International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:9, 2008

A New Model for Discovering XML Association Rules from XML Documents

R. AliMohammadzadeh, M. Rahgozar, and A. Zarnani

***Abstract*—**The inherent flexibilities of XML in both structure and semantics makes mining from XML data a complex task with more challenges compared to traditional association rule mining in relational databases. In this paper, we propose a new model for the effective extraction of generalized association rules form a XML document collection. We directly use frequent subtree mining techniques in the discovery process and do not ignore the tree structure of data in the final rules. The frequent subtrees based on the user provided support are split to complement subtrees to form the rules. We explain our model within multi-steps from data preparation to rule generation.

***Keywords*—**XML, Data Mining, Association Rule Mining.

I. Introduction

DATA mining is usually used to extract interesting knowledge from large amounts of data stored in databases or data warehouses. This knowledge can be represented in many different ways such as clusters, decision trees, decision rules, etc. Among them, association rules have been proved effective in discovering interesting relations in massive amounts of data.

Currently, XML is penetrating virtually all areas of Internet application programming and is bringing about huge amount of data encoded in XML. With the continuous growth in XML data sources, the ability to extract knowledge from them for decision support becomes increasingly important and desirable [3]. Due to the inherent flexibilities of XML, in both structure and semantics, mining knowledge in the XML Era is faced with more challenges than in the traditional structured world.

In this paper, we propose a new model for mining association rules from XML documents. Our approach extracts all possible association rules from one or multiple XML documents based on user provided support and confidence measures. Optionally, our model can take advantage of a user provided rule template to prune the search space and extract association rules that the user is interested in. In our approach, frequent subtree mining techniques are used to mine the XML documents.

R. AliMohammadzadeh and A. Zarnani are with the Database Research Group, faculty of ECE, School of Engineering, University of Tehran, Tehran, Iran (e-mail: r.mohammadzadeh@ece.ut.ac.ir, a.zarnani@ece.ut.ac.ir).

M. Rahgozar is with the Control and Intelligent Processing Center of Excellence, faculty of ECE, School of Engineering, University of Tehran, Tehran, Iran (e-mail: rahgozar@ut.ac.ir).

The paper is organized as follows. We briefly introduce XML association rules in Section II. Section III reviews some works in frequent subtree mining area. In section IV we explain our model within multi-steps from data preparation to rule generation. Finally, in the last section, we conclude the paper and propose some suggestions for future works.

II. XML Association Rules

Association rules were first introduced by Agrawal et al. to analyze customer habits in retail databases. Association rule is an implication of the form $X \Rightarrow Y$, where the rule *body* X and *head* Y are subsets of the set I of *items* ($I = \{I1, I2, \ldots, In\}$) within a set of *transactions* D and $X \cap Y = \varphi$. A rule $X \Rightarrow Y$ states that the transactions T that contain the items in X are *likely* to contain also the items in Y. Association rules are characterized by two measures: the *support*, which measures the percentage of transactions in D that contain both items X and Y; the *confidence*, which measures the percentage of transactions in D containing the items X that also contain the items Y [Fig. 1]. In XML context, both D and I are collections of trees [1], in the same way X and Y are XML fragments [Fig. 2].

Bread ➔ Milk
[Support = 2%, Confidence = 95%]
Fig. 1 Association rule between bread and milk

<Author> Rakesh Agrawal </author>
↓
<Keyword>Data Mining</keyword>
Fig. 2 XML Association rule

III. Frequent Subtree Mining

Mining frequent subtrees has many practical applications in areas such as computer networks, Web mining, bioinformatics, XML document mining, etc [5, 6]. These applications share a requirement for the more expressive power of labeled trees to capture the complex relations among data entities. Frequent subtree mining is a more complex task compared to frequent item-set mining. However most of existing frequent subtree mining algorithms borrows techniques from the relatively mature association rule mining area [7, 8]. So far, many algorithms have been developed for mining frequent subtrees from a collection of trees. We developed W3-Miner [9] for discovering weighted embedded

subtrees from a collection of trees. In [5, 6 and 10] M.J. Zaki presented an algorithm, TreeMiner, to discover all frequent embedded subtrees, i.e., those subtrees that preserve ancestor-descendant relationships, in a forest or a database of rooted ordered trees.

