REVIEW PAPER ON DATA MINING FOR XML QUERY ANSWERING SUPPORT

Ms. Poonam R. Maskare
Department of Computer science ,Sant Gadge Baba Amravati University Amravati,India
poonammaskare@gmail.com

Abstract
We are in an age often referred to as the information age. In this information age, Extracting information from semi structured documents is a very hard task, and is going to become more and more critical as The amount of digital information available on the internet grows. The increasing amount of XML datasets available to casual users increases the necessity of investigating techniques to extract knowledge from these data. Data mining is widely applied in the database research area in order to extract frequent correlations of values from both structured and semi structured datasets. . 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. This mined knowledge is later used to provide: (i) a concise idea – the *gist* – of both the structure and the content of the XML document and (ii) quick, approximate answers to queries. In this work we focus on the second feature. A prototype system and experimental results demonstrate the effectiveness of the approach.

Keywords- **XML, query answering support, Tree based association rule, data mining, Intentional information ,TAR extraction.**

I.INTRODUCTION

XML is a standard for describing how information is structured. XML documents form a tree structure that starts at “the root” and branches to “the leaves” which represent large amount of data. Though XML offers its users many advantages like simplicity, extensibility, interoperability; information retrieval from XML document is very difficult task. So database research field concentrates on XML as a database. User need to know structure of the document before querying the document to know the semantics which require forming query. XML documents are flexible and do not have fixed schema, so user may fail to retrieve information as answer to query.Frequent patterns of XML documents provide intentional knowledge of the document and they specify information of the document in terms of a set of properties instead of only set of data satisfying the query. Intentional answers are approximate and take less time. This knowledge is provided by XML mining tool which in terms of a set of tree based association rules. TAR provides rules in the form TB=>TH, where TB is body tree and TH is the head tree of the rule and TB is a subtree of TH. These rules are helpful for the users to get implicit information about the document and thus it will be more useful for the system in query formulation. The proposed XML query answering support framework is as shown in fig. 1. The purpose of this framework is to perform data mining on XML and obtain intentional knowledge. The intentional knowledge is also in the form of XML. This is nothing but rules with supports and confidence. In other words the result of data mining is TARs (Tree-based Association Rules).

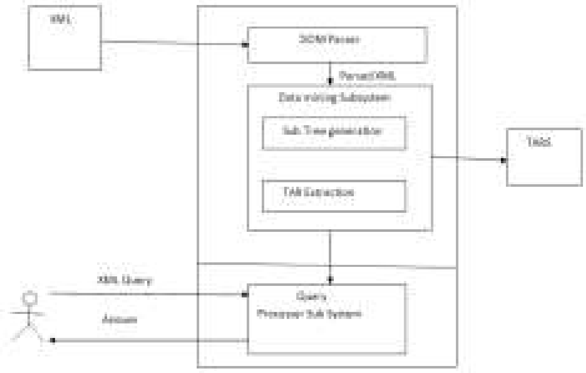


Fig.1 Proposed XML query answering support framework

As can be seen in fig. 1, the framework is to have data mining for XML query answering support. When XML file is given as input, DOM parser will parse it for well formalness and validness. If the given XML document is valid, it is parsed and loaded into a DOM object which can be navigated easily. The parsed XML file is given to data mining sub system which is responsible for sub tree generation and also TAR extraction.The generated TARs are used by Query Processor Sub System. This module takes XML query from end user and makes use of mined knowledge to answer the query quickly.

