based on sets of rules that can be described in "if-then-else" language.

Neural Networks : Neural networks are data models that are meant to simulate cognitive functions.

Apriori Algorithm

Data mining, more specifically the "Apriori Algorithm", is used to derive association rules that represent relationships between input conditions and results of domain experiments. This enables the tool to answer questions such as "Given cooling medium and agitation during material heat treatment, predict cooling rate". This allows users to perform case studies on the Web and use their results to optimize the involved processes, thus increasing customer satisfaction. Another interesting aspect is predicting material microstructure during heat treatment.

3. System Analysis

3.1 Existing System and Its Limitations

A first algorithm called Apriori was proposed, which generates (k+1) candidates using joins over frequent k-itemsets, must be generated by the algorithm. Although many of those frequent itemset may not be useful and may not exploit for finding association rules because some of these frequent itemset haven't any interestingness antecedent or consequent in rules but generate them to find superior frequent itemset.

3.2 Proposed Systems

Input : Transactional data, minimum support and minimum confidence.

Output : Association rules between largest frequent itemsets. Change XML data to binary table form and count support of all frequent 1-itemsets. Remove the itemsets are not satisfied with user define minimum support. Repeatedly apply AND operation must find large frequent itemsets that can not be found. Logic XOR operation is applied to derive all interesting association rules between large frequent itemsets. Display association rules with xml format.

3.3 Advantage in the Proposed System

- Reduces the memory storage.
- Returns association rule with short response time.

4. Description

4.1 Module Analysis

Our proposed system consists of following modules:

1. Data Extraction Module

2. Data Conversion Module

3. Generation of Frequent Item set

4. Generation of Closed Item set

5. Association Rules Generation

6. Convert Rules to XML Module

4.1.1 Data Extraction Module

Sample XML document is given as an input in this module. Extraction of XML document using parser is obtained as an output to this module. The extracted data will be displayed in a notepad. This output is passed as an input to the second module.

4.1.2 Data Conversion Module

The extracted data from XML document is given as input to this module. The transactions were noted. The extracted data is converted into binary format based on transaction items. If the item is present we will have a binary value of one and if the item is not present, the binary value is zero.

4.1.3 Generation of Frequent Item set

The binary formats of the transaction items are given as the input. The candidate and the frequent item sets are found for generating the Association Rules.

A frequent item set have the following characteristics:

- * Any subset of a frequent item set must be also frequent
- * A transaction containing {beer, diaper, nuts} also contains {beer, diaper}
- * {beer, diaper, nuts} is frequent à {beer, diaper} must also be frequent
- * No infrequent item set should be generated or tested.

Frequent Item set Property :

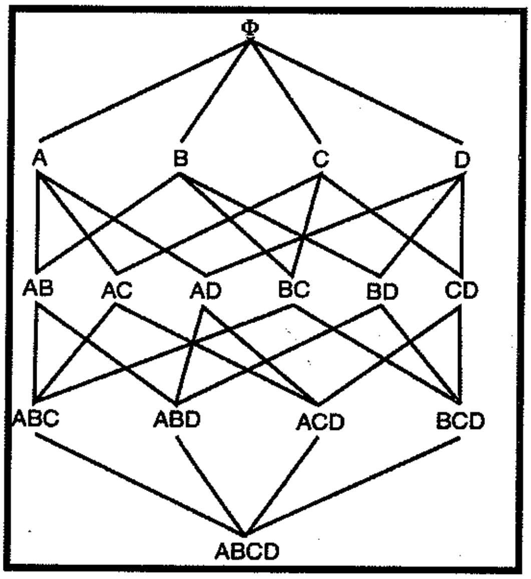


Figure 4.1 : Frequent Item Set Property

For generating the frequent item set, Apriori algorithm is used.

4.1.4 Generation of Closed Itemset

The binary formats of the transaction items are given as the input. The generation of closed itemset is the output to this module. The Mining frequent closed itemsets has the same power as mining the complete set of frequent itemsets. This reduces redundant rules to be generated and increases both efficiency and effectiveness of mining.

The idea behind this approach is to use conditional databases in a divide and conquer format. Given a list of all frequent items above minimum support, find conditional database of item in reverse order of support. Closed itemsets can be extracted iteratively from these conditional databases. Each conditional database may be further divided into more conditional databases.

