Efficient Mining of Frequent Closed XML Query Pattern

Jian-Hua Feng (冯建华), Qian Qian (钱 乾), Jian-Yong Wang (王建勇), and Li-Zhu Zhou (周立柱)

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
E-mail: {fengjh, jianyong, dcszdh}@tsinghua.edu.cn; qgpeter99@mails.tsinghua.edu.cn
Received October 12, 2006; revised April 3, 2007.

Abstract Previous research works have presented convincing arguments that a frequent pattern mining algorithm should not mine all frequent but only the closed ones because the latter leads to not only more compact yet complete result set but also better efficiency. Upon discovery of frequent closed XML query patterns, indexing and caching can be effectively adopted for query performance enhancement. Most of the previous algorithms for finding frequent patterns basically introduced a straightforward generate-and-test strategy. In this paper, we present SOLARIA*, an efficient algorithm for mining frequent closed XML query patterns without candidate maintenance and costly tree-containment checking. Efficient algorithm of sequence mining is involved in discovering frequent tree-structured patterns, which aims at replacing expensive containment testing with cheap parent-child checking in sequences. SOLARIA* deeply prunes unrelated search space for frequent pattern enumeration by parent-child relationship constraint. By a thorough experimental study on various real-life data, we demonstrate the efficiency and scalability of SOLARIA* over the previous known alternative. SOLARIA* is also linearly scalable in terms of XML queries’ size.

Keywords computer software, frequent closed pattern, data mining, XML, XPath

1 Introduction

With the rapid growth for the last decade, XML data has become one of the most important collections of knowledge that the human being has ever had. The discovery of XML query patterns gains its focus with huge demands of efficient query evaluation and information retrieval arising. As much research has been undertaken on XML indexing\cite{1-3} and caching\cite{4}, discovering frequent XML query patterns turns out to be a significant and effective premise of query optimization for its capability of “focus” capturing.

As an example, Fig.1 shows two sample XML queries occurring in an university. Query (a) finds all names of professors who published one paper on “JCST” in the year of “2004”, while Query (b) finds names of professors who teach “Data Mining” and published one paper on “JCST”.

Consider the two query patterns in Fig.1, the sub-structure “Prof/Paper/Journal” is a frequent pattern, which hints that extra indexing or caching established on this “focus” will definitely bring benefits for query evaluation. For this purpose, frequent query pattern mining gains the momentum for its crucial role about query optimization.

In recent years many studies have presented convincing arguments that for mining frequent patterns (for both itemsets and sequences), one should not mine all frequent patterns but the closed ones that contain no super-structure with the same support because the latter leads to not only more compact yet complete result

---

This work is supported in part by the National Natural Science Foundation of China under Grant No. 60573094, the National Grand Fundamental Research 973 Program of China under Grant No. 2006CB601203, the National High Technology Development 863 Program of China under Grant No. 2006AA01A101, and Tsinghua Basic Research Foundation under Grant No. JQ060022.
set but also better efficiency. However, unlike mining frequent itemsets, there are not so many methods proposed for mining closed patterns, especially for XML query patterns.

Traditional frequent pattern mining approaches typically follow straightforward generate-and-test strategy, which includes two phases of frequent pattern generation and containment testing\cite{4-8}. The focus of tree-structured data mining research has recently moved towards efficient frequent pattern enumeration and fast containment testing algorithms. Motivated by the need of reducing costly containment testing, an efficient algorithm, SOLARIA\textsuperscript{*} based on the existing algorithms SOLARIA\textsuperscript{[9]}, SCALER\textsuperscript{[10]} and BIDE\textsuperscript{[11]}, has been proposed, which involves a novel application of frequent sequence mining and can be used to accelerate frequent closed XML queries. SOLARIA\textsuperscript{*} mines the set of frequent sequences that preserve the parent-child and sibling ordering relationships among the XML elements/attributes, which avoids expensive tree containment testing elegantly.

Our Contributions. The contributions of this paper can be summarized as follows.

- First, we demonstrate a novel application of the sequence mining approach to the filed of frequent closed XML query pattern mining.
- An efficient algorithm, SOLARIA\textsuperscript{*}, is provided by pushing the parent-child and sibling ordering constraint into the traditional sequential pattern mining. Without candidate maintenance and costly sub-tree containment testing, great enhancement of performance in frequent XML query pattern discovery is achieved.
- A Bi-Directional Extension closure checking scheme is proposed for XML query pattern mining, which speeds up the complex closure checking and assures the correctness of the algorithm.
- At last, a thorough performance study is conducted to evaluate SOLARIA\textsuperscript{*}'s efficiency and scalability in comparison with the state-of-the-art algorithms.

The rest of this paper is organized as follows. Section 2 gives related work about frequent closed XML pattern mining. We formalize the frequent closed XML query pattern mining problem in Section 3. Section 4 presents SOLARIA\textsuperscript{*}, an effective and elegant sequencing algorithm to mine closed XML query patterns. In Section 5, we present a thorough experimental study to evaluate SOLARIA\textsuperscript{*}. Section 6 is the conclusion.

2 Related Work

Mining frequent substructure of trees, graphs and sequences has drawn much attention as an essential data mining task, with various applications including market and customer analysis, web log analysis, pattern discovery in protein sequences and XML frequent patterns for caching, and so on.

For tree and graph mining, frequent pattern discovering was first addressed in biological science.\cite{6} proposes an efficient algorithm to mine frequent substructures in protein and chemical compounds. In graph database, algorithm FSG in \cite{12} is treated as a fast miner for discovering connected sub-graphs by extending the notion of level-by-level expansion of \cite{13}. Motivated by discovering user navigation patterns in web surfing, \cite{14} proposes sub-tree mining algorithm in forest, which faces more complex data situation. FREQT\textsuperscript{[15]} and TreeFinder\textsuperscript{[16]} aim at finding frequent subtrees in a collection of semi-structured documents, but still cannot solve the problem of XML query pattern mining due to the existence of “"" and “/"”. To our best knowledge, FastXM\textsuperscript{[4]} is the most efficient mining algorithm for XML frequent query pattern discovery, as only valid candidate sub-query patterns are enumerated by FastRSTGen for costly containment testing. It still follows the traditional idea of generate-and-test paradigm for tree-structured data mining. Global query pattern tree needs to be generated for sub-query patterns’ enumeration, as well as expensive candidate generation and containment testing.

On the other hand, for sequence mining,\cite{17-19} are mainly focusing on general and constraint-based sequence mining problems. Various researches have been done on frequent episode mining\cite{5}, cyclic association rule mining\cite{20}, temporal relation mining\cite{7}, partial periodic pattern mining\cite{21}, and long sequential pattern mining in noisy environment\cite{22}. But the voice of a frequent pattern mining algorithm should not mine all frequent patterns but only the closed ones comes out with convincing arguments for its better efficiency and more compact results without valuable information loss. CloSpan\cite{5} and BIDE\textsuperscript{[11]} are two well-known closed sequence mining algorithms, where CloSpan still follows the candidate maintenance-and-test paradigm and BIDE adopts Bi-Directional Extension to avoid candidate maintenance.

Currently, CMTreeMiner\textsuperscript{[23]} is a computationally efficient algorithm that discovers all closed frequent subtrees and maximal frequent subtrees from a database of root-ordered trees. It is the first algorithm that, instead of using post-processing procedures, directly mines only closed and maximal frequent subtrees.

3 Preliminaries

XML Query Pattern (XQP). The XML queries are mainly expressed in XPath\textsuperscript{[24]} or XQuery\textsuperscript{[25]} nowadays, whose basic features are regular path expres-