This algorithm used a new data structure, scope-list, to efficiently count the frequency of candidate subtrees. The algorithm was further extended in [11] to build a structural classifier for XML data. Asai *et al.* in [12] presented an algorithm, FREQT, to find frequent rooted ordered subtrees. Also two algorithms were proposed by Asai et al. and Yun Chi et al. to mine rooted unordered subtrees, based on enumeration graph and enumeration tree data structures [13, 14]. Another work has been done in [15] where a model-validating approach for non-redundant candidate generation has been proposed. Wang and Liu [16] developed an algorithm to mine frequently occurring subtrees in XML documents. They mine induced subtrees only.

There are several other recent algorithms that mine different types of tree patterns, which include FreeTreeMiner [17] which mines induced, unordered, free trees (i.e., there is no distinct root); FreeTreeMiner for graphs [18] for extracting free trees in a graph database; and PathJoin [19], uFreqt [20], uNot [13], and HybridTreeMiner [21], which mine induced, unordered trees. TreeFinder [22] uses an Inductive Logic Programming approach to mine unordered, embedded subtrees, but it is not a complete method, i.e., it can miss many frequent subtrees, especially as support is lowered or when the different trees in the database have common node labels. SingleTreeMining [23] is another algorithm for mining rooted, unordered trees, with application to phylogenetic tree pattern mining. Recently, XSpanner [24], a pattern growth-based method, has been proposed for mining embedded ordered subtrees.

In our model, we use FSM techniques for mining generalized association rules from xml documents. Following section elaborate more details about our approach.

IV. XML Association Rule Mining

There is a large body of work in the field of mining association rules from XML data. Braga *et. al* [1] proposed XMINE Operator for this task. This operator is based on XPath (XQuery) and allows the user to specify the template of the desired rules. Then, XMINE transforms the XML data in hand to the relational transactions format. Hence, the common algorithms of frequent itemset mining can be easily applied on the new format. Low performance is of the main shortcomings of XMINE, as the authors also point. Also the tree structure of the data, which is of high importance in XML, is ignored in the discovery process and in the extracted rules as a result. Finally, XMINE can not discover all of the possible rules (generalized XML association rules) as it is depended on the user-provided rule template.

In [26] Ding *et. al* developed a method to discover all of the possible rules, i.e. generalized association rules from XML documents. In this method, all of the possible combinations of XML nodes based on their multiple nesting are used to generate the relational transactions format. This method suffers form some shortcomings including generation of redundant rules [26]. Like XMINE, this method ignores the valuable tree structure of the documents too.

Of other researches in the area of XML association rule mining are [27] and [28]. The proposed methods solve some of the shortcomings seen in the previous approaches. The two papers use almost the same idea that only covers the discovery of rules that have a very restricted tree structure. The extracted rules can only have one node in the antecedent and descendent parts.

Considering the problem of mining generalized XML association rules from XML documents and the mentioned problems of the previously proposed methods, we propose a new approach for the extraction of all possible association rules from one or multiple XML documents based on user provided support and confidence measures. Optionally, our model can take advantage of a user provided rule template to prune the search space and extract association rules that the user is interested in.

In our approach, frequent subtree mining techniques are used to mine the XML documents. Our method differs from the other works in this area in two folds: First, We find XML or tree association rules. By tree association rule we mean that supposing an association rule in a form of $X \rightarrow Y$ then both X and Y are subsets of nodes with their corresponding tree structure. Second, we do not necessarily use any restricting template or rule structure pattern in contrast to [1] and [26], respectively. Therefore, our method can find all possible association rules with respect to the user provided support and confidence measures. Our model consists of several steps, as can be seen in figure 3. In the following we elaborate on each of the steps in our model.

