ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 8, October 2012

XML Query Answering using Tree based Association Rules

Mrs.Mopuri Sujatha
M.Tech(CSE)
C.V.S.R College Of Enginering
Hyderabad, India

Mrs. Dhyaram Lakshmi Padmaja M.Tech(Ph.d)
Associate Professor
C.V.S.R College Of Engineering
Hyderabad, India

***Abstract*— Mainly two approaches are used to access XML document: keyword-based search and query-answering. The first one comes from the tradition of information retrieval. Where most searches are performed on the textual content of the document; this means that no advantage is derived from the semantics conveyed by the document structure. As for query-answering, since query languages for semi structured data rely the one document structure to convey its semantics, in order for query formulation to be effective users need to know this structure in advance, which is often not the case. When users specify queries without knowing the document structure, they may fail to retrieve information which was there, but under a different structure.**

***Index Terms*— approximate query-answering, data mining, intentional information, XQuery, XML.**

I. INTRODUCTION

The goal of data mining is to extract or mine" knowledge from large amounts of data. Consequently, data mining consists of more than collecting and managing data, it also includes analysis and prediction. The eXtensible Markup Language (XML)[1] has become a standard language for data representation and exchange XML is a Standard, flexible syntax for data exchanging Regular, structured data. Mining of XML documents significantly differs from structured data mining and text mining. XML allows the representation of semi-structured and hierarchal data containing not only the values of individual items but also the relationships between data items. Due to the inherent flexibility of XML, in both structure and semantics, discovering knowledge from XML data is faced with new challenges as well as benefits. Mining of structure along with content provides new insights and means into the process of knowledge discovery.

As for query-answering, since query languages for semi structured data rely the on document structure to convey its semantics, in order for query formulation to be effective users need to know this structure in advance, which is often not the case. . This limitation is a crucial problem which did not emerge in the context of relational database management systems. As a consequence, when accessing for the first time a large dataset, gaining some general information about its main structural and semantic characteristics helps investigation on more specific details. 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 intentional 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. As opposed to detailed and precise information conveyed by the data, this information is partial and often approximate, but synthetic, and concerns both the document structure and its content.

XQuery:

XQuery was designed to query XML data. XQuery is designed to query XML data - not just XML files, but anything that can appear as XML, including databases. XQuery[2] is built on XPath expressions. XQuery is a language for finding and extracting elements and attributes from XML documents. XQuery can be used to (a) Extract information to use in a Web Service. (b) Generate summary reports. (c) Transform XML data to XHTML. (d) Search Web documents for relevant information

XQuery Example

for $x inoc("student.xml")/college/stud where $x/rollno>30 order by $x/rollno return $x/name .

XQuery is compatible with several W3C standards, such as XML, Namespaces, XSLT, XPath, and XML Schema.

II. TREE-BASED ASSOCIATION RULES FROM XML DOCUMENT

Association rule mining, one of the most important and well researched techniques of data mining. It aims to extract interesting correlations, frequent patterns, associations or casual structures among sets of items in the transaction databases or other data repositories. Mining Association rules [3] are widely used in various areas such as

telecommunication networks, market and risk management, inventory control etc.

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. In particular, the idea of mining association rules to provide summarized representations of XML documents has been investigated in many proposals either by using languages (e.g. XQuery) and techniques developed in the XML context, or by implementing graph- or tree-based algorithms. A proposal is proposed for mining and storing TARs (Tree-based Association Rules)[4] as a means to represent intensional knowledge in native XML. Intuitively, a TAR represents intensional knowledge in the form SB → SH, where SB is the body tree and SH the head tree of the rule and SB is a sub tree of SH. The rule SB → SH states that, if the tree SB appears in an XML document D, it is likely that the wider, tree SH also appears in D. fig 1 shows that the sample xml document and its induced sub trees

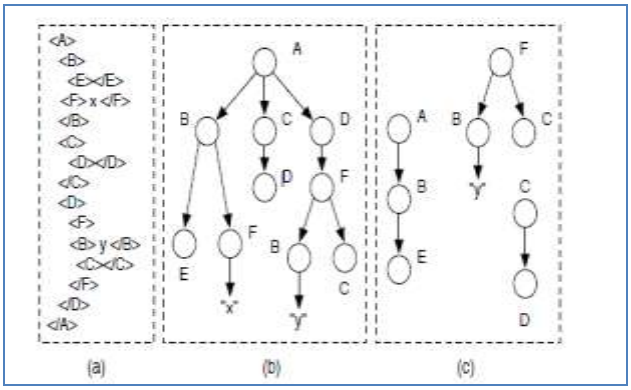

Figure 1: a) an example of XML document, b) its tree-based representation, and c) three induced subtrees

