DESIGNING TEMPLATES FOR MINING ASSOCIATION RULES FROM XML DOCUMENTS

F. H. Ismail, H. K. Mohammed
Faculty of Computer Science, Misr International Univ. Cairo, Egypt
Faculty of Engineering, Ain Shams Univ. Cairo, Egypt

M. A. Ismail, I. EL-Maddah
Faculty of Engineering, Ain Shams Univ. Cairo, Egypt

Keywords: Mining association rules from semi-structured data, meta-rule guided mining of association rules, discovering frequent structure from XML.

Abstract: Nowadays, some information is semi-structured. The main characteristic of semi-structured data (XML) is that they have irregular structure. There is no distinction between data and structure. Even though, it is quite common that semi-structured objects representing the same sort of information have similar, though not identical, structure (pattern). Previous work has introduced templates for mining association rules from XML based on prior knowledge about the structure of the XML document. If the users do not have any knowledge about the structure in advance, what would be their clue in writing templates? In this paper, we introduce a new approach for designing association rule templates based on the automatic discovery of frequent structure in the XML document. Frequent structure serves as a schema built over the semi-structured data. This layer guides the user to the useful structure that might yield useful associations rather than choosing any piece of structure at random. The structured layer is displayed from which the user can select templates of interest. Association rules that comply with the specified templates are generated.

1 INTRODUCTION

Many methods have been proposed and developed for mining association rules from relational database. However, the generated association rules could be poorly focused or lack of interest to users. According to Fu (1995 a), two major factors may contribute to this phenomenon: (1) lack of focus to the set of data to be studied, and (2) lack of constraints on the forms and/or kind of rules to be discovered. In the context of relational database, the first problem can be handled by introducing a data mining interface which specifies the set of data relevant to a particular mining task. The work introduced in Fu(1995 b) used an SQL-like interface to specify the task relevant set for a data mining query. For example, in order to find the general characteristics of computer science graduate students in Canada, a where-clause is used to retrieve only those students of interest. What about semi-structured data which is known to have irregular structure and can be populated without having a schema in advance? XML is the standard for presenting semi-structured data. X-query also is the standard XML query language (World Wide Web Consortium, 2002). How can we exploit the emerging XML query language to constrain the knowledge to be discovered to avoid the second problem mentioned above? Fing (2005) introduced an XML-enabled data mining query language XML-DMQL. The design philosophy of XML-DMQL is based on allowing the user to declare what kinds of interesting XML rules are to be mined. Users provide templates to declare their interesting knowledge. User defined templates are translated into X-query statements to constrain the XML data to be mined. Thus, the mining cost incurred is proportional to what users want. The concept of defining the rule-template by the user is not new. It comes from relational database community. However, the templates presented in Fing (2005) are

H. Ismail F., K. Mohammed H., A. Ismail M. and EL-Maddah I. (2007).
DESIGNING TEMPLATES FOR MINING ASSOCIATION RULES FROM XML DOCUMENTS.
In *Proceedings of the Third International Conference on Web Information Systems and Technologies - Web Interfaces and Applications*, pages 214-219
DOI: 10.5220/0001276802140219

formed based on the users' prior knowledge about the XML structure. In our study, we provide an automated discovery of templates that might yield useful associations. We also allow the user to select the interesting templates from the pool of useful generated templates. We also introduce how to exploit the expressiveness power of X-query to generate association rules that comply with the templates chosen by the user. In section 2 and 3, we introduce the terms of association analysis and meta-rule guided mining of association rules in relational database. In section 4 and 6, we introduce the meaning of the same terms but in the context of XML. In section 6, the proposed system design is discussed. In section 7, conclusions and future work are introduced.

2 ASSOCIATION ANALYSIS

Association analysis is the discovery of association rules showing attribute-value conditions that occur frequently together in a given set of data. Association analysis is widely used for market basket or transaction data analysis. Let I = {$I_1, I_2, \ldots\ldots I_m$} be a set of items. Let D is a set of database transactions where each transaction T is a set of items such that T $\subseteq$ I. Each transaction is associated with an identifier, called TID. Let X be a set of items. A transaction T is said to contain X if and only if X $\subseteq$ T. An association rule is an implication of the form X $\Rightarrow$ Y, where X $\subset$ I, Y $\subset$ I, and X $\cap$ Y = ϕ. The rule X $\Rightarrow$ Y holds in the transaction set D with support s, where s is the percentage of transactions in D that contain X $\cup$ Y (i.e., both X and Y). This is taken to be the probability, P(X $\cup$ Y). The rule X $\Rightarrow$ Y has confidence c in the transaction set D if c is the percentage of transactions in D containing X that also contain Y. this is taken to be the conditional probability, P(Y|X) (Han, 2001).

