**Available Online at www.ijcsmc.com**

**International Journal of Computer Science and Mobile Computing**

**A Monthly Journal of Computer Science and Information Technology**

**ISSN 2320–088X**

***IJCSMC, Vol. 3, Issue. 7, July 2014, pg.905 – 911***

**RESEARCH ARTICLE**

# MINING ASSOCIATION RULES FROM XML DOCUMENT

**Neha M. Shroff, G. V. Gujar**

M.TECH CSE, Dr. Babasaheb Ambedkar Marathwada University Aurangabad, India
neha3shroff@gmail.com; ganeshvgujar@gmail.com

***Abstract: In this work we describe an approach to mine Tree-based association rules from XML documents. Such rules provide information on both the structure and the content of XML documents; moreover, they can be stored in XML format to be queried later on. The mined knowledge is approximate, intensional knowledge used to provide: (i) quick, approximate answers to queries and (ii) information about structural regularities that can be used as dataguides for document querying. A prototype of the proposed system is also briefly described.***

## I. INTRODUCTION

XML is a standard for Web data; XML query processing takes a particular importance. The efficient exploitation of XML documents has attracted significant attention since the publication of the XML standard in 1998. More recently the problem has been investigated in the XML context [2], [3], [4], [5], [6], [7], [8]. In [9] authors use XQuery [10] to extract association rules from simple XML documents. XML queries can now be evaluated by mainstream relational database engines extended to support the XML data type and the XQuery language, as well as by in-memory processors. In order for query formulation to be effective users need to know this structure in advance, which is often not the case. In fact, it is not mandatory for an XML document to have a defined schema: 50% of the documents on the web do not possess one [1]. When users specify queries without knowing the document structure, they may fail to retrieve information which was there, This paper addresses the need of *getting the gist* of the document before querying it, both in terms of content and structure. Discovering recurrent patterns inside XML documents provides high-quality knowledge about the document content: frequent patterns are in fact *intensional* information about the data contained in the document itself, that is, they specify the document in terms of a set of properties rather than by means of data This paper introduce a proposal for mining and storing TARs (Tree-based Association Rules) as a means to represent intensional knowledge in native XML. Intuitively, a TAR represents intensional knowledge in the form $SB \Rightarrow SH$, where $SB$ is the body tree and $SH$ the head tree of the rule and $SB$ is a subtree of $SH$. procedure is characterized by the following key aspects: a) it works directly on the XML documents, without transforming the data into any intermediate format, b) it looks for general association rules, without the need to impose what should be contained in the antecedent and consequent of the rule, c) it stores association rules in XML format, and d) it translates the queries on the original dataset into queries on the TARs set. Figure 1 shows sample XML document. The aim of our proposal is to provide a way to use intensional knowledge as a substitute of the original document during querying and not to improve the execution time of the queries over the original XML dataset.

## II. METHODOLOGY

Proposed system

- In this work we describe an approach based on Tree-based Association Rules (TARs) mined rules, which provide approximate, intensional information on both the structure and the contents of XML documents, and can be stored in XML format as well.
- By mining frequent patterns from XML documents, we provide the users with partial, and often approximate, information both on the document structure and on the content [7].
- Such patterns can be useful for the users to obtain information and implicit knowledge on the documents and thus be more effective in query formulation.
- Moreover, this information is also useful for the system, which is provided with discovered information.

The proposed method consists of the following steps such as;

**1)** Indexing: The input XML data is given to indexing process that converts the XML data into the two indices (data index and node index) which will make search easier.

**2)** Selecting the exact T-type node: The corresponding nodes will be selected through our designed statistical dependent formulae such as Dscore and Tscore .

**3)** Data search and Ranking of search results: Once selection of T-type nodes, the relevant data are obtained based on the sorting the node type paths. Finally, ranking will be done based on the search results obtained from the previous steps with our designed ranking [8].