4.1.5 Association Rules Generation

This module generates rules based on the support and confidence values from the frequent item sets. Association rule induction is a powerful method for so-called market basket analysis, which aims at finding regularities in the shopping behavior of customers of supermarkets, mail-order companies and the like. With the induction of association rules one tries to find sets of products that are frequently bought together, so that from the presence of certain products in a shopping cart one can infer (with a high probability) that certain other products are present. Such information, expressed in the form of rules, can often be used to increase the number of items sold, for instance, by appropriately arranging the products in the shelves of a supermarket (they may, for example, be placed adjacent to each other in order to invite even more customers to buy them together) or by directly suggesting items to a customer, which may be of interest for him/her.

An association rule is a rule like “If a customer buys wine and bread, he often buys cheese, too.” It expresses an association between (sets of) items, which may be products of a supermarket or a mail-order company, special equipment options of a car, optional services offered by telecommunication companies etc. An association rule states that if we pick a customer at random and find out that he selected certain items (bought certain products, chose certain options etc.), we can be confident, quantified by a percentage, that he also selected certain other items. Association rule mining is always done by checking the regularities in data which can be done by asking questions like

1. What products were often purchased together? — Milk and Egg
2. What are the subsequent purchases after buying a PC?
3. What kinds of DNA are sensitive to this new drug?
4. Can we automatically classify web documents?
5. Some applications where Association rule mining is used broadly are

   5a. Basket data analysis, cross-marketing, catalog design, sale campaign analysis

   5b.Web log (click stream) analysis, DNA sequence analysis, etc.[6,7]

The following is an example for generating Association rules. The table contains the Transaction ID and the items.[8]

Table 4.1: Transactions and Itemsets

TID	ITEAM
T100	I1,I2,I6
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T600	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I6
T900	I1,I2,I3

Let us consider the 3-Itemset {I1, I2, I6} with support of 2%.The generation of association rules for this itemset is as follows

I1 ^ I2 ⟹ I6 confidence=2/4=60%
I1 ^ I6 ⟹ I2 confidence=2/2=100%
I2 ^ I6 ⟹ I1 confidence=2/2=100%
I1 ⟹ I2 ^ I6 confidence=2/6=33%
I2 ⟹ I1 ^ I6 confidence=2/7=29%
I6 ⟹ I1 ^ I2 confidence=2/2=100%

4.1.6 Convert Rules to XM

An XML file is called a document which has one top-level Element. The elements much match the start and end tags or a combined tag. The content between the tags can be a text or an element. The attributes are the Name/Value pairs with the Start or combined tag.[4]

XML and HTML Difference

In HTML, both the tag semantics and the tag set are fixed. An <h1> is always a first level heading and the tag <ati.product.code> is meaningless. The W3C, in conjunction with browser vendors and the WWW community, is constantly working to extend the definition of HTML to allow new tags to keep pace with changing technology and to bring variations in presentation (stylesheets) to the Web. However, these changes are always rigidly confined by what the browser vendors have implemented and by the fact that backward compatibility is paramount. And for people who want to disseminate information widely, features supported by only the latest releases of Netscape and Internet Explorer are not useful.

XML specifies neither semantics nor a tag set. In fact XML is really a meta-language for describing markup languages. In other words, XML provides a facility to define tags and the structural relationships between them. Since there's no predefined tag set, there can't be any preconceived semantics. All of the semantics of an XML document will either be defined by the applications that process them or by stylesheets.[5]

XML and SGML Differences

XML is defined as an application profile of SGML. SGML is the Standard Generalized Markup Language defined by ISO 8879. SGML has been the standard, vendor-independent way to maintain repositories of structured documentation for more than a decade, but it is not well suited to serving documents over the web (for a number of technical reasons beyond the scope of this article). Defining XML as an application profile of SGML means that any fully conformant SGML system will be able to read XML documents. However, using and understanding XML documents does not require a system that is capable of understanding the full generality of SGML. XML is, roughly speaking, a restricted form of SGML.[6]

For technical purists, it's important to note that there may also be subtle differences between documents as understood by XML systems and those same documents as understood by SGML systems. In particular, treatment of white space immediately adjacent to tags may be different.

Why XML?

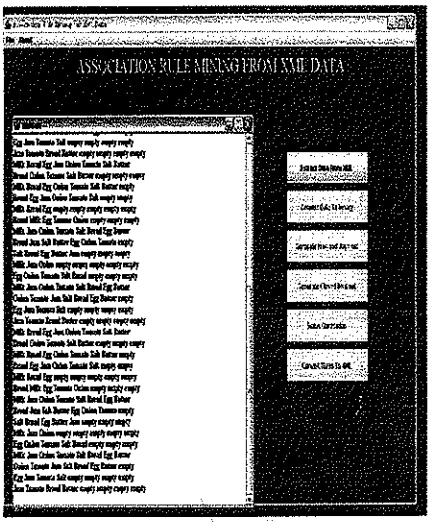
In order to appreciate XML, it is important to understand why it was created. XML was created so that richly structured documents could be used over the web. The only viable alternatives, HTML and SGML, are not practical for this purpose.