At the first step, the user selects the XML documents subject to search in the system. Optionally, the user can define a rule template for the rules that he/she is interested in. This template determines the desired subset of the collection that is used extract the rules and is in the form of $X \rightarrow Y$, where X and Y are both the specified XML fragments. The interested reader can refer to [25] or [2] for more details. If user provides a rule template then XML documents are filtered so that unrelated XML fragments will be removed. Otherwise, the whole XML document is passed to the next steps.

After selecting the required sections of XML documents, in the next step, we convert the XML documents to a canonical string representation [7], which is usually used in most of tree mining algorithms. By using this canonical form, we can use frequent subtree mining algorithms like w3-miner [9] for extracting frequent subtrees from XML document.

We traverse XML documents in preorder manner and generate the canonical string representation in the following way: After reaching to a new node or new content we extract and encode all tokens. We use traditional IR methods including stemming and stop-word removal to increase the quality of the data source (the XML documents). In addition,

in this step, we join sibling nodes with same label and generate a new unique node. Joining the nodes with same labels can affect the tree mining process in the next steps.

After converting the XML documents to the canonical string representation, all frequent subtrees (with respect to the user provided support measure) are extracted. By default, we use w3-miner in this step but we can use any other tree mining algorithm with small modification to the current model. The fact that XML documents have hierarchical and tree structure leads us to use frequent subtree mining techniques for mining association rule mining from XML documents.

We model each XML document as a rooted unordered tree. Thus, the XML document collection creates a forest. By applying w3-miner on this forest, some frequent subtrees will be extracted (with respect to the predefined support measure). These subtrees will be used for the generation of the final association rules in the next step. The idea of generating the XML association rules from the discovered frequent subtrees is somehow similar to the way that traditional itemset mining works. In traditional itemset mining methods, at first the large itemsets are extracted and then these large itemsets are partitioned to all of the possible two subsets that construct the two sides of the association rules. However, there is no unique way to partition a tree to two subtrees (subset of the frequent tree). The method that we use to partition a discovered frequent tree is as follows: By considering all bottom-up subtrees in a frequent tree, we disconnect each of these subtrees and put it on the right side of the rule. The left side consists of the remaining subtree. Figure 4 shows an example of this partitioning method. After constructing the rule, we must compute its confidence. For this purpose we use tree matching algorithms for finding the subtree in the left hand of rule and we count all occurrences of this subtree in the forest. Then we search for the subtree in the right hand of rule and we count the number of co-occurrences with left hand side of the rule. This means the subtree in the left side of the rule occurs with the one in the right side in a way that they form the initial discovered frequent tree. The confidence of a rule is equal to division of these two numbers. Of course in each partitioned subtree is tested as the antecedent and the descendent. The last step is converting the extracted rules to XML association rules by a simple post-processing algorithm.