II. TREE-BASED ASSOCIATION RULES

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

$$Support = \frac{\textbf{The searched Element length}}{\textbf{Total No of length in Document}}$$

$$Confidence = \frac{\textbf{The Searched Element}}{\textbf{Total No of length in Document}}$$

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 $supp(X \cup Y)/supp(X)$. In this work we extend the notion of association rule introduced in the context of relational databases to adapt it to the hierarchical nature of XML documents. Following the Info set conventions, we represent an XML document as a tree2 _$N,E, r,$ _, c_ where N is the set of nodes, $r \in N$ is the root of the tree, E is the set of edges, _ : $N \rightarrow L$ is the label function which returns the tag of nodes (with L the domain of all tags) and $c : N \rightarrow C \cup \{\perp\}$ is the content function which returns the content of nodes (with C the domain of all contents). We consider the *element-only* Info set content model [28], where XML nonterminal tags include only other elements and/or attributes, while the text is confined to terminal elements. 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. As a consequence, for the sake of readability, we do not report the edge label and the node type label in the figures. Attributes and elements are characterized by empty circles, whereas the textual content of elements, or the value of attributes, is reported under the outgoing edge of the element or attribute it refers to.

III.TAR EXTRACTION

TAR mining is a the process composed of two steps: 1) mining frequent sub trees[1], which means sub trees with a support above a user-defined threshold, from XML document; 2)computing interesting rules, that is, rules with a confidence above a user-defined threshold , from the frequent sub trees. When the mining process has been finished and frequent TARs have been extracted, and are kept in XML format. This decision has been taken to allow the use of the same language (XQuery)[12] for analyzing both the original dataset and the finded rules. One of the reasons for using TARs instead of the original document is that processing iTARs for query answering is faster than processing the document. To take full advantage, we introduce indexes on TARs to further to increase performance for access to mined trees and in general of intentional query answering. In the literature the problem of making XML query-answering quickly by means of path-based indexes has been founded. In general, path indexes are put forword to quick answer queries that follow some frequent path template, and are built by indexing only those paths having highly frequent queries. We start from a different perspective: we want to provide a quick, and often approximate, answer also to casual

Association rules describe the frequent occurrence of data items in a large amount of data collected. A and B are the two data items. They are represented in the form of A∩B. Association rule is measured by means of Support and Confidence. Support represents the frequency of the set (A and B) found in the data set. Confidence represents the conditional probability of finding B, having got A. The interesting patterns among the subtrees of the given XML document can be identified. TAR mining is a process composed of two steps: 1) Mining frequent sub trees, that is, sub trees with a support above a user defined threshold value, from the XML document; 2) Computing interesting rules, defines finding the interesting rules that are with user-defined confidence value. The frequent pattern of subtrees had been extracted into TAR files. The TAR files are stored in XML document. These TAR files contain the rules which are computed over confidence values. Each rule is saved inside the <rule> element. These files represent the intensional knowledge about the XML document. This process of mining TAR eases the exploitation of the query-answering system. TARs are mined by generating the rules with the more number of nodes in the body tree. This reduces the complexity.

Algorithm 1 .Get-Interesting-Rules **(***D*, *minsupp*, *minconf***)**

```
// frequent subtrees
$FS$ = FindFrequentSubtrees ($D$, $minsupp$)
ruleSet = $\phi$
for all $s \in FS$ do
    // rules computed from s
    tempSet = Compute-Rules($s$, $minconf$)
    // all rules
    ruleSet = ruleSet U tempSet
end for
return ruleSet
```

Function 1 Compute-Rules **(***s*, *minconf***)**

```
ruleSet = $\phi$ ; blackList = $\phi$
for all $cs$, subtrees of $s$ do
    if $cs$ not a subtree of any element in blackList then
        $conf = supp(s) / supp(cs)$
        if $conf \geq$ minconf then
            newRule = $\{cs, s, conf, supp(s)\}$
            ruleSet = ruleSet U {newRule}
        else
            blackList = blackList U $cs$
        end if
    end if
end for
return ruleSet
```

The rules obtained from Algorithm 1 are written to an XML file. Then indexing is done to the XML file. Afterwards when XML queries are prepared, the proposed system uses index and TARs to quickly answer the query. An index [3] is created for the extracted TAR file to make fast access of the document when queries are posted. This index file is also created in XML format. This contains the set of trees and every node in the tree contains references to the rules generated. The query will be made to the original XML document. This will be automatically translated into TAR files. By using the translated TAR files, the XML documents can be queried easily compared to the other operators [3]. This is because the XQuery, the XML query language which is specifically designed for XML documents. This consists of three class queries [1] to be transformed. They are as follows:

Class 1**:** σ/π- queries**:** This query is used to bring down simple and complex operators with restrictions on them.

