MINING TREE-BASED ASSOCIATION RULES FROM XML DOCUMENTS

Saranya T.J., t.j.saranya59@gmail.com,
Dr.M.G.R.Educational & Research Institute

Abstract

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 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, intentional knowledge used to provide:
(i) Quick, approximate answers to queries and
(ii) Information about structural regularities that can be used as data guides for document querying. A prototype of the proposed system is also briefly described.

1. Introduction

In the recent years the database research held has concentrated on XML (eXtensible Markup Language) as an expressive and flexible hierarchical model suitable to represent huge amounts of data with no absolute and fixed schema, and with a possibly irregular and incomplete structure. Despite its impressive growth in popularity, XML is still lacking appropriate techniques to retrieve datasets available to casual users; such datasets, on one hand, have a limited or absent structure, and on the other hand contain a huge amount of data.

Together with intrinsically unstructured documents, there is a significant portion of XML documents which have only an implicit structure, that is, their structure has not been declared in advance, for example via a DTD or an XML- Schema http://www.w3C.org/TR/xmlschema-1/. Querying such documents is quite difficult for users for two main reasons:

1) They are not able to specify a reasonably probable structure in the query conditions and
2) They are very often confused by the large amount of information available.

This limitation of XML is a crucial problem, which did not emerge in the past years in the context of traditional (relational) database management systems, and must be addressed in order to provide access to these data to a wider set of users.

The application of data mining techniques to extract useful knowledge from XML datasets has received a lot of attention in the recent years due to the wide availability of these datasets. In particular, the process of mining association rules to provide summarized representations of XML documents has been investigated in many proposals and in particular either by using languages (e.g. XQuery) and techniques developed in the XML context, or by implementing graph/tree-based algorithms.

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. 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, like hidden integrity constraints, which can be used for semantic optimization.

2. Related Work

The problem of association rule mining was initially proposed in Agrawal (R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases) and successively many implementations of the algorithms, downloadable from B.Goethals and M.J.Zaki. Advances in frequent item set mining implementations, were developed and described in the database literature, Weka 1 being a known framework. More recently the problem has been investigated also in the XML context "Discovering interesting information in xml data with association rules", "Extracting association rules from xml documents using XQuery" and "A new method for mining association rules from a collection of xml documents". In "Discovering interesting information in xml data with association rules" the authors use XQuery (http://www.w3C.org/TR/xquery) to extract association rules from simple XML documents. They propose a set of

functions written only in XQuery which implement together the Apriori algorithm. It is show that their approach performs well on simple XML documents; however it is very difficult to apply this proposal to complex XML documents with an irregular structure. This limitation has been overcome in "Extracting association rules from xml documents using XQuery", where the authors introduce a proposal to enrich XQuery with data mining and knowledge discovery capabilities, by introducing XMINE RULE, a specific operator for mining association rules for native XML documents. They formalize the syntax and an intuitive semantics for the operator and propose some examples of complex association rules.

However, the operator proposed uses the MINE RULE operator, which works on relational data only. This means that, after a step of pruning of unnecessary information, the XML document is translated into the relational format. Moreover, both "Discovering interesting information in xml data with association rules" and "Extracting association rules from xml documents using XQuery" force the designer to specify the structure of the rule to be extracted and then to mine it, if possible. This means that the designer has to specify what should be contained in the body and head of the rule, i.e. the designer has to know the structure of the XML document in advance, and this is an unreasonable requirement when the document has not an explicit DTD. Another limitation of these approaches is that the extracted rules have a fixed root, thus once the root node of the rules to mine has been fixed, only its descendants are analyzed. Let us consider the dataset in Figure 1 to explain this consideration.

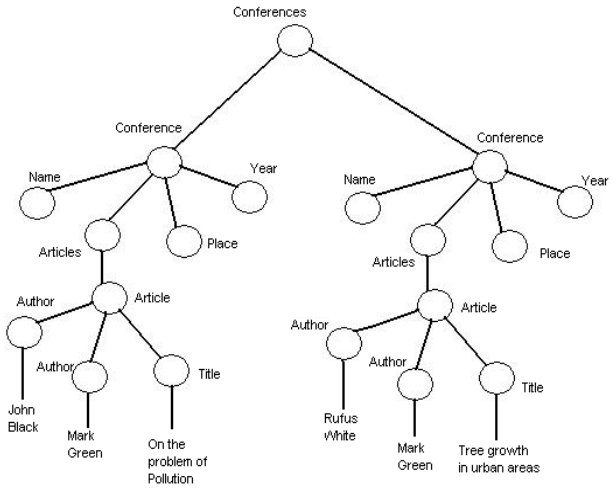

Figure 1. XML sample file: "conferences.xml"

In order to infer the co-author relationship among authors of conferences it is necessary to x the root node of the rules in the article element, the body and head in author. In such way it is possible to learn that "John Black" and "Mark Green" frequently write papers together. However, it is not possible to mine item sets stating that frequently, during "2008" conferences have been held in "Milan". Indeed, to mine such property the body of the rules should be fixed in the year element, which is not contained in the sub-tree of the article node, and the head in place.