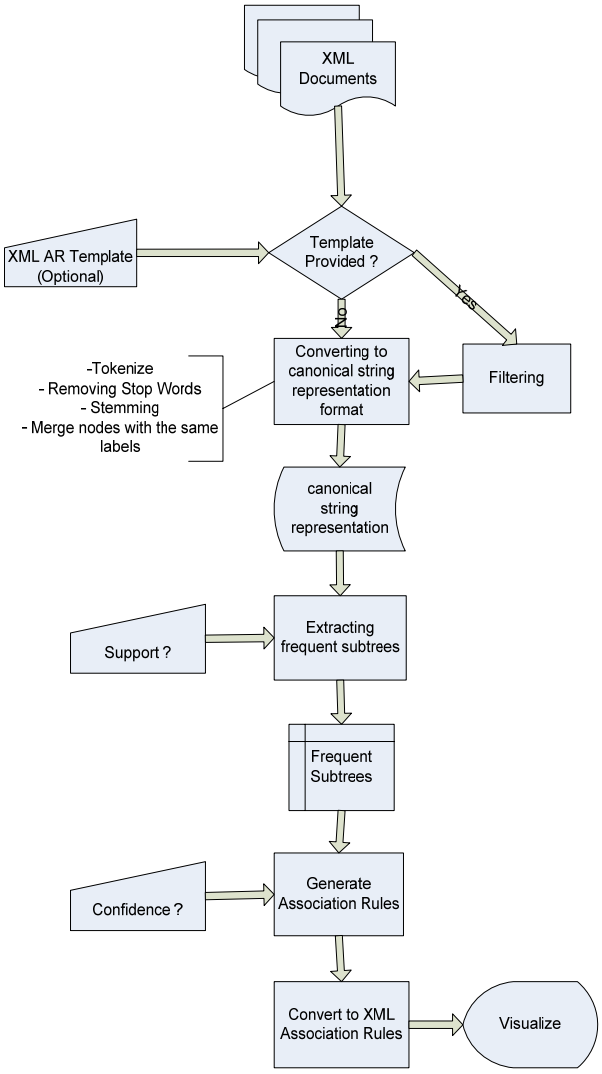

Fig. 3 Association Rule Mining from XML documents

We can use the tree structure in Fig. 5 to illustrate our model on a sample XML database. For instance, the root node Conference in Fig. 5. A relates to the root element in the XML structures, ConfID to the inner node, and City to the most inner node.

Although the structures of the trees in Fig. 5 vary so much, we still can group them into two groups since the tree t1 in Fig. 6 is commonly included by tree A, B, C, and D in Fig. 5 and the tree t2 in Fig. 6 are commonly included by tree E, F, and G. We call t1 and t2 frequent trees corresponding to the input trees of Fig. 5. Based on these frequent subtrees, 2 association rules are displayed in Fig. 7.

