Valuable Association Rules Extraction from XML-Based Tree Data*

Juryon Paik', Junghyun Nam[e], Ung Mo Kim', and Dongho Won[*"]

Department of Computer Engineering, Sungkyunkwan University,
300 Cheoncheon-dong, Jangan-gu, Suwon-si, Gyeonggi-do 440-746, Korea
{wise96, umkim} @ece.skku.ac.kr, dhwon@security.re.kr
[2] Department of Computer Engineering, Konkuk University,
322 Danwol-dong, Chungju-si, Chungcheongbuk-do 380-701, Korea
jhnam@kku.ac.kr

Abstract. XML is increasingly popular for knowledge representations. However, mining association rules from XML-based data is a challenging issue. Several encouraging approaches for mining rules in tree dataset have been proposed, but simplicity and efficiency still remain significant impediments for further development. In this paper, we adjust and fine-tune the label projection method which was published to compute valuable information from trees. The suggested approach avoids the computationally intractable problem caused by the number of nodes contained in the tree dataset.

Keywords: XML mining, maximal frequent subtree, XML association rule.

1 Introduction

Since the problem of extracting association rules was first introduced in [1], a large amount of work has been done in various directions. The famous Apriori algorithm for extracting association rules was published independently in [2] and in [7]. Then, a number of algorithms for extracting association rules from multivariate data have been proposed [4,5].

Under the traditional framework for association rule, the basic unit of data to deal with is database record, and the construct unit of a discovered association rule is item having an atomic value. However, since the structure of tree is fundamentally different, it is required to have counterparts of record and item in association relationships. Several methodologies for XML data were suggested [3,8,9] in the interest of flexibility and hierarchy of tree, the construct unit of a tree association rule is usually generated by repeated tree joins which are performed by nodes combinations. The combinatorial time for unit generations, therefore, becomes an

* This work was supported by Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-0005861).

** The corresponding author (e-mail: dhwon@security.re.kr).

inherent bottleneck of mining association rules from trees. If one can only obtain the units without tree join operations, the rule mining performance can be substantially improved. In this paper, we define some fundamental concepts applied to the association rules for tree data, which are guided by a label projection that was published in [6].

2 Label Projection

What if the database has been organized in a label-driven layout? The label itself plays the key role which is usually performed by tree or transaction indexes. All trees in a database *D* are reorganized according to labels. During scan of the trees in *D* , all nodes with the same label are grouped together. The nodes composed of the same tree form a member of the group and the number of members actually determines the frequency of the given label; the maximum number of members is a number of trees in *D* , which is called *label-projection.* After all labels are projected, the document-driven layout is changed into label-driven layout in which the time complexity to check labels' frequency requires at most 0(1 *L I I D* , where *L* is a set of labels. If hash-based search is used, the complexity is reduced up to 0(1 *D* .

Let 1 be a label in *L* . During pre-ordered scanning trees, tree indexes and node indexes which are projected by 1 construct a single linked list. It is called label list and the label list for a given label 1 is denoted 1 — *list* . The constructed label lists are collected and stored in the memory. Whenever a label is given, a corresponding label list is retrieved and the count of its members is returned. Due to its activity, the collection is named as *L* -dictionary, denoted L_{dw} .

3 Label List Extension

L_{aw} contains all label lists constructed from label projection. To be a frequent label, a label list has members more than or equal to a given threshold 8 . The current *L* however, does not differentiate label lists according to their projected label frequency.

A label list is said to be projected from a frequent label iff 1 — *list* 6·. Now the L_{dC} has only the label lists which are the projection of frequent labels and, thus, it is differently notated as rd_{ie} . All labels which are mapped to nodes of a tree should be frequent in order for the tree to be a maximal frequent tree. And usually a maximal frequent tree is produced by repeatedly growing smaller frequent subtrees. Therefore, the label of an attaching node should be frequent, if the grown subtree is required to be frequent. We will develop a candidate hash table, but how to do it will be given in a full version of the paper due to the space limit.

The current $L+_{die}$ contains all frequent labels and all *potential frequent* paths. A path is a sequence of edges and an edge is a line segment joining two nodes in a tree. Two nodes composing an edge should have frequent labels and appear together as

many as *8* if the edge wants to be frequent, and all edges composing a path should be frequent and they appear together as many as 8 if the whole path wants to be frequent. It is not guaranteed, however, with frequent label lists in r_d .

To verify path frequencies, explicit edges between any two nodes have to be unveiled from $L_{a=y}$. During the read of 1 —*list* , edges are formed by joining a symbolic node whose label is 1 and symbolic nodes of parent indexes' labels in its members. Unveiling edges totally relies on every frequent label lists because the symbolic nodes of parent indexes' labels have also their frequent label lists. The hidden paths between 1 — *list* and other label lists are discovered by extending the node of label 1 with the nodes of other 1 —*lists.* We call such processes label list extensions and the label list extension is committed to each label list in $_{Lary}$.

After completing the work, the labels of frequent label lists are joined together via symbolic nodes. The structure of the result is a tree whose root is labeled by 00 which is a dummy root. This tree contains all of potentially maximal frequent subtrees and thus is named *potentially maximal pattern tree* (PMP-tree in short). The tree is actually derived from r_d , where each edge has its own count to keep how many often it is occurred in the tree. Based on those counts, the edges whose counts are less than a given 8 are cleared off from PMP-tree. After deleting such edges and rearranging the tree, the goal of this paper is produced.

4 Conclusion

This paper has presented some key definitions and skeleton outline for label-driven association rules extraction from XML-based data. We are currently finishing touches of practical algorithms for our approach and evaluating the performance results.

Acknowledgments. This work was supported by Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-0005861).

References

1. Agrawal, R., Imielinski, T. and Swami, A. N.: Mining association rules between sets of items in large databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 207-216, 1993.
2. Agrawal, R. and Srikant, R.: Fast algorithms for mining association rules. In Proceedings of the 20th International Conference on VLDB, pp. 478-499, 1994.
3. Feng, L. and Dillon, T.: Mining interesting XML-enabled association rules with templates. In Proceedings of the 3rd International Workshop on Knowledge Discovery and Inductive Databases 2004, LNCS vol. 3377, pp. 66-88, 2005.
4. Han, J. and Fu, Y.: Discovery of multiple-level association rules from large databases. In Proceedings of the 21st International Conference on VLDB, pp. 420-431, 1995.

5. Srikant R. and Agrawal, R.: Mining generalized association rules. In Proceedings of the 21st International Conference on VLDB, pp. 409-419, 1994.
6. Paik, J., Nam, J., Youn, Y., and Kim, M.: Discovery of useful patterns from tree-structured documents with label-projected database. LNCS vol. 5060, pp. 264-278, 2008.
7. Toivonen, H.: Sampling large databases for association rules. In Proceedings of the 22th International Conference on VLDB, pp. 43-52, 1996.
8. Zhang, J., Ling, T. W., Bruckner, R. M., Tjoa, A. M., and Liu, H.: On efficient and effective association rule mining from XML data. In Proceedings of the 15th International Conference on Database and Expert Systems Applications, pp. 497-507, 2005.
9. Zhang, S., Zhang, J., Liu, H., and Wang, W.: XAR-miner: Efficient association rules mining for XML data. In Proceedings of the 14th international conference on World Wide Web, pp. 894 - 895, 2005.