The increasing amount of very large XML datasets available to casual users is a most challenging problem and calls for an appropriate support to anciently gather knowledge from these data. Data mining, already widely applied to extract frequent correlations of values from both structured and semi structured datasets, is the appropriate tool for knowledge elicitation. Here it can described an approach to extract Tree-based association rules from XML documents. Such rules provide approximate, intentional information on both the structure and the content of XML documents, and can be stored in XML format to be queried later on. The mined knowledge is used to provide: (i) quick, approximate answers to queries and (ii) information about structural regularities. A prototype system demonstrates the electiveness of the approach.

A. *Mining Both Closed and Maximal Frequent Sub trees:*

Tree structures are used extensively in domains such as computational biology, pattern recognition, XML databases, computer networks, and so on. One important problem in mining databases of trees is to find frequently occurring subtrees. However, because of the combinatorial explosion, the number of frequent subtrees usually grows exponentially with the size of the subtrees. The CMTreeMiner [5] is a computationally efficient algorithm that discovers all closed and maximal frequent subtrees in a database of rooted unordered trees. The algorithm mines both closed and maximal frequent sub trees by traversing an enumeration tree that systematically enumerates all subtrees, while using an enumeration DAG to prune the branches of the enumeration tree that do not correspond to closed or maximal frequent subtrees. Also compare the performance of our algorithm with that of Path Join [6], a recently published algorithm that mines maximal frequent subtrees. The experiments show that this algorithm avoids the exponential explosion and therefore has better performance than Path Join for large tree sizes.

B. *Existing work and its problems:*

There is no existing approach has studied the problem of relevance oriented result ranking in depth yet. The search intention for a keyword query is not easy to determine and can be ambiguous, because the search via condition is not unique; so, how to measure the confidence of each search intention candidate, and rank the individual matches of all these candidates are challenging. Existing methods cannot resolve this ranking strategy to rank the individual matches challenge, thus it return low result quality in term of query relevance.

Disadvantages of Existing System

- Search intention for a keyword query is not easy to determine.
- It returns low result quality in term of query relevance.
- Rank the individual matches of all these queries are challenging

C. *Proposed approach:*

- Mine all frequent association rules without imposing any a-priori restriction on the structure and the content of the rules.
- Store mined information in XML format.
- Use extracted knowledge to gain information about the original datasets.

The first one comes from the tradition of information retrieval where most searches are performed on the textual content of the document; this means that no advantage is derived from the semantics conveyed by the document structure. As for query-answering, since query languages for semi structured data rely the on document structure to convey its semantics, in order for query formulation to be effective users need to know this structure in advance, which is often not the case.

Advantages of proposed system:

- Resolve keyword ambiguity Problems.
- To effectively identify the type of target node, i.e. search for node.

- To effectively infer the types of condition nodes, i.e. search via node.
- Rank the individual matches of all possible search intentions.

Fig 2 shows the sample conference.xml file and table1 shows its support and confidence of fig 3

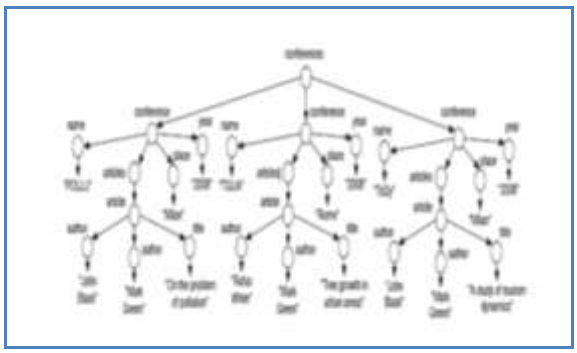

fig2: XML Sample file: "Conference.xml"

able 1: Support and confidence of rules in Fig 3

rule	rule support	body support	rule confidence
(1)	3/28 = 0.10	3/28 = 0.10	3/3 = 1.00
(2)	2/28 = 0.07	3/28 = 0.10	2/3 = 0.66
(3)	3/28 = 0.07	3/28 = 0.10	3/3 = 1.00