3 META-RULE GUIDED MINING OF ASSOCIATION RULES IN RELATIONAL DATABASE

According to (Han, 2001), Meta-rule guided mining of association rules is a rule template in the form of "$P_1 \wedge \ldots\ldots \wedge P_m \Rightarrow Q_1 \wedge \ldots\ldots \wedge Q_n$" where Pi (for i=1,…..,m) and Qj (for j=1,…,n) are predicates. Each predicate name is an attribute name of a database relation and can be instantiated by a certain value. The rule template is used to describe what forms of rules are expected to be found in the database and used as a constraint in the data mining process. However, relational databases are structured with a predefined schema. Suppose that as a market analyst, you have access to the data describing customers (such as customer age, address, and credit rating) as well as the list of customer transactions. You are interested in finding associations between customer traits and the items that customers buy. However, rather than finding all of the association rules reflecting these relationships, you are particularly interested only in determining which pairs of customer traits promote the sale of educational software. A meta-rule can be used to specify this information describing the form of rules you are interested in finding. An example of such a meta-rule is $P_1(X,Y) \wedge P_2(X,W) \Rightarrow$ buys(X,"educational software"); where P_1 and P_2 are predicate variables that are instantiated to attributes from the given database during the mining process, X is a variable representing a customer, and Y and W take on values of the attributes assigned to P_1 and P_2, respectively. Typically, a user will specify a list of attributes to be considered for instantiation with P_1 and P_2. In general, a meta-rule forms a hypothesis regarding the relationships that the user is interested in probing or confirming. The data mining system can then search for rules that match the given meta-rule. For instance, the following rule matches or complies with the above meta-rule.

Ages (X, "35-45") $\wedge$ income(X, "40-60K") $\Rightarrow$ buys(X,"educational software").

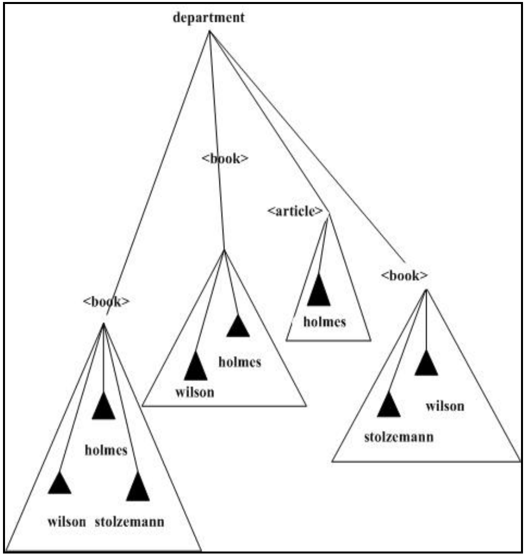

Figure 1: Research.xml.

4 ASSOCIATION RULES IN XML

(Fing, 2003; Nayak, 2005) presented an xml-enabled framework for mining association rules. The association rules in XML describe relationships between tags which tend to occur together in XML documents that can be useful in the future. In XML documents, tags and attributes contain data that attracts the user interest. The path formed by concatenating tags from the root to the current element describes the meaning of the enclosed data. To define association rules from XML documents, (Baraga, 2003) mapped the basic concepts of association rules in the XML context. As a working example, (Baraga, 2003) introduced the XML depicted in figure 1 research.xml. The problem of mining frequent association among people that appear as coauthors in the publications is considered. i.e. associations in the form "<book> <author> Wilson </author> </book> ⇒ <book> <author>Holmes </author> </book>". Since any part of an XML document is a tree (XML fragment), both D, I are sets of trees. In particular, the transactions T ∈D are the XML fragments that define the context in which the items *I* ∈ I must be counted. The items *I* ∈ I are the fragments that must be counted. If we consider the problem of mining association rules among authors appearing in the same papers, we have that D are the set of fragments of the publications. The items *I* ∈ I are the set of fragments of all the authors who appear in the works published by people within the department. This situation is depicted in figure 1 where an example of research.xml document is represented. The white triangles represent the fragments (transaction objects) corresponding to the various publications authored by people of the department. The black triangles represent the fragments corresponding to the authors who appear in various publications. The value of the <author> tag is depicted at the bottom of the black triangles thus a black triangle labeled Wilson is equivalent to the XML fragment <author> Wilson </author>. While the value of the white triangle corresponding to a <book> tag represents an entire XML fragment. In the example of figure 1, D contains the four white triangles and I contains the three black triangles corresponding to the authors Holmes, Wilson and Stolzman. The support of association rule Wilson ⇒ Holmes is 0.5 since the fragments <author> Wilson </author> and <author> Holmes </author> appear together in two white triangles out of four. We can now compute the confidence of the same rule which returns 0.66.