Definition 3.1(Structural Node) A tag name is used to label XML node called a structural node. Internal node is defined as children's of structural node; otherwise, it is called a leaf node [3].

Definition 3.2(T type node) A T type node is considered as a desired search for node if, T type node is intuitively related to every query keyword, XML nodes of T type should be informative enough to contain enough relevant information and XML nodes of type T should be not overwhelming to contain too much irrelevant information .

Definition 3.2 (Data Node) the leaf node of XML data containing text values and have no tag name is called as data node.

The primary intention of our research is to design and develop a technique for keyword search over XML data. The real motivation of the work is come out from the XML search technique given in [10], in which they have used TF*IDF strategy by addressing two challenges. When analyzing the existing work [10], the finding is that term frequency-based score computation was not much impressive in selecting the exact T-type node. Incorporating some other features along with frequency can lead to effective T-type search in XML data. Also, the ranking of the search results is important for the users if search output is significantly high. This problem can be solved easily by putting the effective ranking mechanism.

The above mentioned two challenges will be solved using the proposed methodology.

The proposed method consists of the three major steps such as,

1) Indexing:

A specific indexing method is proposed that builds two indices viz. Nodeindex and Data index for structural nodes and data nodes respectively. These two indices are represented in Table1 and Table 2 for DBLP XML document. In contrast to the indices presented in[10], the proposed approach stores node name of each structural node, frequency of occurrence of each structural node either in Ttyped nodes or their subtrees, prefix path of the corresponding T-typed nodes in the node index and name of data nodes. Corresponding node names and frequency of occurrences of each data node in XML document is stored in data index. The data node information table is dependent on the Node index in relation with the node name

2) Selecting the exact T-type node:

A T type node is considered as a desired search for node if, T type node is intuitively related to every query keyword, XML nodes of T type should be informative enough to contain enough relevant information and XML nodes of type T should be not overwhelming to contain too much irrelevant information .

3) Data search and Ranking of search results.

At first, the input XML data is given to indexing process that converts the XML data into the indexed format to make search easier. Then, the corresponding T-type nodes are selected through our designed statistical dependent formulae. Once we select T-type nodes, the relevant data are obtained based on the similarity matching with the input query. Finally, ranking will be done based on the search results obtained from the previous steps with our designed ranking measure.

- Mathematics and Statistical:

1) D Score:

$$\text{Dscore} = \frac{\text{depth of LCA node}}{\text{depth of HCA node}} \tag{1}$$

2) T Score:

$$P\left(\frac{q}{T}\right) = \frac{p(q \cap T)}{p(T)} \tag{2}$$

Where;

P(q/T) is defined as the chance of event ‘q’ when event ‘T’ have occurred, P(q n T) is the occurrence of event ‘q’ in event ‘T’, P(T) is defined as the probability of occurrence of event ‘T’.

$$P\left(\frac{q}{T}\right) = \sum \frac{p(k)}{p(T)} \tag{3}$$

$$P\left(\frac{q}{T}\right) = \left(\frac{1}{p(T)}\right) \times \sum P(k) \tag{4}$$

P (T) is constant for no of keywords (‘k’=1 to n) in the query

$$P\left(\frac{q}{T}\right) = \alpha \times \sum p(k) \;\}\alpha = \frac{1}{p(T)} \tag{5}$$

Thus, to estimate the best T-node type the percentage of frequency of occurrence of ‘k’ at that node type is very important and hence it is considered as the Tscore% of a particular node and the node having highest Tscore% is the relevant type node and is defined as- Therefore,

$$\text{T Score} = \alpha \times \sum p(k) \tag{6}$$

But, P (k) can also be defined as the frequency of occurrence of ‘k’ at that node type
‘T’ and P (T) can also be defined as the frequency of the node type-T. And hence defined in equation (6) as;

$$\text{TSCORE} = \alpha \times \sum f(k) \;\} \text{ for, } \alpha = \frac{1}{p(T)} \tag{7}$$

Thus the Tscore percentage is defined as,

$$T_{\text{SCORE}} = \alpha \times \sum f(k) \times 100 \tag{8}$$

The percentage score of the optimal node type Tscore% is thus defined as, the percentage of frequency of occurrence of keywords in the query at a particular node type with respect to the frequency of occurrence of that node type defined in equation(8).