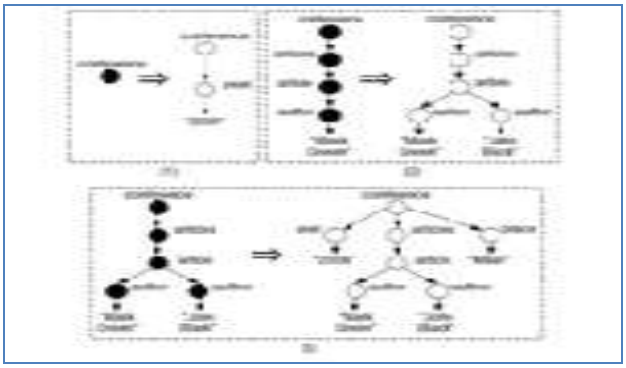

Figure 3: Sample iTARs (instance Tree-based Association Rules)

III. TAR EXTRACTION

The proposed algorithm that extends CMTTreeMiner to mine generic tree-based association rules from XML documents. To provide summarized representations of XML documents has been investigated in many proposals either by using languages and techniques developed in the XML context, or by implementing graph- or tree-based algorithms. An improved version of the TARs extraction algorithm introduced in the new version uses the better-performing CMTreeMiner to mine frequent subtrees from XML documents

Class 1: σ/π-queries. 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, possibly ordering the result. The query imposes some conditions on a node's content and on the content of its descendants, orders the results according to one of them and returns the node itself. For example "Retrieve all incidents where country types of incident were used, ordered by the date the incident was reported".

Class 2: count-queries. Used to count the number of elements having a specific content. The query creates a set containing the elements which satisfy the conditions and then returns the number of elements in such set. For example "Retrieve the number of incidents".

Class 3: top-k queries. Used to select the best k answers satisfying a counting and grouping condition. The query counts the occurrences of each distinct value of a variable in a desired set; then orders the variables with respect to their occurrences and returns the most frequent k. For example "Retrieve the k most used types of country".

A. The tree-ruler prototype

TreeRuler is a tool that integrates the functionalities proposed in our approach. Given an XML document, it enables users to extract intensional knowledge and compose traditional queries as well as queries over the intensional knowledge, receiving both extensional and intensional answers. Users formulate XQueries over the original data, and queries are automatically translated and executed on the intensional knowledge. Fig 3 shows the tree ruler architecture

.

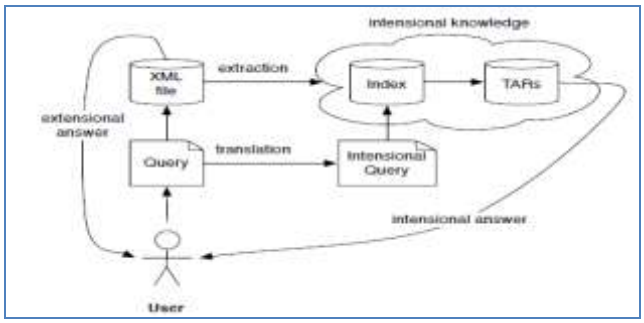

Fig3 Tree ruler architecture

The answer is given in terms of the set of TARs which reflect the search criteria. TreeRuler interface offers three tabs:

- get the Gist allows intensional information extraction from an XML document, given the support, confidence and the files where the extracted TARs and their index are to be stored.
- get the Idea allows the visualization of the intensional information as well as the original document, in order to give the user the possibility to compare the

two kinds of information.

- get the Answers allows to query the intensional knowledge and the original XML document. The user has to write an extensional query in the box on the left; when the query belongs to the classes we have analyzed it is translated into the intensional form, shown to the user in the right part of the form. Finally, once the query is executed, the TARS that reflect the search criteria.

The idea of using association rules as summarized representations of XML documents was also introduced in [7] where the XML summary is based on the extraction of rules both on the structure (schema patterns) and on content (instance patterns) of XML datasets. The limitations of this approach are:

I. The root of the rule is established a-priori and
II. The patterns used to describe general properties of the schema applying to all instances, are not mined, but derived as an abstraction of similar instance patterns and are less precise and reliable.

Table 3 shows a brief overview of the most frequent tree mining algorithms with respect to the features of the input tree (ordered, unordered) and the features of the mined patterns (induced, embedded, maximal, closed).