HTML, as we've already discussed, comes bound with a set of semantics and does not provide arbitrary structure. SGML provides arbitrary structure, but is too difficult to implement just for a web browser. Full SGML systems solve large, complex problems that justify their expense. Viewing structured documents sent over the web rarely carries such justification.

This is not to say that XML can be expected to completely replace SGML. While XML is being designed to deliver structured content over the web, some of the very features it lacks to make this practical, make SGML a more satisfactory solution for the creation and long-time storage of complex documents. In many organizations, filtering SGML to XML will be the standard procedure for web delivery. [3]

5. Screen Shots

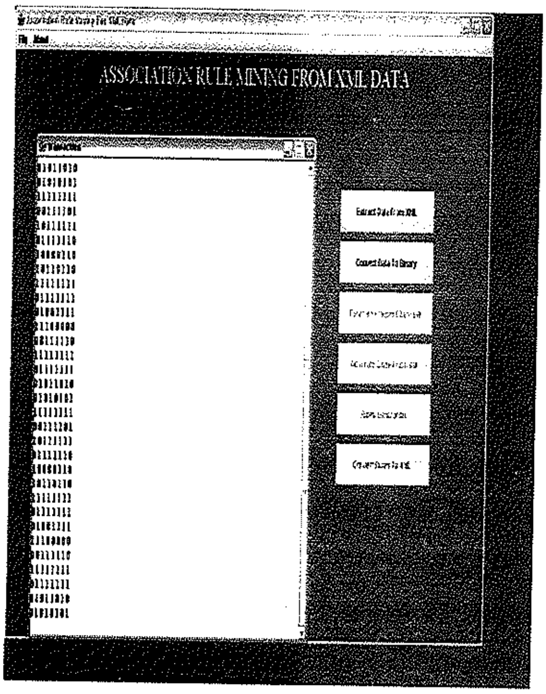
5.1 Data Extraction from XML Document



Egg Jam Tomato Salt empty empty empty empty
Jam Tomato Bread Butter empty empty empty empty
Milk Bread Egg Jam Onion Tomato Salt Butter
Bread Onion Tomato Salt Butter empty empty empty
Milk Bread Egg Onion Tomato Salt Butter empty
Bread Egg Jam Onion Tomato Salt empty empty
Milk Bread Egg empty empty empty empty empty
Bread Milk Egg Tomato Onion empty empty empty
Milk Jam Onion Tomato Salt Bread Egg Butter
Bread Jam Salt Butter Egg Onion Tomato empty
Salt Bread Egg Butter Jam empty empty empty
Milk Jam Onion empty empty empty empty empty
Egg Onion Tomato Salt Bread empty empty empty
Milk Jam Onion Tomato Salt Bread Egg Butter
Onion Tomato Jam Salt Bread Egg Butter empty
Egg Jam Tomato Salt empty empty empty empty
Jam Tomato Bread Butter empty empty empty empty
Milk Bread Egg Jam Onion Tomato Salt Butter
Bread Onion Tomato Salt Butter empty empty empty
Milk Bread Egg Onion Tomato Salt Butter empty
Bread Egg Jam Onion Tomato Salt empty empty
Milk Bread Egg empty empty empty empty empty
Bread Milk Egg Tomato Onion empty empty empty
Milk Jam Onion Tomato Salt Bread Egg Butter
Bread Jam Salt Butter Egg Onion Tomato empty
Salt Bread Egg Butter Jam empty empty empty
Milk Jam Onion empty empty empty empty empty
Egg Onion Tomato Salt Bread empty empty empty
Milk Jam Onion Tomato Salt Bread Egg Butter
Onion Tomato Jam Salt Bread Egg Butter empty
Egg Jam Tomato Salt empty empty empty empty
Jam Tomato Bread Butter empty empty empty empty