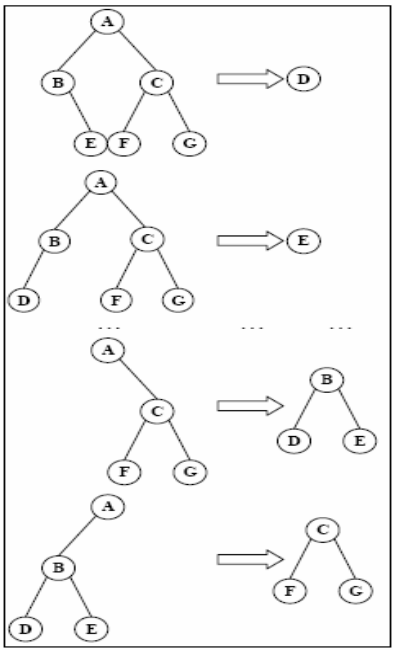

Fig. 4 Generating Association Rules from Frequent Subtrees

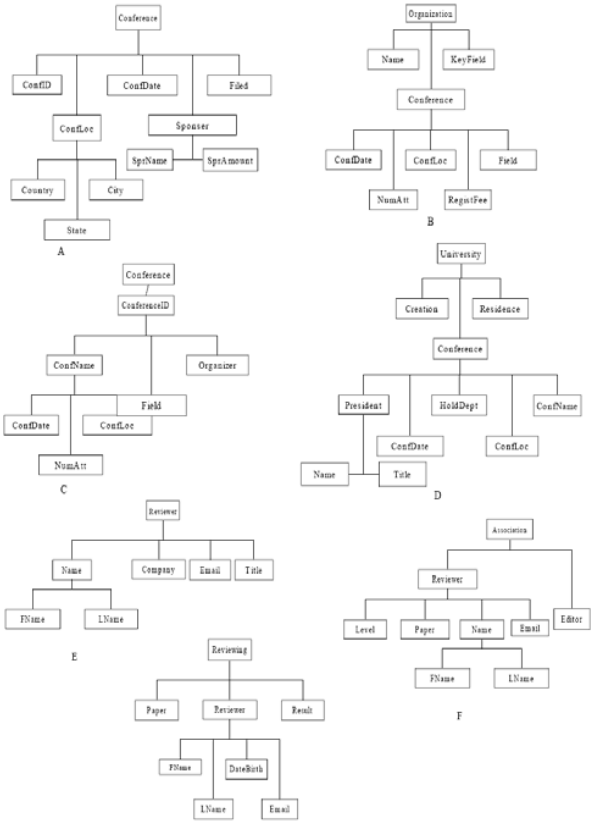

Fig. 5 Sample Tree Dataset

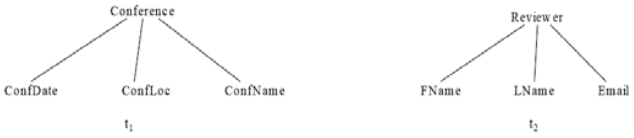

Fig. 6 Frequent subtrees with (support = 3)

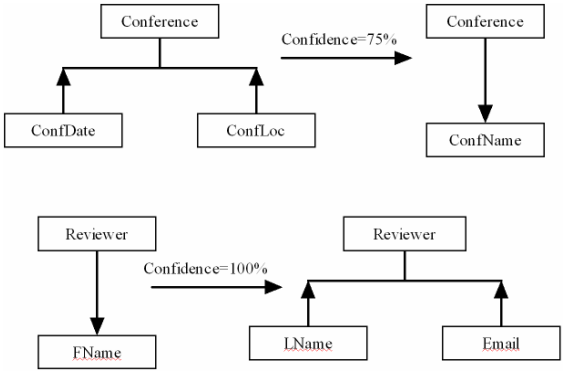

Fig. 7 Sample Association Rules

V. Conclusion

We proposed a new model for the discovery of XML association rules from a collection of XML documents. Our method does not use any relational transformation as it applies frequent tree mining algorithms directly on the XML documents for the extraction of rules. Also in contrast to the previously proposed methods, our approach will discover all of the possible rules because it does not use any restrictions on the rule structures in the discovery process. We plan to experimentally study the performance of our model on a real world XML collection with larger size and improve the performance of the confidence calculation algorithm used, in our future works.

References

[1] Braga D., A. Campi, M. Klemettinen, and P. L. Lanzi. Mining association rules from XML data. In Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery, September 4-6, Aixen-Provence, France 2002.

[2] Feng L. & T. Dillon. Mining XML-Enabled Association Rule with Templates. In Proceedings of KDID04, 2004.

[3] Nayak, R. Discovering Knowledge from XML Documents, in Wong, John, Eds. Encyclopedia of Data Warehousing and Mining. Idea Group Publications, 2005.

[4] Tan, H., T.S. Dillon, L. Feng, E. Chang, F. Hadzic, "X3-Miner: Mining Patterns from XML Database," In Proc. Data Mining '05. Skiathos, Greece, 2005.

[5] M. .J. Zaki, "Efficiently Mining Frequent Trees in a Forest: Algorithms and Applications," in *IEEE Transaction on Knowledge and Data Engineering*, vol. 17, no. 8, pp. 1021-1035, 2005.

[6] M. J. Zaki,.. "Efficient Mining of Trees in the Forest". SIGKDD '02, Edmonton, Alberta, Canada, ACM. 2002.

[7] Y. Chi, S. Nijssen, R.R. Muntz, J. N. Kok, "Frequent Subtree Mining An Overview," Fundamental Informatics, Special Issue on Graph and Tree Mining, 2005.

[8] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. Inkeri Verkamo, "Fast Discovery of Association Rules," Advances in Knowledge Discovery, and Data Mining, U. Fayyad et al., eds.,pp. 307-328, Menlo Park, Calif.: AAAI Press, 1996.