Table 3: Tree mining algorithms overview

Algorithm	Ordered	Unordered	Induced	Embedded	Maximal	Closed
TreeMiner [31]	⋆			⋆		
PathJoin [29]		⋆	⋆		⋆	⋆
FREQT [19]	⋆		⋆			
DRYADE [20]		⋆		⋆		⋆
DRYADEPARENT [21]		⋆		⋆		⋆
CMTTreeMiner [5]	⋆	⋆	⋆		⋆	⋆
POTMiner [13]	⋆	⋆	⋆	⋆	⋆	⋆

Algorithm 5 Class1-Query (vF,VW,CONN,vOB)

1. // the intensional query is empty
2. IQ = _€
3. if VW ≠ ∅ then
4. // get instance rules for paths with a constraint
5. IQ = IQ •get iTARs(vF,VW,CONN,false)
6. else
7. // structure rules for the path without constraint
8. IQ = IQ •get sTARs(vF)
9. end if
10. // order the results
11. IQ = IQ •"for $r in $Rules/Ruleorder by $r/vF/vOB

12:return : IQ

Fig 4 shows the example for class 1 query

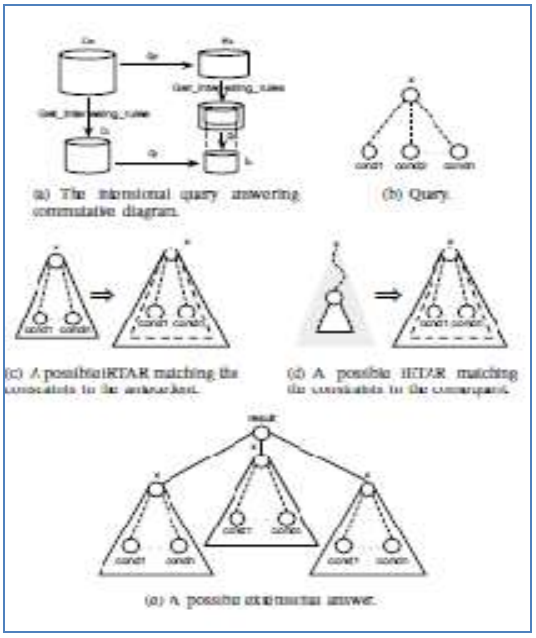

Fig 4 the intensional query answering commutative diagram and an example for class 1 queries

IV. CONCLUSIONS AND FUTURE WORK

The main goals we have achieved 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. We have developed a C++ prototype that has been used to test the effectiveness of our proposal. We have not discussed the updatability of both the document storing TARs and their index .As an ongoing 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

REFERENCES

[1] World Wide Web Consortium. XML Information Set, 2001.http://www.w3C.org/xml-infoset/
[2] World Wide Web Consortium. XQuery 1.0: An XML query language, 2007. http://www.w3C.org/TR/xquery.

[3] C 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.
[4] M. Mazuran, E. Quintarelli, and L. Tanca. Mining tree-based association rules from xml documents.In Technical Report, Politecnico di Milano. http://home.dei.polimi.it/quintare/Papers/MQT09-RR.pdf, 2008
[5] Y. Chi, Y. Yang, Y. Xia, and R. R. Muntz. Cmtreeminer: Mining both closed and maximal frequent subtrees. In Proc. of the 8th Pacific-Asia Conf. on Knowledge Discovery and Data Mining, pages 63–73, 2004.
[6] Y. Xiao, J. F. Yao, Z. Li, and M. H. Dunham. Efficient data mining for maximal frequent subtrees. In Proc. of the 3rd IEEE Int. Conf. on Data Mining, page 379. IEEE Computer Society, 2003.
[7] World Wide Web Consortium. Extensible Markup Language(XML)1.0,1998.
http://www.w3C.org/TR/REC-xml/.

AUTHORS PROFILE

Mrs. Mopuri.Sujatha pursuing Masters in Technology in CVSR college of Engineering, Anurag Group of Institutions. Hyderabad

.

Mrs. Dhyaram Lakshmi Padmaja is working as an Associate Professor, Department of Information Technology in CVSR college of Engineering, Anurag Group of Institutions. She has 13 years of teaching experience. She is presently pursuing her Ph.D. in JNTU, Hyderabad. She is the Life Member of ISTE, IEEE. She has organized and attended various workshops and conferences National and International level.