Our idea is to take a more general approach to the problem of extracting association rules from XML documents, i.e. to mine all frequent rules, without having any a-priori knowledge of the XML dataset. A similar idea was presented in "A new method for mining association rules from a collection of xml documents" where the authors introduced HoPS, an algorithm for extracting association rules in a set of XML documents. Such rules are called XML association rules and are implications of the form X. Y, where X and Y are fragments of an XML document. In particular the two trees X and Y have to be disjunct.

The limitation of this proposal is that it does not contemplate the possibility to mine general association rules within a single XML dataset, while achieving this feature is one of our goals.

The idea of using association rules as summarized representations of XML documents was also introduced where the XML summary is based on the extraction of association rules both on the structure (schema patterns) and on content values (instance patterns) from XML datasets. The limitation of such an approach is that the so-called schema 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.

In our work, XML association rules are mined starting from frequent subtrees of the tree-based representation of a document. In the database literature it is possible to and many proposals of algorithms to extract frequent structures from tree/graph-based data structures. Just to cite some of them, Tree Miner, Path Join, CloseGraph propose algorithms to directly mine frequent itemsets-not association rules-from XML documents. Tree Miner and CloseGraph do not preserve the exact structure of the itemsets extracted -only the "descendant-of" (and not the "child-of") relationship between nodes is preserved - whereas Path Join does.

In this work we propose an algorithm that extends Path Join to mine generic tree-based association rules directly from XML documents.

3. The Treeruler Prototype

TreeRuler is a prototype tool that integrates all the functionalities proposed in our approach. Given an XML document, the tool is able to extract intensional knowledge, and allows the user to compose traditional queries as well as queries over the intensional knowledge.

Figure 2 shows the architecture of the tool. In particular, given an XML document, it is possible to extract Tree-based rules and the corresponding index le. The user formulates XQuery expressions on the data, and these queries are automatically translated in order to be executed on the intensional knowledge. The answer is given in terms of the set of Tree-based rules which reflect the search criteria.

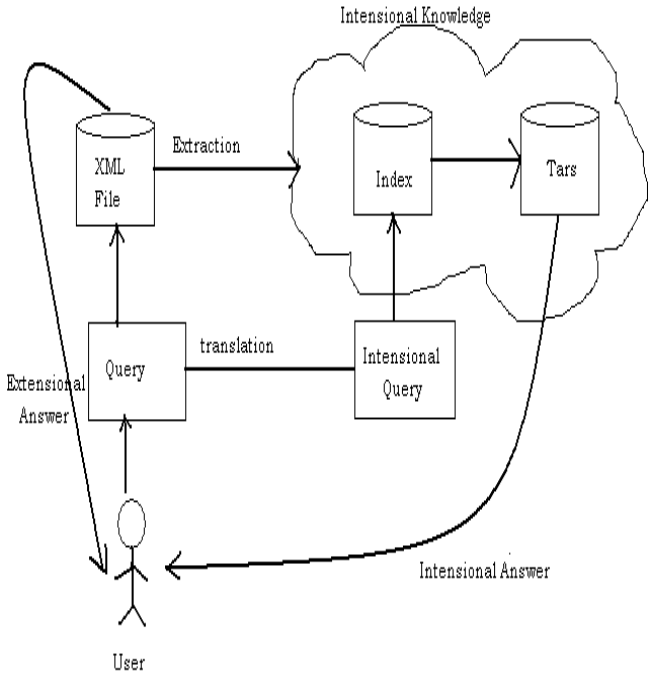

Figure 2 TreeRuler Architecture

A screenshot of the tool is shown in Figure 3: it is composed by several tabs for performing different tasks. In particular, there are three tabs:

• Get the Gist (Figure 3) allows intensional information extraction from an XML document, given the desired support, confidence and the les where the extracted tree-based rules and their index must 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 (Figure 3) 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 Tree-based rules that reflect the search criteria are shown in the box at the bottom of the form.