[9] R. AliMohammadzadeh, M. Haghir Chehreghani, A. Zarnani, M. Rahgozar, "W3-Miner: Mining Weighted Frequent Subtree Patterns in a Collection of Trees". In Proceedings of the Second International Conference on Pattern Analysis (Budapest, Hungary, May 26-28, 2006). ICPA'06. Transaction on Engineering, Computing and Technology, ISSN 1305-5313, Pages 164-168, World Enformatika Society.
[10] M. Zaki. Efficiently mining frequent embedded unordered trees. Fundamental Informatics, 65:1-20, 2005.
[11] M. J. Zaki and C. C. Aggarwal. XRules: An effective structural classifier for XML data. In *Proc. of the 2003 Int. Conf. Knowledge Discovery and Data Mining*, 2003.
[12] K. Abe, S. Kawasoe, T. Asai, H. Arimura, and S. Arikawa, "Optimized Substructure Discovery for Semi-structured Data," *In Proc. PKDD'02*, 1–14, LNAI 2431, *2002*.
[13] T. Asai, H. Arimura, T. Uno, and S. Nakano. Discovering frequent substructures in large unordered trees. In *Proc. of the 6th Intl. Conf. on Discovery Science*, 2003.
[14] Y. Chi, Y. Yang, and R. R. Muntz. Mining frequent rooted trees and free trees using canonical forms. Technical Report CSD-TR No. 030043, UCLA, 2003.
[15] H. Tan, T.S. Dillon, L. Feng, E. Chang, F. Hadzic, "X3-Miner: Mining Patterns from XML Database*," In Proc. Data Mining '05. Skiathos, Greece*, 2005.
[16] K. Wang and H. Liu, "Discovering Typical Structures of Documents: A Road Map Approach," Proc. ACM SIGIR Conf. Information Retrieval, 1998.
[17] Y. Chi, Y. Yang, and R.R. Muntz, "Indexing and Mining Free Trees," Proc. Third IEEE Int'l Conf. Data Mining, 2003.
[18] U. Ruckert and S. Kramer, "Frequent Free Tree Discovery in Graph Data," Special Track on Data Mining, Proc. ACM Symp. Applied Computing, 2004.
[19] Y. Xiao, J.-F. Yao, Z. Li, and M.H. Dunham, "Efficient Data Mining for Maximal Frequent Subtrees," Proc. Int'l Conf. Data Mining, 2003.
[20] S. Nijssen and J.N. Kok, "Efficient Discovery of Frequent Unordered Trees," Proc. First Int'l Workshop Mining Graphs, Trees, and Sequences, 2003.
[21] Y. Chi, Y. Yang, and R.R. Muntz, "HybridTreeMiner: An Efficient Algorihtm for Mining Frequent Rooted Trees and Free Trees Using Canonical Forms," Proc. 16th Int'l Conf. Scientific and Statistical Database Management, 2004.
[22] A. Termier, M-C. Rousset, and M. Sebag, "Treefinder: A First Step Towards XML Data Mining," Proc. IEEE Int'l Conf. Data Mining, 2002.
[23] D. Shasha, J. Wang, and S. Zhang, "Unordered Tree Mining with Applications to Phylogeny," Proc. Int'l Conf. Data Eng., 2004.
[24] C. Wang, M. Hong, J. Pei, H. Zhou, W. Wang, and B. Shi, "Efficient Pattern-Growth Methods for Frequent Tree Pattern Mining," Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining, 2004.
[25] R. AliMohammadzadeh, S. Soltan, and M. Rahgozar, "Template guided association rule mining from XML documents". In Proceedings of the 15th international Conference on World Wide Web (Edinburgh, Scotland, May 23 - 26, 2006). WWW 2006, ACM Press, New York, NY, 963-964. DOI= http://doi.acm.org/10.1145/1135777.1135966.
[26] Q Ding, K Ricords, J Lumpkin, "Deriving General Association Rules from XML Data", In Proceedings of Fourth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'03) October 16-18, 2003 Lübeck, Germany.
[27] YL Chen, CH Ye, SY Wu, "Mining Predecessor–Successor Rules from DAG Data", International Journal of Intelligent Systems, 2006.
[28] C. Combi, B. Oliboni, R. Rossato. "Complex Association Rules for XML Documents". In Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES05).