5 TEMPLATES FOR MINING ASSOCIATION RULES FROM XML

In XML documents, (Fing, 2003) extended the notion of an item to a tree-structured item. A tree-structured item is made up of a series of nodes (tags) that are connected to each other.

For example in figure 2, If the users are interested in associations among book titles, vdc titles and people jobs, they can form a template with antecedent predicates order/item/book/title and order/item/vcd/title and consequent predicate order/person/job.

The following association rule that complies with the template can be generated by the system:-

Order/book/title/*"startwarI"* and

ordre/vcd/title/*"starwarI"* ⇒

order/person/job/*"student"*. The support of this rule is 2/3 as well as its confidence.

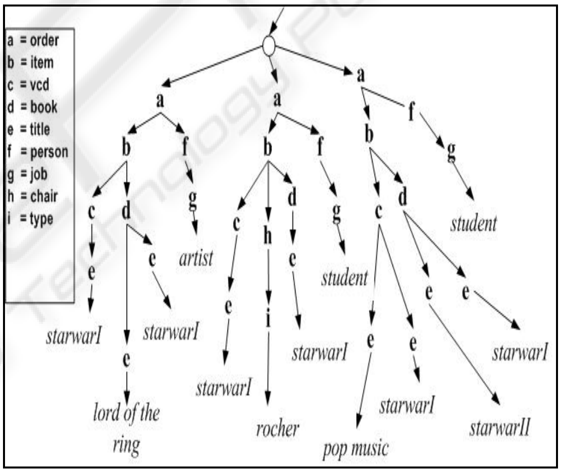

Figure 2: Orders.xml.

6 PROPOSED SYSTEM DESIGN

Our proposed system design contains two main steps. The first step is discovering frequent structure. The second step is generating association rules. The users can interact with the system by constraining the data to be mined and by choosing a template from the automatically generated ones. The following stages describe the system design.

6.1 Discovering Frequent Structure

(Maruyama, 2000) presented an Apriori-like algorithm to mine frequent substructures based on the "downward closure" property. They regarded the XML tree as a set of tree expressions. Let p_i denotes path expression which is the concatenation of tags from root node to leaf node. A *k-tree* expression is a tree expression containing *k* leaf nodes. They first found the frequent *1-tree-expressions* that are frequent individual *label path*s. Discovered frequent *1-tree-expression*s are joined to generate candidate *2-tree-expression*s. The process is executed iteratively till no candidate *k-tree-expression*s is generated. However, Apriori is one of the most widely used algorithms in data mining applications (Agrawal, 1994). This is a consequence of its good performance when minimal support is high. However, we introduce a pruning algorithm based on pushing the support constraint into the selection of *1-tree-expressions* rather than counting all *1-tree expressions.* We also break up the black box of the mining process to provide user interaction in an early stage. Figure 3 shows the system components.

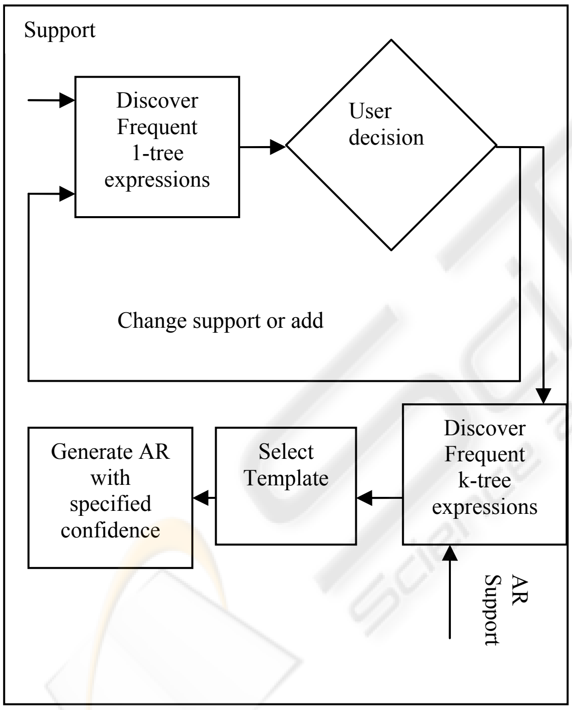
Support
Discover Frequent 1-tree expressions
User decision
Change support or add
Discover Frequent k-tree expressions
Select Template
Generate AR with specified confidence
AR Support