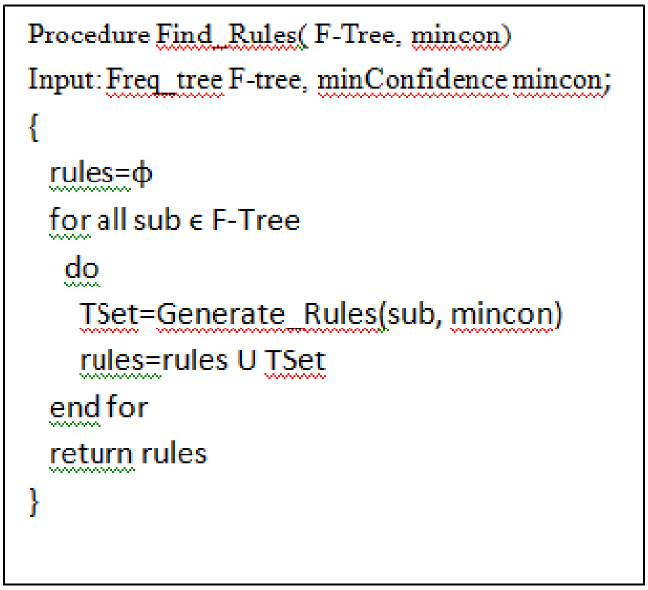
Class 2**:** count-queries**:** This query is used to count the number of elements having a specific data mentioned in the query.

Class 3**:** top-k queries**:** This query is used to select the top k queries which satisfy a grouping condition. By using these class queries, the users can pose a query over the XML document.

The users can pose a query over the XML document.

IV.INTENTIONAL AND EXTENTIONAL ANSWER:

Discovering frequent patterns from XML document provides implicit knowledge about the document which is nothing but intentional knowledge about the data contained in the document. Intentional information gives data in terms of its properties that is properties of frequent items are extracted. Query fired over the original document is converted into a query on the indexed tree based association rules. This is known as intentional query. Answer to this intentional query is intentional answer which is in fact a set of properties of the frequent items along with its support and confidence



```
Procedure Find_Rules( F-Tree, mincon)
Input: Freq_tree F-tree, minConfidence mincon;
{
  rules=ϕ
  for all sub ϵ F-Tree
    do
      TSet=Generate_Rules(sub, mincon)
      rules=rules U TSet
  end for
  return rules
}
```

As the intentional query fires on the extracted rules rather than the original document, it requires less time to calculate nswer. As well as one more advantage is that it will generate intentional answer even though the original document is not valuable or corrupted. Extensional answers are normal answers to any query fired which is in terms of set of data satisfying he query. These are just a list of data so they don't provide properties of the data. These answers are not more useful compared to intentional answers in some cases.

V.TREE RULER PROTOTYPE

TreeRuler is a tool used in our approach. When the XML document is given, it makes users to retrieve the intentional information for the queries. Users formulate XQueries over the original data, and these queries are automatically translated and executed on the intentional knowledge. Get the Gist allows intentional information extraction from an XML document, when given the supports, confidence and the files where the extracted TARs and their index are to be stored.

The tree ruler interface offers three tabs : Get the Idea: this allows showing the intentional information as well as the original document, to give users the possibilities to compare the two kinds of information. Get the Answers: it allows querying the intentional knowledge and the original XML document.

