Volume 6, Issue 6, June 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

TAR: Algorithm for Mining XML Query Answering

Swarupa N. Sable
ME Student, Government College of Engineering, Aurangabad,
Maharashtra, India

Abstract*— Data mining techniques extracts required information from semi structured XML document. Association rule mining method provides a tree representation for the XML structure document. Data mining is widely used in the database research in order to extract frequent interconnection of values from both structured and the semi structured datasets. The increasing amount of XML datasets available to users increases the necessity of new techniques to extract knowledge from the dataset. Here we are describing 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 the intentional knowledge used to provide: (i) Quick and approximate answers to queries and (ii) Information about structural regularities that can used as data guides for structure document querying.

Keywords: Data Mining, TAR, XML.

I. INTRODUCTION

In recent years the database research field has concentrated on XML (eXtensible Markup Language)[2] as an expressive and flexible step by step model suitable to represent large amounts of data with no perfect and fixed schema, and with a possibly non uniform and incomplete structure. The XML (eXtensible Markup Language) [2] is used in many condition of web development and very often to the simplify storage of data and also sharing. It having various functionalities which simplify the data storage, platform changes and makes the data more available thus different applications can access our data. Because of these functionalities, the usage of XML document also grows higher in the organizations and companies. There are many techniques available to extract correct information from these documents. However, these techniques were not sufficient to retrieve an efficient answering from the semi-structured XML document. It leads to answering the query in a satisfactory manner as the semi-structured document may have fluctuating structure of document and redundant data available. To recover that, an approach was introduced called the Tree Based on Association Rule, which provides a Tree representation, which makes the users to get a less approximate detail to the query given. A fast retrieval to query can be getting by the approach called Path Based Indexing mechanism. This indexing mechanism will help to visit the elements in a path manner thus it increases the speed of the getting answer. The above mechanism would take a time to get the answer for user query. Another technique called classification is applied in TAR. This will help to reduce the complexity of calculating the weight and it give an exact answering.

XML:

XML (eXtensible Markup Language) [2] is an efficient way to create common information formats and sharing of both the format and the data on the World Wide Web, intranets, and elsewhere. The XML format having a structured and a semi structured formats in the usages. XML can be used by any individual or group of individuals or companies that wants to share data in a consistent way. Thus there should be enhanced and new techniques to get exact information from the huge amount of data residing in the internet or in a specified organization. As the XML document growing in the internet may sometimes because the XML document to be a semi structured format. In the semi structured document[7] format the data retrieval will not be exact and efficient. Some of the information in the semi structured document will be null values and redundant values. This would create an unstructured format of XML document. For Example, the creation of new Account in the Gmail of various mandatory and optional requirements, consider a new user fills all the mandatory details. The optional details are filled or leave empty according to the user's wish Thus, user left some of the optional details and these details are stored in an XML format. The above processes are happening in the front end. If the back end user wants to get the detail of any of optional requirements, there could be two possibilities: some of the user's data is absent and some of the data repeats. This kind of possibility leads to a semi-structured XML Document [7]. It is a difficult process to formulate a query to get the answer from these semi structured document. This project provides a way to extract the answer from the query by using association rule mining and by indexing mechanism.

Tree Based on Association Rule(TAR):

Association rules[3]explains the 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 random sets of data items. The quality of an association rule is resolute by means of support and confidence

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

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

Here the support and confidence is calculated by length analysis in XML document Association rules [3] describe the appear with the data items in a large amount of collected data and are usually represents a simplications in the form X=>Y, where X and Y are two random sets of data items, such that X=>Y= Ø. The quality of an association rule is usually counted by the support and confidence. Support corresponds to the frequency of the sets X U Y in the dataset, while confidence corresponds to the conditional probability of finding having found X and is given by sup(XUY)/sup(X) .

Here, we changed the formula of association rule introduced in the context of relational databases as per our project requirement to adapt it to the ordered nature of XML documents. The representation of the XML document as a tree (N, E ,r, L,C) where N is the set of nodes, r is the root of the tree, E is the set of edges, L is the label function which returns the tag of nodes (with L the domain of all tags) and the content function which returns the content of nodes (with C the domain of all contents). In the Existing system, the TAR is used by element only Info set content model, where we added some functionality in the TAR of our proposed system and it retrieves the data by both element info set and path based search.

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 queries.