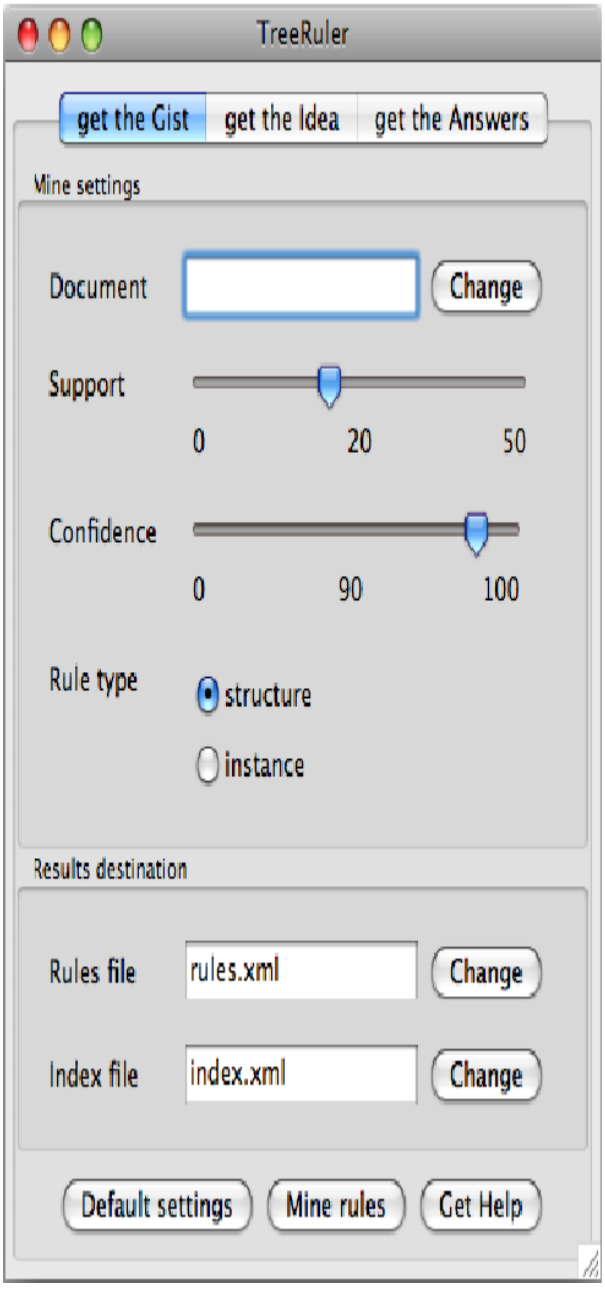

Figure 3 TreeRuler Tool

4. The Sedna Tool Prototype

Sedna is a powerful, open source, native XML Database, written from the ground up in C/C++ by Team MODIS. The team has developed and is continuing to

develop a XML Database which is starting to seriously boast the functionality and performance of mature relational databases such as MySQL and thus can be taken very seriously by application developers for production grade applications.

Virtually all other XML Databases are written in Java™, also the majority if not all of those provide a network based API which works upon SOAP, XML-RPC, REST or some other bloated protocol. Sedna's network protocol is completely binary based.

4.1Features

1.Stands up to immense usage stress, built-in Database Connection Pooling manager.

2.Allows XML documents to be streamed to Sedna directly from http:// and ftp: // locations.

3.Zero dependencies. Other than xmldb.jar interface APIs this package requires nothing other than a JVM.

4.Sedna supports the XUpdate standard for updating data.

5.Sedna can support Binary BLOBS as well as Java™ Object storage.

6.Sedna is hierarchical collections friendly.

7.Extensible, supports custom XML: DB Service plug-in on XML: DB Collections.

8.Makes full use of Sedna's ACID Transactions capability. Manual/Auto - Commit/Rollback 100% supported.

9.Very small memory footprint and very fast execution, the server carries the burden where-ever possible.

10.Via this API, Sedna can now execute an XQuery/XPath directly against a resource or a collection.

.

11.All XML processing is 100% JAXP based.

12.Meets all the requirements for XML: DB API Core Level 1 compliance.

5. References

[1] World Wide Web Consortium. Extensible Markup Language (XML) 1.0, 1998.
[2] http://www.w3C.org/TR/REC-xml/.
[3]World Wide Web Consortium. XML Schema, 2001.
[4]http://www.w3C.org/TR/xmlschema-1/
[5]R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases, pages 487{499, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.
[6]S. Amer-Yahia, S. Cho, L. V. S. Lakshmanan, and D. Srivastava. Minimization of tree pattern queries. In SIGMOD Conference, pages 497{508, 2001.
[7]T. Asai, H. Arimura, T. Uno, and S. Nakano. Discovering frequent substructures in large unordered trees, 2003.
[8]E. Baralis, P. Garza, E. Quintarelli, and L. Tanca. Answering xml queries by means of data summaries. ACM Transactions of Information Systems, 25(3):10, 2007.
[9]E. Bertino, B. Catania, and W. Q. Wang. Xjoin index: Indexing xml data for ecient handling of branching path expressions. In International Conference on Data Engineering, page 828, 2004.
[10]D. Braga, A. Campi, S. Ceri, M. Klemettinen, and P. Lanzi. Discovering interesting information in xml data with association rules. In SAC '03: Proceedings of the 2003 ACM symposium on Applied computing, pages 450{454, New York, NY, USA, 2003. ACM Press.
[11]Sedna Native XML Database System. www.sedna.org