### 2.1 Tree-Based Association Rules

Association rules [2] describe the co-occurrence of data items in a large amount of collected data and are usually represented as implications in the form $X \Rightarrow Y$ , where X and Y are two arbitrary sets of data items, such that $X \cup Y = \emptyset$;. The quality of an association rule is usually measured by means of support and confidence. Support corresponds to the frequency of the set $X \cup Y$ in the dataset, while confidence corresponds to the conditional probability of finding Y, having found X and is given by sup(X ∪ Y )/sup(X). In this work we extend the notion of association rule originally introduced in the context of relational databases, in order to adapt it to the hierarchical nature of XML documents. We are interested in finding relationships among subtrees of XML documents. Thus, since both textual content of leaf elements and values of attributes convey "content", we do not distinguish between them. Given a tree $T = (N_T ,E_T , r_T , l_T , c_T)$ , a subtree of T, $t = (N_t ,E_t, r_t, l_t, c_t)$ and a user-fixed support threshold *smin*: (i) *t* is frequent if its support is greater or at least equal to *smin*; (ii) *t* is maximal if it is frequent and none of its proper supertrees is frequent; (iii) *t* is closed if none of its proper subtrees has support greater than that of *t*.

Figure 2 shows an example of an XML document (Figure 2a), its tree-based representation (Figure 2b), induced subtrees and a rooted subtree (Figure 2c). Thus, every tree-based association rule is characterized by two measures:

a) $sT_r$ support, measures the frequency of the tree SH in the XML document

b) $cT_r$ confidence, measures the reliability of a rule, that is the frequency of the tree SH, once SB has already been found.

Given function count(S;D) denoting the number of occurrences of a subtree S in the tree D and function cardinality(D) denoting the number of nodes of D, it is possible to define formally the two measures as:

Support and confidence.

$$\text{Support}\,(SB \Rightarrow SH) = \frac{\text{count}(sH,D)}{\text{cardinality}(D)}$$

$$\text{Confidence}(SB \Rightarrow SH) = \frac{count(sH,D)}{count(SB,D)}$$

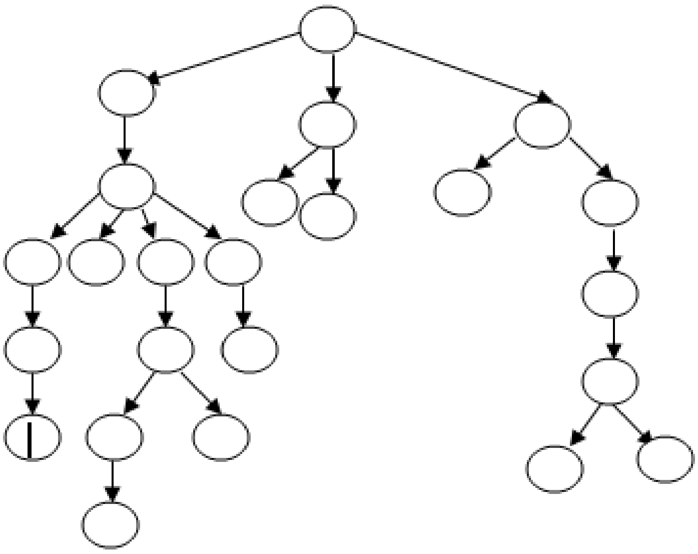
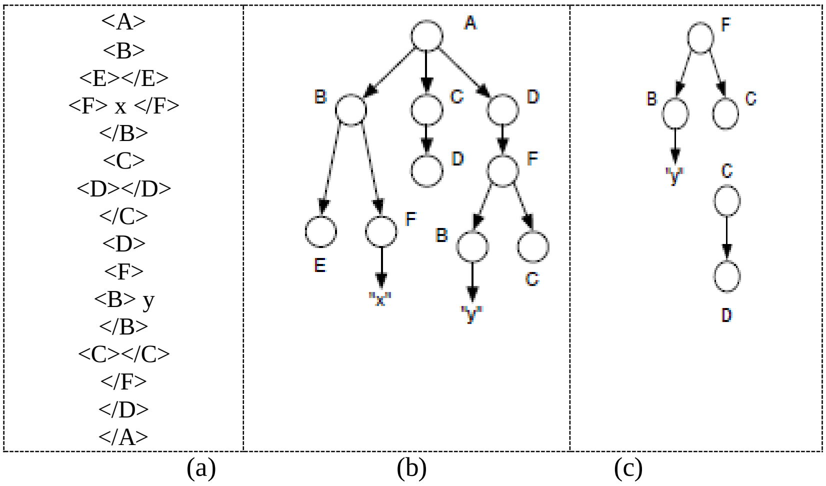