Figure 5.1 : Data Extraction from XML Document

5.2 Data Conversion



```
[illegible]
```

Figure 5.2 : Data Conversion

5.3 Generate Frequent Itemset

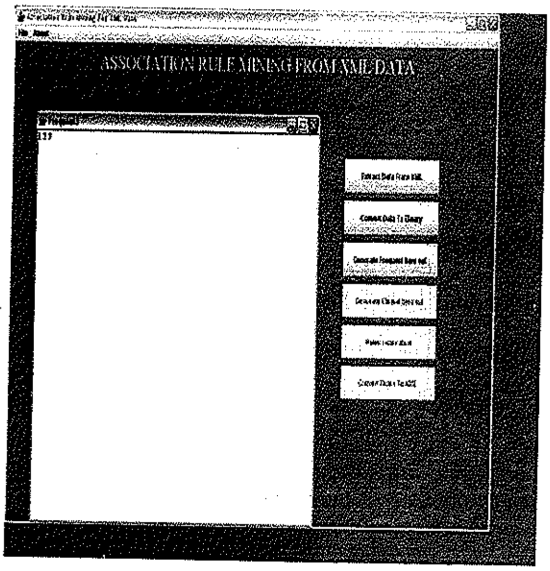
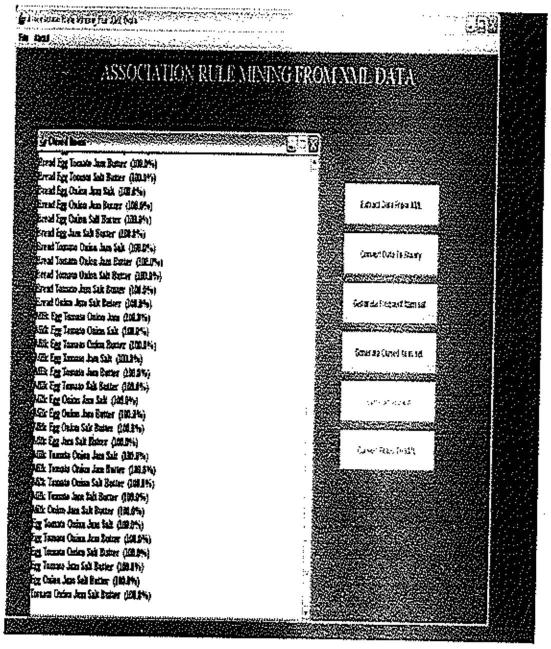


Figure 5.3 : Generate Frequent Itemset

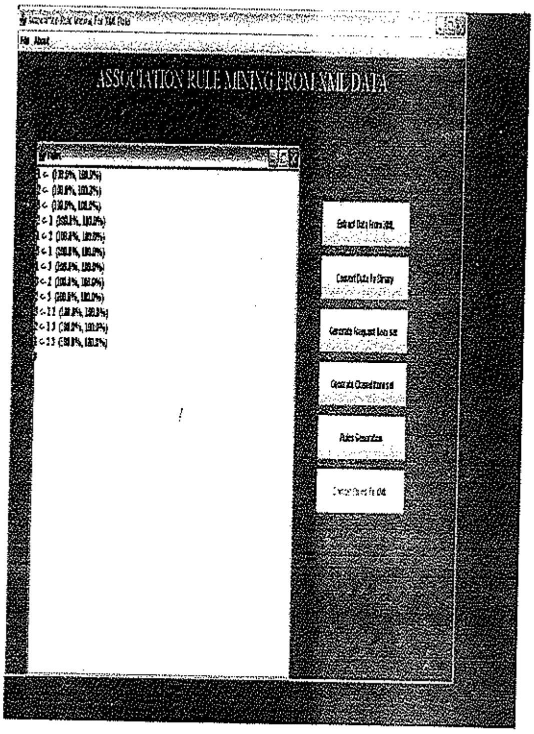
5.4 Generate Closed Itemset



```
Bread Egg Tomato Jam Butter (100.0%)
Bread Egg Tomato Salt Butter (100.0%)
Bread Egg Onion Jam Salt (100.0%)
Bread Egg Onion Jam Butter (100.0%)
Bread Egg Onion Salt Butter (100.0%)
Bread Egg Jam Salt Butter (100.0%)
Bread Tomato Onion Jam Salt (100.0%)
Bread Tomato Onion Jam Butter (100.0%)
Bread Tomato Onion Salt Butter (100.0%)
Bread Tomato Jam Salt Butter (100.0%)
Bread Onion Jam Salt Butter (100.0%)
Milk Egg Tomato Onion Jam (100.0%)
Milk Egg Tomato Onion Salt (100.0%)
Milk Egg Tomato Onion Butter (100.0%)
Milk Egg Tomato Jam Salt (100.0%)
Milk Egg Tomato Jam Butter (100.0%)
Milk Egg Tomato Salt Butter (100.0%)
Milk Egg Onion Jam Salt (100.0%)
Milk Egg Onion Jam Butter (100.0%)
Milk Egg Onion Salt Butter (100.0%)
Milk Egg Jam Salt Butter (100.0%)
Milk Tomato Onion Jam Salt (100.0%)
Milk Tomato Onion Jam Butter (100.0%)
Milk Tomato Onion Salt Butter (100.0%)
Milk Tomato Jam Salt Butter (100.0%)
Milk Onion Jam Salt Butter (100.0%)
Egg Tomato Onion Jam Salt (100.0%)
Egg Tomato Onion Jam Butter (100.0%)
Egg Tomato Onion Salt Butter (100.0%)
Egg Tomato Jam Salt Butter (100.0%)
Egg Onion Jam Salt Butter (100.0%)
Tomato Onion Jam Salt Butter (100.0%)
```

