Data Mining on XML Data

Qin Ding
East Carolina University, USA

INTRODUCTION

With the growing usage of XML data for data storage and exchange, there is an imminent need to develop efficient algorithms to perform data mining on semi-structured XML data. Mining on XML data is much more difficult than mining on relational data because of the complexity of structure in XML data. A naïve approach to mining on XML data is to first convert XML data into relational format. However the structure information may be lost during the conversion. It is desired to develop efficient and effective data mining algorithms that can be directly applied on XML data.

BACKGROUND

In recent years, XML has become very popular for representing semi-structured data and a standard for data exchange over the web. XML stands for Extensible Markup Language. It is a simple and very flexible text format derived from SGML (Standard Generalized Markup Language). Both XML and SGML are meta-languages because they are used for defining markup language. Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere.

Below is a simplified example of an XML document. As can be seen, elements (or called tags) are the primary building blocks of an XML document. Unlike HTML which uses a fixed set of tags, XML allows user to define new collections of tags that can be used to structure any type of data or document. An element can have descriptive attributes that provide additional information about the element. Document Type Definitions (DTDs) can be used to specify which elements and attributes we can use and the constraints on these elements, such as how elements can be nested. XML allows the representation of semi-structured and hierarchal data containing not only the values of individual items but also the relationships between data items.

Data mining is the process to extract useful patterns or knowledge from large amount of data. As the amount of available XML data is growing continuously, it will be interesting to perform data mining

Figure 1. An XML document

```
<Department>
   <People>
     <Employee>
       <PersonalInfo> …</PersonalInfo>
       <Education> … </Education>
       <Publications>
          <Book year=”2002” name=”XML Query Languages”>
             <Author> … </Author>
             <Publisher> … </Publisher>
             <Keyword>XML</Keyword> … <Keyword>XQuery</Keyword>
          </Book>
          <Journal year=“2000” vol=”4” name=”DMKD” Publisher=”Kluwer”>
             <Author> ... </Author>>
             <Keyword>RDF</Keyword> … <Keyword>XML</Keyword>
          </Journal>
       </Publications>
     </Employee>
   </People>
</Department>
```

on XML data so that useful patterns can be extracted from XML data. From the example in Figure 1, we might be able to discover such patterns as "researchers who published about XML also published something related to XQuery" where "XML" and "XQuery" are keywords of the publications. This interesting pattern can be represented as an association rule in the format of "XML => XQuery". The task of data mining on XML data can be significant, yet challenging, due to the complexity of the structure in XML data.

Data mining has been successfully applied to many areas, ranging from business, engineering, to bioinformatics (Han & Kamber, 2006). However, most data mining techniques were developed for data in relational format. In order to apply these techniques to XML data, normally we need to first convert XML data into relational data format, and then traditional data mining algorithms can be applied on converted data. One way to map XML data into relational schema is to decompose XML documents entirely, remove the XML tags, and store the element and attribute values in relational tables. In this process, most aspects of the XML document structure are usually lost, such as the relative order of elements in the document, and the nested structure of the elements.

In order to avoid mapping the XML data to relational format, researchers have been trying to utilize XQuery, the W3C (World Wide Web Consortium) standard query language for XML, to support XML mining. On the one hand, XQuery provides a flexible way to extract XML data; on the other hand, by adding another layer using XQuery, the mining efficiency may be greatly affected. In addition, a query language such as XQuery may have limited querying capabilities to support all the data mining functionalities.

MAIN FOCUS

Data mining on XML data can be performed on the content as well as the structure of XML documents. Various data mining techniques can be applied on XML data, such as association rule mining and classification. In this section, we will discuss how to adapt association rule mining and classification techniques to XML data, for example, how to discover frequent patterns from the contents of native XML data and how to mine frequent tree patterns from XML data. We will also discuss other work related to XML data mining, such as mining XML query patterns (Yang et al, 2003) and using XML as a unified framework to store raw data and discovered patterns (Meo & Psaila, 2002).

Association Rule Mining on XML Data

Association rule mining is one of the important problems in data mining (Agrawal et al, 1993; Agrawal & Srikang, 1994; Han et al, 2000). Association rule mining is the process of finding interesting implication or correlation within a data set. The problem of association rule mining typically includes two steps. The first step is to find frequently occurring patterns. This step is also called "Frequent Pattern Mining". The second step is to discover interesting rules based on the frequent patterns found in the previous step. Most association rule algorithms, such as Apriori (Agrawal & Srikant, 1994) and FP-growth (Han et al, 2000), can only deal with flat relational data.

Braga et al addressed the problem of mining association rules from XML data in their recent work (Braga et al, 2002; Braga et al, 2003). They proposed an operator called "XMINE" to extract association rules from native XML documents. The XMINE operator was inspired by the syntax of XQuery and was based on XPath, which is the XML path language, an expression language for navigating XML document. Using the example in Figure 1, the "XMINE" operator can be used, as shown in Figure 2, to discover rules such as "XML => XQuery", where both sides of the rule have the same path, which is specified as "ROOT/Publications//Keyword" using the XPath language. The limitation of this work is that it only focuses on mining specific rules with pre-specified antecedent (the left-hand side) and consequence (the right-hand side), while the general association rule mining should target all the possible rules among any data items (Ding et al, 2003).

Dobbie and Wan proposed an XQuery-based Aprioir-like approach to mining association rules from XML data (Dobbie & Wan, 2003). They show that any XML document can be mined for association rules using only XQuery without any pre-processing and post-processing. However, since XQuery is designed only to be a general-purpose query language, it puts a lot of restriction on using it to do complicated data mining process. In addition, the efficiency is low due to the extra cost of XQuery processing.

Another algorithm called TreeFinder was introduced by Termier et al (Termier et al, 2002). It aims at search-