Figure 3: Proposed system design.

6.1.1 Discovering Frequent 1-tree-Expressions

For each tag on a distinct path (identified uniquely by the list of tags/nodes from root to the tag being counted), the count of its appearances is computed (for example, tag book appears 3 times on path order/item/book).

An optimization technique can be adopted, based on the knowledge that an ancestor is a superset of its descendants: the search avoids examining paths containing any tags whose ancestors do not have minimum support. For each tag, the ratio of its count in each distinct path to the total number of transaction objects is calculated. If the ratio is less than a certain threshold value, this tag and its descendants will be pruned. The search is simplified as a result of pushing the support threshold into the discovery process. For example, the support of path order/person/job is 3/3, while the support of order/item/chair is 1/3. For a threshold value greater than 40%, the path order/item/chair and all its descendants will be pruned.

Each tag on a distinct path that passes the support test will have a unique identifier and its full path will be saved as a description on a table. Let the identifier of the frequent path "order/item/vcd/title" is P1 and its description is its path. For a threshold value greater than 40%, table 1 will be constructed. P_1 is an attribute describing the recorded orders. Its support means that at least 40% of orders contain the job of the persons issued that order.

6.1.2 Constraining the Data to be Mined

Upon discovering the 1-tree-expressions, the user is given the chance to view them. This step is optional. The importance of this step is to allow the user to change the support value before the mining process starts. Additionally, the user can get insight into the data. For example, this stage can supply the user with information about the most frequently repeated structure. For example, The user might want to constrain the search space to the orders issued by *students* only. A where-clause expressed in X-query can be used to retrieve the data of interest. This step goes forward to open up the black box of the mining process and to allow user interaction. If the user changed the support value or chose to restrict the search space, the discovery process would be executed from the start again. Each transaction object id contains frequent paths will be recorded along with paths identifiers. For example order1 (ID=1) contains the frequent set P1, P2, P3.

Table1: Frequent 1-tree expressions.

Identifier	description
P1	order/person/job
P2	order/item/book/title
P3	order/Item/vcd/title

6.1.3 Generating k-tree-Expressions

(Pavón, 2006) presented an algorithm named MATRIX APRIORI, which incorporates the positive characteristics from Apriori and FP-growth. Matrix Apriori offers a simpler and more efficient solution for the process of mining association rules than previous proposals. The details of implementation are mentioned in (Pavón, 2006). The generated *frequent 1-tree-expressions* are passed as an input. The output is frequent patterns that pass the minimum support test. The set of frequent paths are displayed to the user. The users can drag the paths of interest to form the antecedent and the consequent templates and can supply the support and confidence of the rule as well. Frequent structure might yield useful associations while infrequent structure will never yield useful ones. The user can select a template from the pool of frequent structure.

6.2 Generating Association Rules

In relational database, each predicate name is an attribute name and can be instantiated by a certain value. In terms of XML, a predicate is the full path describing data. For example, "*student*" is the value of attribute order/person/job. The only relation we have now is the structured layer (set of frequent paths). The attributes of the frequent structure can guide the user to form an interesting template. The user can supply the template, the rule support and the rule confidence. Here, the user introduced two support values, one for structure and the other for association rules support (AR support.). The support of structure is the frequency of paths relative to the whole number of transaction objects. While the support of AR is the frequency of a certain attribute value relative to the transaction objects containing this attribute. For example, suppose 50% of the whole number of orders contains the job of the customer. If the database contains 4 orders, it means that 2 orders only contain the job. Assume the two orders had "*student*" as a value. The support of "*student*" will be 100%. The question now is how to derive associations that comply with the chosen templates? The answer is that we can apply the algorithm of meta-rule guided mining of single variable rules introduced in (Fu, 1995 a). (Fu, 1995 a) borrowed the concept of Apriori presented by (Agrawal, 1994). In a nutshell, the count of each distinct value of each predicate is retrieved using (DISTINCT, COUNT) X-query statements. Each frequent predicate values can be joined to generate more frequent candidates. Association rules that satisfy the supplied confidence are generated.