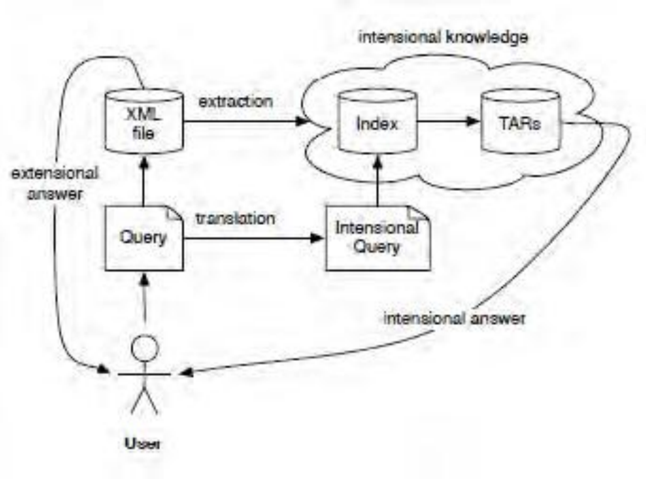


Fig.2 .Tree ruler architecture

Users have to write an extensional query. When the query belongs to the classes have analyzed, then it is translated and applied to the intentional knowledge. Finally, once the query is executed, the TARs that reflect the search criteria are shown in Fig.2 .

VI EXPERIMENTAL RESULTS

Extraction Time:

Extraction time depends on the number of nodes in xml document. Extraction time growth is almost linear with the respect to cardinality of the XML tree . Time required for the extraction of the intentional knowledge from in XML database . As no of nodes increases extraction time increase initially, it remains stable for sometimes and as no of nodes becomes too high again it increases very fast .

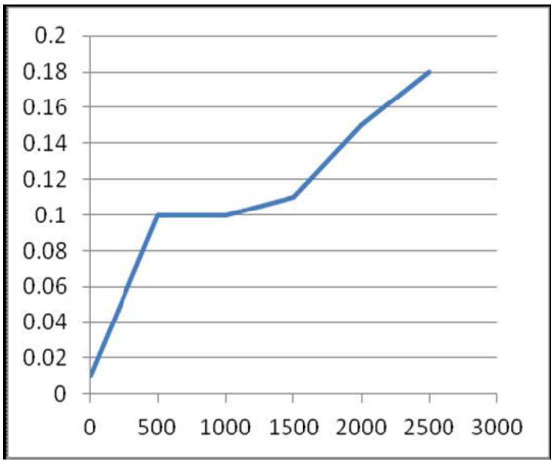


Fig.3 Extraction time w.r.t no. of nodes

Answering Time:

Answer Time of getting intentional answer is comparatively less than that of extensional answers , as instead of accessing original document mined rule file is used to answer the query. Comparison with Support and Confidence Extraction time of generating rules from XML documents changes according to support and confidence . This can be show in graph by keeping first confidence constant and vary support and then keeping support constant . It is seen in the figure3 that more the support means frequent data items are less hence extraction time is less.

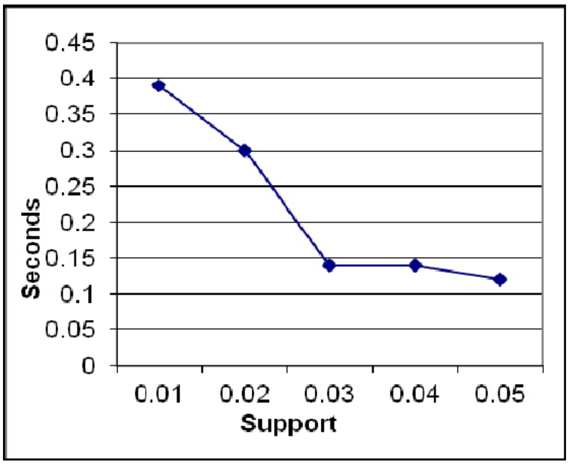


Fig.4 Extraction time with constant confidence=0.95

Similarly, confidence is more so less data items are frequently presents no. of rules extracted are less hence less time is required.