Figure 5.4 : Generate Closed Itemset

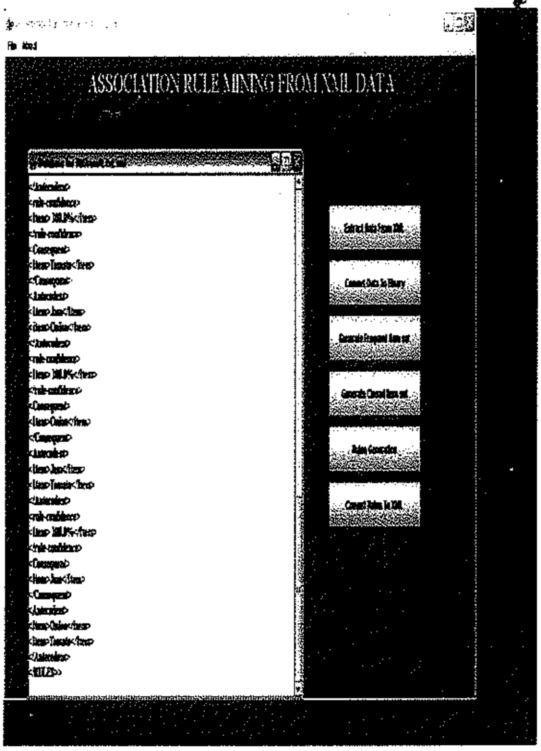
5.5 Association Rule Generation



```
1 <- (100.0%, 100.0%)
2 <- (100.0%, 100.0%)
3 <- (100.0%, 100.0%)
2 <- 1 (100.0%, 100.0%)
1 <- 2 (100.0%, 100.0%)
3 <- 1 (100.0%, 100.0%)
1 <- 3 (100.0%, 100.0%)
3 <- 2 (100.0%, 100.0%)
2 <- 3 (100.0%, 100.0%)
3 <- 1 2 (100.0%, 100.0%)
2 <- 1 3 (100.0%, 100.0%)
1 <- 2 3 (100.0%, 100.0%)
```

Figure 5.5 : Association Rule Generation

5.6 Convert Rules To XML Format



```
</Antecedent>
<rule-confidence>
<Item> [illegible]</Item>
</rule-confidence>
<Consequent>
<Item>Tomato</Item>
</Consequent>
<Antecedent>
<Item>Jam</Item>
<Item>Onion</Item>
</Antecedent>
<rule-confidence>
<Item> [illegible]</Item>
</rule-confidence>
<Consequent>
<Item>Onion</Item>
</Consequent>
<Antecedent>
<Item>Jam</Item>
<Item>Tomato</Item>
</Antecedent>
<rule-confidence>
<Item> [illegible]</Item>
</rule-confidence>
<Consequent>
<Item>Jam</Item>
</Consequent>
<Antecedent>
<Item>Onion</Item>
<Item>Tomato</Item>
</Antecedent>
</RULES>>
```

Figure 5.6 : Convert Rules to XML Format

6. Conclusion

The proposed system for mining association rules for XML data has been successfully implemented. This proposed algorithm may reduce storage memory size. The user-friendly interface designed for this paper can enable anyone to learn about the association rule mining for XML databases.

7. Future Enhancement

Currently, this system uses static XML data and the algorithm uses text file as input. This can be extended to generate output for the dynamic XML data and from the XML databases. The system can be tested with the real-time data in the future for improving it's accuracy in generating association rules. Since, this is an ongoing research work in the area of data mining; any open issues can be addressed in the future.