Intentional Procedures:

iTARs provide an exact intentional view of the content of an XML document, which is in general more concise than the extensional one because it explains the data in terms of its properties, and because only the properties that are verified by a high number of items are extracted. A user query over the original dataset can be automatically transformed into a query over the extracted iTARs. The answer will be intentional, because, rather than providing the set of data satisfying the query, the system will answer with a set of properties that these data "frequently satisfy", along with support and confidence. There are two main benefits: i) querying iTARs requires less time than querying the original XML document; ii) approximate, intentional answers are in some cases more useful than the Extensional ones. Not all queries lend themselves to being transformed into queries on iTARs ; here list three classes of queries that can be transformed by preserving the soundness; moreover, it explain how such transformation can be automatically done. The classes of queries that can be managed with this approach has been introduced and further analysed in the relational database context [13]. They include the main retrieval functionalities of XQuery [12], i.e.path expressions, FLOWR expressions, and the COUNT aggregate operator. We have not considered operators for adding new elements or attributes to the result of a query, because our purpose is to retrieve slender and approximate descriptions of the data satisfying the query, as opposed to modifying, or adding, new elements to the result. Moreover, since aggregate operators require an exact or approximate numeric value as answer, they do not admit intentional answers in the form of implications, thus queries containing aggregators other than COUNT are excluded. Note however that mined TARs allow us to provide exact answers to counting queries. The emphasized objects are meta-expressions (queries or variables) which need to be replaced in the actual query.

Class 1: s /p -queries : Used to prescribe 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 forces 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.

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 dataset.

Class 3: top-k queries: Used to select the best k answers satisfying a counting and grouping condition.

The query counts the occurrences of no 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 .

Algorithm1 shows how tree-based association rules are mined. The inputs of the algorithm are the set of frequent sub-trees , FS ,and the minimal threshold for the confidence of the rules, minconf

Algorithm1 Get-Interesting-Rules (FS, minconf)

1: ruleSet = Ø
2: for all s ∈ FS do
3: tempest = Compute-Rules (s,minconf)
4: ruleSet=ruleSet U tempSet
5: endfor
6: returnruleSet

Depending on the amount of frequent subtrees and their cardinality, the amount of generated rules may be very high. The access of the number of mined association rules w.r.t. the number of frequent itemsets occurs also in the relational context an optimization no f the basic algorithm has been Proposed in[2].Such optimization will be adapted to our XML context, for mining tree-based association rules.

Algorithm2 Compute-Rules (s, minconf)

1: ruleSet = Ø
2: for all cs subtrees of s do
3: conf = supp(s)/supp(cs)
4: if conf > minconf then
5: newRule = < cs, s, conf , supp(s) >
6: ruleSet = ruleSet U {newRule}
7: endif
8: endfor
9: return ruleSet .

In fact, prescribed a frequent sub tree S , all rules derived from that tree have the same support and different confidences. Since the confidence of a rule SB =>SH can be computed as support(SH)/support(SB) , the support of SB influences the confidence of rules having the same body tree ; the higher The support of the body tree , the lower is the confidence of the rule .

II. 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 an 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 .

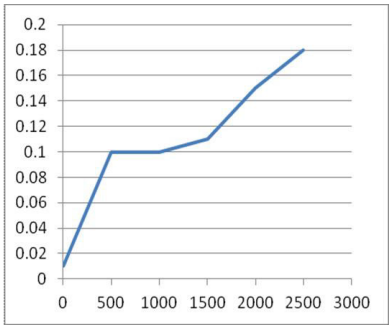

Fig. 1. Extraction time w.r.t no. of nodes

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.

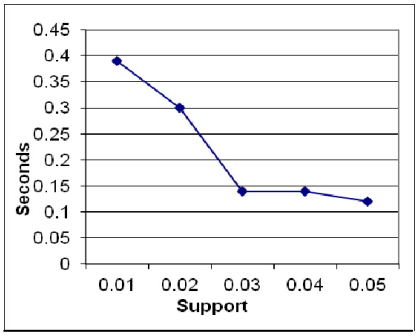

Fig. 2. Extraction time with constant confidence=0.95

Fig. 3. Extraction time with constant Support=0.02

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.

III. 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.

REFERANCES

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

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

[3] 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.

[4] 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.

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

[6] 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.

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

[8] 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

[9] A.Termier ,M.Rousset, and M.Sebag. Dryade: "An ew optimized approach for discovering closed frequent trees in heterogeneous stree databases". InProc.of the 4th IEEE Int.Conference. On knowledge and DataMining, pages544–548.

[10] K.Wang and H. Liu. Discovering typical structures of documents: a road map approaches for XML query answering support. In Proc.of the21stInt.Conf.on Research and Development in Intensional Information Retrieval ,pages145–154,1998.

[11] D.Braga, A.Campi, S.Ceri, M.Klemettinen, and P.Lanzi, "Discovering of Interesting Information in XML Data with Association Rules,"Proc. ACM Symp.Applied computing , pp.450-454, 2003

[12] World Wide Web Consortium,XQuery 1.0:An XML Query Language, http://www.w3C.org/TR/xquery,2007.

[13] E.Baralis,P.Garza,E.Quintarelli and L.Tanca,"Answering XML Queries by means of Data Summeries",ACM Trans.Information System, vol.25,no.3,p.10.2007

