TREE BASED ASSOCIATION RULES MINED FROM XML DOCUMENT FOR XML QUERY ANSWERING

1SWATI N. PATIL, 2R. V. MANE

1M. Tech Student, Computer Science and Technology Department of Technology, Shivaji University, Kolhapur
2Assistant Professor, Computer Science and Technology Department of Technology, Shivaji University, Kolhapur
Email; swati.patil4600@gmail.com, rvm_tech@unishivaji.ac.in

Abstract: Tree structures are used mostly in pattern recognition, XML database, computer network and so on. By considering XML database as a tree structure, one of the major problems in mining database is to find frequent subtrees. Tree based association rules are derived from frequent subtrees which provide approximate information of XML document. This information is used to answer XML query which is known as intensional answer.

Keywords: Trees, XML, Data Mining, Frequent Subtree, Tar, Xquery, Approximate Answering, Intensional Information

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 intensional 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. Intensional 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 $T_B \Rightarrow T_H$, where $T_B$ is body tree and $T_H$ is the head tree of the rule and $T_B$ is a subtree of $T_H$. 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.

II. RELATED WORK

The concept of mining TAR and applying XQuery on TAR to give an quick approximate answer was initially proposed in [1], [2]. Here, TARs were extracted and stored in XML format, so even though the original XML file is not available, user can fire a query on TAR and get an intensional answer.

Concept of extracting subtrees which maintain the parent-child relationship is mentioned in an algorithm CMTTreeMiner that extended to mine TAR from XML document [3]. To use XQuery to extract approximate answer from simple XML document [2], [3] propose a set of functions written in XQuery [4], [5]. Simple Optimization technique to optimize association rules known as Ant colony method is proposed in [6].
2) TAR:

Association rules describe the co-occurrence of data items in large amount of data of the document and are represented in the form A=>B, where A and B are two arbitrary sets of data items. The quality of association rules is measured by means of support and confidence. Support is the frequency of the set AUB in dataset, while confidence is conditional probability of finding B having found A and is given by Sup(AUB)/Sup(A).

Here, Association rules are extracted to adapt hierarchical nature of XML document and thus they are known as TAR. XML documents are converted into tree structure which gives labeled tree formulation of the XML documents.

Representation of XML document as a tree <N, E, r, l, c> where N be a set of nodes, r::N be the root of the tree, E be set of edges, l: N->l be the function which returns tag of the nodes and c be the content function which returns content of nodes. Main intension is to find relationship between subtrees of XML documents. For this consider two labeled trees P=<Np, Ep, rp, lp, cp> and Q=<Nq, Eq, rq, lq, cq>. A TAR is a tuple of the form Tr=< Tp, Tq, sTr, cTr>, where Tp and Tq are trees whose association is going to discover and sTr and cTr are support and confidence of the association rule whose values are between [0, 1] interval. Means TAR represents co-occurrence of the two trees Tp and Tq in an XML document and association rule is denoted as Tp => Tq, where Tp is called the body or antecedent of Tr and Tq is called the head or consequent of the rule Tr. From the two trees Tp and Tq, Tp is a subtree of Tq. Support STr and confidence CTr are calculated by using count and cardinality. For document D, count ( T, D) is calculated as number of occurrences of subtree T in the tree D and cardinality is calculated as number of nodes of D. So, count (T1, D) / cardinality(D) gives support of the TAR T1 => T2 and count(T1, D) / count(T2, D) gives confidence of the TAR.

There are two types of TARs for an given XML document.

1) sTAR : These are association rules which provide information only on the structure of the document. These are used to get an idea of original document.

2) iTAR : These are association rules which provide information both on structure and on the data values (PCDATA values) contained in XML document. These are used to provide intensional answer to user queries.

III. INTENTIONAL AND EXTENTIONAL ANSWER:

Discovering frequent patterns from XML document provides implicit knowledge about the document which is nothing but intensional knowledge about the data contained in the document. Intensional 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 intensional query. Answer to this intensional query is intensional answer which is in fact a set of properties of the frequent items along with its support and confidence.

As the intensional query fires on the extracted rules rather than the original document, it requires less time to calculate answer. As well as one more advantage is that it will generate intensional answer even though the original document is not available or corrupted.

Extensional answers are normal answers to any query fired which is in terms of set of data satisfying the 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 intensional answers in some cases.

IV. EXTRACTION OF TAR:

Association rules calculation from tree structure of the original XML document is very complex task. First, it is required to calculate all subtrees of the XML tree and then finding frequent subtrees. Subtrees whose support is above the threshold support value are known as frequent subtrees. Rules are extracted from these frequent subtrees whose confidence is above user defined threshold. So, TAR extraction is divided into two steps.

1) Finding frequent subtrees from the XML document.
2) Extraction of the rules by using frequent subtrees

```
Procedure Generate_Rules(sub, mincon)
Input: Subtree sub, minConfidence mincon;
{ rules=∅;
  for all SSub (subtrees of sub)
    do conf=supp(sub) / supp(SSub)
      if conf>=mincon then
        newRule={SSub, sub, conf, supp(sub)}
        Rules=Rules u (newRule)
      End if
  end for;
return rules
}
```

Rules whose confidence is above the threshold will be useful for answering a query whereas rules whose confidence is below the threshold will be discarded. Extracted rules are stored in XML format to allow the use of Xquery for querying original XML document and the mined rules. Rules <rule> has three attributes as ID, support and confidence of the rule. Since the body of the rule is a subtree of the head of the rule, a boolean attribute is used to indicate whether the node belongs to head of the tree only or belongs to the body of the rule.

XQuery are fired on TARs instead of the original XML document because it takes less time for processing TAR than processing XML document. To make it more fast TARs are indexed. A set of references are added to the each node n which gives TARs which has path from root of the tree to the node n. So, the answer to the query is the set of rules whose antecedent or consequent are same to user query.

CONCLUSION:

The main goal of this project are 1) Get an extensional answer by considering only set of data; 2) Mine all frequent association rules without imposing any a priori restriction on the structure and content of the rules; 3) store mined information in XML format; 4) optimize mined rules for improving performance; 5) use the extracted knowledge to gain information about original data sets for getting intensional answer. Out of these, consideration of changes in original XML data set to update mined TARs, optimization of mined rules and extraction of intensional answer will be implemented to complete the project and to further improve the efficiency.

REFERENCES