Fig 1. Sample XML Document

Given an XML document it is possible to extract two types of tree-based association rules:

a) iTARs: instance TARs are association rules providing information both on the structure and on the PCDATA values contained in a target XML document

b) sTARs: structure TARs are association rules on the structure of the XML document. An sTAR is a tuple $Ti = (SB, SH, sT_i , cT_i)$ where, for each node n either in SB or in SH, n has as label a tuple(TAG, TYPE, ⊥), i.e. no PCDATA is present in an sTAR



```
<A>
<B>
<E></E>
<F> x </F>
</B>
<C>
<D></D>
</C>
<D>
<F>
<B> y
</B>
<C></C>
</F>
</D>
</A>
```

(a) (b) (c)

Fig 2: a) an Xml document b) its tree baesd representation and c) induced subtrees

### 2.2. The use of TAR's

Association rules describe the co-occurrence of data items in a large amount of collected data and are represented as implications of the form X => Y, where X and Y are two arbitrary sets of data items. The quality of an association rule is measured by means of support and confidence.

#### *2.2.1. Intensional answers to queries*

The classes of queries that can be managed with our approach have been introduced in and further analyzed in the relational database context in. Here are some simple examples for four classes of queries, discussed in the following.

**Class 1:** This kind of query is used to impose a simple, or complex (containing AND and OR operators), restriction on the value of an attribute or the content of a leaf node.

**Class 2:** This kind of query is used to retrieve some properties described in the subtrees rooted in a specified element, possibly ordering the result

**Class 3:** This kind of query is used to count the number of elements with a specific content. This paper uses an association rule whose body matches the query conditions, and obtain as answer.

**Class 4:** This kind of query is used to select the best k answers satisfying a counting an grouping condition, for example Retrieve the k authors who wrote the highest number of articles".

## III. Results

The experimental results obtained are tabulated and these results are compared with the existing method XReal. The results generated and compared are tested for the real datasets; viz., DBLP, WSU, and eBay [10, 2], and are further discussed in terms of effectiveness and efficiency

**Effectiveness test**

The effectiveness of our approach for a statistical dependent and ranking measure for keyword search over XML data is addressed by identifying the user search intention and resolving the ambiguity issues. The accuracy of our approach is tested by evaluating the user search intention for the search for node type for the query tabulated in the table

TABLE I
Node Index

| Sr. No | Node | Frequency | Path |
|---|---|---|---|
| 300 | Author | 212898 | Dblp,artile |
| 302 | url | 106805 | Dblp,artile |
| 303 | Publisher | 4 | Dblp,artile |
| 307 | year | 72 | Dblp,phdthesis |
| 311 | Publisher | 3 | Dblp,phdthesis |
| 319 | Author | 14 | Dblp,www |

| | | | |
|---|---|---|---|
| 320 | Editor | 21 | Dblp,www |
| 321 | Booktitle | 1 | Dblp,www |
| 324 | Title | 2609 | Dblp,proceeding |
| 326 | Series | 1599 | Dblp,proceeding |