Accuracy:

Accuracy of intentional answer is measured in terms of precision and recall . Query answering depends on support threshold. When support is high then chance of correct answering is high as less no of rules are to be access

VII.CONCLUSION

Towards this end, the aim of this paper is to mine frequent association rules and store them in XML format use the TARs to support query answering or to gain information from XML databases . A prototype application is built to test the efficiency of the proposed framework. The application takes XML file as input and generates TARs and then finally index file that helps in query processing. The experimental results revealed that the proposed application is useful and can be used in real time applications. Mined all frequent association rules without imposing any restriction on the structure and the content of the rules. The proposed algorithm extends Path Based Indexing and allows users to extract efficient answering from XML documents. The main goals we have achieved are: 1) Mined frequent association rules gives the structure and the content of the XML file using tree representation; 2) Stored mined information in XML format as a consequence, 3) It can effectively use the extracted knowledge to gain information, by using query languages for XML, about the original datasets where the mining algorithm has been applied. The exact information in TARs provides a valid support in several cases. It allows obtaining and storing implicit knowledge of the documents. When compared to the Association rule the classification would increases the efficiency of query answering and time reduction in searching a document. For any kind of XML document the user can easily get the accurate answering. The aim of this project is to provide a way to use intentional knowledge as a substitute of the original document during querying and to improve the execution time of the queries over the original XML dataset. The method used in this project can be further used to optimize mining algorithms.

REFERENCES

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

[2] T. Asai, K. Abe, S. Kawasoe, H. Arimura, H.Sakamoto, and S. Arikawa. *Efficient substructure discovery from large semi-structured data*. In Proc.of the SIAM Int. Conf. on Data Mining, 2002.

[3] T. Asai, H. Arimura, T. Uno, and S. Nakano.*Discovering frequent substructures in large unordered trees*. In Technical Report DOI-TR 216, Department of Informatics, Kyushu University.

[4] E. Baralis, P. Garza, E. Quintarelli, and L. Tanca. *Answering xml queriesby means of data summaries*.ACM transactions on InformationSystems, 25(3):10,2007.

[5] D. Barbosa, L. Mignet, and P. Veltri. *Studying the xml web: Gathering statistics from an xml sample*. World Wide Web,8(4):413–438, 2005.

[6] 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.

[7] 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

[8] Mirjana Mazuran,Elisa Quintarelli,and Letiziatanca. *Optimized Data Mining for XML query answering support*. IEEETransactionson Knowledge Data Engineering, Volume:PPIssue:99,2011

[9] WorldWideWeb Consortium, Extensible Markup Language(XML)1.0,http*://www.w3C.org/TR/REC-xml/, 1998*

[10] R. Agrawal and R.Srikant. *Fast algorithms for mining association rules in large databases*. InProc. Of the 20thInt .Conf.on Very Large DataBases. Morgan Kaufmann PublishersInc.,1994.

[11] J.W.W. Wanand G.Dobbie, "*Extraction of Association rules from XML Documents Using XQuery parser*, "Proc. Fifth ACM Int Workshop Web Information and Data Management,pp.95-97,2003.

[12] J.Paik,H.Y.Youn,and U.M.Kim,"*New Method for Mining Association Rules from a Collective XML Documents*,"

[13] A.Termier, M.Rousset , M.Sebag , K.Ohara, T.Washio, and H.Motoda, "*DryadeParent: An Effective, efficient and Robust Closed Attribute algorithm for tree Mining*, "IEEETransactionon DataMining., vol.20, Pg: 301-321,Mar.2008.

[14] R.Goldmanand J.Widom, "*DataGuides : Enabling Query Formulations and Optimization techniques in Semistructured Databases*, "Proc.23rd Int Conf.on Very Large DataBases.

[15] T.Asai, H.Arimura, T.Uno, and S.Nakano. *Discovering of frequent substructures in large disordered trees*. InTechnical Report DOI-TR216, Department of Informatics, Kyushu university. http://www.i.kyushuu.ac.jp/doitr/trcs216.pdf,2003