References

[1] A. Das, Y. K. Woon and W. K. Ng, *"Rapid Association Rule Mining"*, 10th International Conference on Information and Knowledge Management, 2000.

[2] E.H.Han, G. Karypis and V. Kumer, *"Scalable Parallel Data Mining for Association rules"*, IEEE Transactions on Knowledge and Data Engineering,12930:337-362, May/June 2000.

[3] J. S. park, M. S. Chen and P. S. Yu, *"In Effective Hash Based Algorithm for Mining Association Rules"*, Dawak 2001, Springer Verlag Berlin Hidelberg 2001.

[4] J.Park, D. Won, F. Fotouhi and U. Kin, *"ExiT-B: Anew approach for extracting maximal frequent subtreefrom XML da*ta", IDEAL 2008.

[5] A. Terminer, M. C. Rouset and M. Sebag, *"TreeFinder: a First Step towards XML Data Mining"*, IEEE International Conference on Data Mining, 2002.

[6] J. Zhang, T. W. Ling, R. M. Bruckner, A. M. Tjoa and H. Lui, *"described on Effective Association Rule Mining from XML data*", LNCS 3180, PP.494608.

[7] J. D. Holt and S. M. Chung, *"Efficient mining of association rules in text databases"*, In Proceedings of CIKM'99, 1999.

[8] H. Lu, L. Feng and J. Han, *"Beyond intratransaction association analysis: mining multidimensional intertransaction association rules"*, ACMTrans. Inf.Syst, 18(4):423–454, 2000.

[9] A. K. H. Tung, H. Lu, J. Han and L. Feng, *"Efficient mining of intertransaction association Rules"*, IEEE Transactions on Knowledge and Data Engineering, 15(1):43–56, 2003.

[10] D.Janetzko, H.Cherfi, R.Kennke, A.Napoli and Y.Toussaint, *"Knowledge-based selection of association rules for text mining"*, In Proceedings of ECAI'2004, 2004.

[11] C.H. Lee, C.R. Lin and M.S.Chen, *" On mining general temporal association rules in a publication database"*, In Proceedings of ICDM'2001, 2001.

[12] R. Agrawal and R. Srikant, *"Fast algorithms for mining association rules"*, In Proc. of Intl. Conf. on Very Large Databases (VLDB), Sept. 1994.

[13] J. Han, J. Pei and Y. Yin, *"Mining frequent patterns without candidate generation"*, In Proc. of ACM SIGMOD Intl. Conf. on Management of Data, May 2000.

[14] C. Hidber, *"Online association rule mining"*, In Proc. of ACM SIGMOD Intl. Conf. on Management of Data, June 1999.

[15] A. Savasere, E. Omiecinski and S. Navathe, *"An efficient algorithm for mining association rules in large databases"*, In Proc. of Intl. Conf. on Very Large Databases (VLDB), 1995.

[16] R. Srikant and R. Agrawal, *"Mining generalized association rules"*, In Proc. of Intl. Conf. on Very Large Databases (VLDB), Sept. 1995.

[17] Y. Xiao and M. H. Dunham, *"Considering main memory in mining association rules"*, In Proc. of Intl. Conf. on Data Warehousing and Knowledge Discovery (DAWAK), 1999.

Author's Biography

A.Bharathi received her Bachelor of Engineering Degree from Kongu Engineering College in 1998, Perundurai, Master of Engineering Degree from Bannari Amman Institute of Technology, Sathyamangalam, in 2007 and she is doing Doctor of Philosophy in Computer Science and Engineering from Anna University, Coimbatore. She had 12 years of teaching experience. currently she is working as Assistant Professor, Department of IT, Bannari Amman Institute of Technology, and Sathyamangalam. Her Professional activities include Guided Ten UG projects and guiding Seven UG and Three PG projects. She is presented 9 papers in International and National Conferences. She is published 3 national and 1 international journals.

K.AnandaKumar was born in TamilNadu, India on March 1975. He received the B.Sc Degree in Physics from Bharathiar University in 1995. He received his MCA Degree in Computer Applications from Bharathiar University in 1998. He received his M.Phil from Periyar University in 2006 and he is doing Doctor of Philosophy in Computer Science and Engineering from Bharathiar University, Coimbatore. He had 12 years of teaching experience. Currently he is working as HOD in Computer Applications Department, Dr.SNS Rajalakshmi College of Arts and Science College, Coimbatore. His Professional activities include Guided Fifteen PG projects and Five M.Phil and guiding Seven PG and Three M.Phil projects. Published and presented 4 papers in International and National journals and also Conferences.