TABLE II
DATA INDEX

| **Sr.No** | **Data** | **Node** | **Frequency** |
|---|---|---|---|
| 30 | Db/labs/gte/index.html#TR-0169-12 | url | 1 |
| 32 | Db/labs/gte/TR-0231-08-93-165.html | ee | 1 |
| 33 | Sandra heiler | author | 7 |
| 35 | TR-0231-08-93-165 | vol | 8 |
| 36 | 1993 | Year | 4144 |
| 38 | June | Cdrom | 1 |
| 42 | GTEMAN093c.pdf | Ee | 8 |

## IV. CONCLUSIONS

The main goals to achieve in this work are: 1) mine all frequent association rules without imposing any a-priori restriction on the structure and the content of the rules; 2) store mined information in XML format; 3) use extracted knowledge to gain information about the original datasets. performed four types of experiments: 1) time required for the extraction of the intensional knowledge from an XML database; 2) time needed to answer intensional and extensional queries over an XML file; 3) a use case scenario on the XML database, in order to monitor extraction time given a specific support or confidence; 4) a study of the accuracy of intensional answers.. In this paper, not discussed the updatability of both the document storing TARs and their index. As future work, we are studying how to incrementally update mined TARs when the original XML datasets change and how to further optimize our mining algorithm; moreover, for the moment we deal with a (substantial) fragment of XQuery; we would like to find the exact fragment of XQuery which lends itself to translation into intensional queries.

## ACKNOWLEDGMENT

First and foremost I would like to thank my advisor, Prof. Dr. Sachin Deshmukh, for his guidance, patience, and support. I would also like to thank my parents for their support throughout my education.

## REFERENCES

[1] D. Barbosa ,L. Mignet, andP. Veltri. Studying the xml web:Gathering statistics from an xml sample. World Wide Web, 8(4):413–438, 2005.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In Proc. of the 20th Int. Conf. on Very Large Data Bases, pages 487–499. Morgan Kaufmann Publishers Inc., 1994

[3] D. Braga, A. Campi, S. Ceri, M. Klemettinen, and P. Lanzi. Discovering interesting information in xml data with association rules. In Proc. Of the ACM Symposium on Applied Computing, pages 450–454, 2003.

[4] J. W. W. Wan and G. Dobbie. Extracting association rules from xml documents using xquery. In Proc. of the 5th ACM Int. Workshop on Web Information and Data Management, pages 94–97. ACM Press, 2003.

[5] J. Paik, H. Y. Youn, and U. M. Kim. A new method for mining association rules from a collection of xml documents. In Proc. of Int. Conf. on Computational Science and Its Applications, pages 936–945, 2005.

[6] L. Feng, T. S. Dillon, H. Weigand, and E. Chang. An xml-enabled association rule framework. In Proc. of the 14th Int. Conf. on Database and Expert Systems Applications, pages 88–97, 2003.

[7] H. C. Liu and J. Zeleznikow. Relational computation for mining association rules from xml data. In Proc. of the 14th ACM Conf. on Information and Knowledge Management, pages 253–254, 2005.

[8] K. Wang and H. Liu. Discovering structural association of semi structured data. IEEE Transactions on Knowledge and Data Engineering, 12(3):353–371, 2000.

[9] J. W. W. Wan and G. Dobbie. Extracting association rules from xml documents using xquery. In Proc. of the 5th ACM Int. Workshop on Web Information and Data Management, pages 94–97. ACM Press, 2003.

[10] World Wide Web Consortium. XQuery 1.0: An XML query language, 2007. http://www.w3C.org/TR/xquery.