7 CONCLUSIONS AND FUTURE WORK

As the emerging standard for semi-structured data (XML) has been widely used, the need to mining XML becomes important. In this study, we proposed a method for designing templates for mining association rules from XML. (Fing, 2004) proposed templates based on user's prior knowledge about XML structure specifications. Our design differs in that we don't assume users' knowledge because of the irregular structure of XML. Discovering frequent structure that satisfies the user support is the first stage. To implement this stage, we applied an Apriori-Matrix algorithm (Pavón, 2006) which outperforms Aprioir and FP growth. While mining frequent structure, we allowed the user exploration of data to be able to define constrains on the mined data even before the whole mining process starts. X-query can be used to constraint the XML data without the need to convert the whole xml into relational database. Frequent paths serve as a structured layer over the semi-structured data. Each one of its paths can be considered as an attribute. The users can select the paths of interest to define the rule template. Now, the process of discovering association rules can take the same direction of (Fu, 1995 a) in the context of relational database. There are many open issues related to the implementation. Highly friendly user interface is required to encourage the user intervention. Also, tightly coupling the implementation with X-query is a step forward towards coupling the data mining process with database servers. The idea of integrating the data mining and data warehouse comes from relational database world. (Chaudhuri, 1998) mentioned the advantages of such integration. Our work can be used as an interface to the proposed XML-DMQL (Fing, 2005).

REFERENCES

Agrawal, R. and Srikant, R., 1994. Fast algorithms for mining association rules in large databases. In *Proceedings of 20th International Conference on Very Large Data Bases*, Santiago, Chile, September 12-15. pages 487–499.

Baraga, D., Campi, A., Ceri, S., Klemettinen, M. and Lanza, P., 2003. Discovering interesting information in XML data with association rules. In *Proc. Of the 18th Symposium on Applied Computing*, Florida, USA, March, pp.450-454.

Chaudhuri, S., 1998. Data Mining and Database Systems: Where is the Intersection? In *Data Engineering Bulletin*, 21(1):4–8.

Feng, L. and Dillon, T., Weigand, H. and Chang, E., 2003. An xml-enabled association rule framework. In *Proc. Of the 14th Intl. Conf. on Database and Expert Systems Applications*, Prague, Czech Republic, September, pp.88-97.

Feng, L. and Dillon, T., 2004. Mining xml-enabled association rules with templates. In *Proc. Of the Intl. Workshop on Knowledge Discovery in Inductive Databases*, Pisa, Italy, September, pp.61-72.

Feng, L. and Dillon, T., 2005. an XML-enabled data mining query language XML-DMQL. In *Int. J. Business and Data Mining*, Vol. 1, No. 1, pp. 22-41.

Fu, Y. and Han, J., 1995 a. Meta-rule-guided mining of association rules in relational databases. In *Proc. 1st Int'l Workshop on Integration of Knowledge Discovery with Deductive and Object-Oriented Databases*, Singapore, Dec, pages 39-46.

Fu, Y. and Han, J., 1995 b. Exploration of the Power of Attribute Oriented Induction in Data Mining. In *Advances in Knowledge Discovery and Data Mining*.

Han, J. and Kamber, M., 2001. *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, San Francisco, California, USA.

Maruyama, K. and Uehara, K., 2000. Mining association rules from semi-structured data. In *Proc. Of the ICDCS Workshop of Knowledge Discovery and Data Mining in the World-Wide Web*, April, Taipei, Taiwan.

Nayak, R., 2005. Discovering Knowledge from XML Documents. In *Wong, John, Eds. Encyclopedia of Data Warehousing and Mining*, Idea Group Publications.

Pavón, J., Viana, S. and Gómez, S., 2006. Matrix Apriori: Speeding Up the Search for Frequent Patterns**.** In *Proc. Of the 24th IASTED International Mukti-conference on Database and Applications*, Innsbruck, Austria.

World Wide Web Consortium, 2002. X*Query 1.0: An XML Query Language*, April, http://www.w3.org/TR/xquery-operators/.