International Journal of Database Theory and Application
Vol.7, No.1 (2014), pp.83-92
http://dx.doi.org/10.14257/ijdta.2014.7.1.08

Research on XML Data Mining Model Based on Multi-level Technology

Jie Ma

Chongqing University of education, Chongqing, 400065, China

172787469@qq.com

Abstract

The era of Web 2.0 has been coming, and more and more Web 2.0 application, such social networks and Wikipedia, have come up. As an industrial standard of the Web 2.0, the XML technique has also attracted more and more researchers. However, how to mine value information from massive XML documents is still in its infancy. In this paper, we study the basic problem of XML data mining-XML data mining model. We design a multi-level XML data mining model, propose a multi-level data mining method, and list some research issues in the implementation of XML data mining systems.

Keywords: *XML, data mining model, multi-level technique, World Wide Web*

1. Introduction

The World Wide Web (or the Web for short) has impacted on almost every aspect of our lives. It is the biggest and most widely known information source that is easily accessible and searchable. It consists of billions of interconnected documents (called Web pages) which are authored by millions of people. Since its inception, the Web has dramatically changed our information seeking behavior. Before the Web, finding information means asking a friend or an expert, or buying/borrowing a book to read. However, with the Web, everything is only a few clicks away from the comfort of our homes or offices. Not only can we find needed information on the Web, but we can also easily share our information and knowledge with others [1]. The Web has also become an important channel for conducting businesses. We can buy almost anything from online stores without needing to go to a physical shop. The Web also provides convenient means for us to communicate with each other, to express our views and opinions on anything, and to discuss with people from anywhere in the world. The Web is truly a virtual society.

The rapid growth of the Web in the last decade makes it the largest publicly accessible data source in the world. The Web has many unique characteristics, which make mining useful information and knowledge a fascinating and challenging task. There are many kinds of data on the web, video, audio, pictures, text, *etc*., and they are organized in the format of html, extensible Markup Language (XML), or database. As XML is more and more popular, and it has been the standard of Web 2.0, so more and more data on the web will be organized with XML. Hence, XML data mining will be a very important data mining technique in the future [2]. The past few years have seen a growth in the adoption of the XML, thanks to its ability to provide a standardized, extensible mean of including semantics information within semi-structured data. The flexible nature of XML makes it an ideal basis for data mining arbitrary languages. One such example is the Predictive Modelling Markup Language (PMML) [3], an industry standard for the representation of mined models as XML documents.

ISSN: 2005-4270 IJDTA

Different from relational database, where data are stored in tables and data mining is to find useful information from these data tables, data are organized in the form of tree in XML documents, and this make researchers have to find new data mining models for this kind of data.

Several attempts have been made to overcome the limitations of database systems over data mining tasks [4]. Two well-known approaches are the development of data mining query languages XQuery [5] and XSLT [6] for XML data. Most of these attempts propose extensions of SQL that might be difficult to achieve, and after the years have not been adopted by commercial DBMSs [7]. Another problem is that most of these works focus on the mining step itself, without addressing the most important and time consuming task in the discovery process: XML data mining model. For instance, Calders *et al.*, [8] proposed an algebra method to integrate data and patterns. Li *et al.*, [9] proposed a XML text classification based on genetic-SVM classifier. This is especially true for the XML documents.

In this paper, we propose a multi-level XML data mining model. An XML document is the structure of tree, and query of such tree is from its root to its leaves. This model takes the tree structure of the XML documents into consideration, and proposes a multi-level data mining method.

The rest of the paper is organized as follows. Related work is given in Section 2. In Section 3, we give some background about XML. The multi-level XML data mining model is given in Section 4. In Section 5, we give two examples about the multi-level data mining method. Research issues are given in Section 6, and conclusion is given in Section 7.

2. Related work

Zhang *et al.*, [10] proposed three data mining models suitable to data processing according to some technical challenges on the characteristics data for Internet of Things. The first model is the multi- layer data mining model, which consists of data collection, data processing, event processing and data mining service. The second model is distributed data mining model that could be used to solve the problem of data stored in different locations. The third model is grid-based data mining model, which could utilize the potentially unlimited amount of data by using grid. At the same time, the paper learns from “decomposition - build "theory to solve the data mining problem in the internet of things. Finally, the key issues are discussed about model application. In order to realize the function of data mining reusability, Su *et al.*, [11] studied the data mining model for distributed databases.

Providing efficient mining algorithm to discover frequent XML user query patterns is crucial, as many applications use XML to represent data in their disciplines over the Internet. These frequent XML user query patterns can be used to design an index mechanism or cached and thus enhance XML query performance. Several XML mining algorithms have been proposed to record all of user queries in a global tree and thus discover the frequent XML query patterns on the tree. However, none of them encodes user queries and thus stores the codes in the system to enhance the mining performance. By using these codes, the user query tree information is preserved and less memory space is used. Chang [12] proposed a new idea to encode XML user query trees in a two-dimensional coordinate system, where x and y are the coordinates of the two-dimensional space.

With feature of simplicity, flexibility and cross-platform, XML has been more and more used for data storing and exchanging. Because of this trend, designing data warehouse based on XML also becomes a hot topic in realm of data warehouse researching now. Zhou *et al.*, [13] proposed a data cube model named XTree Cube whose constructing data cube is based on a tree structure. XTree Cube transforms dimensions of data cube into a tree-style structure

and stores it in an XML file. This structure reduces the data redundancy of XML data cube by saving the relationship between dimensions in the XML file and increases the efficiency of data cube constructing and data querying.

Bogorny *et al.*, [14] proposed a novel solution to reduce the gap between databases and data mining in the domain of trajectories of moving objects, aiming to reduce the effort for data preprocessing. They proposed a general framework for modeling trajectory patterns during the conceptual design of a database. The proposed framework is a result of several works including different data mining case studies and experiments performed by the authors on trajectory data modeling and trajectory data mining. It has been validated with a data mining query language implemented in PostGIS, which allows the user to create, instantiate and query trajectory data and trajectory patterns.

Abdullah *et al.*, [15] proposed a data mining system framework using the ERP framework. They applied data mining applications to evaluate the best result for the growth and establishment of a company using the ERP database. They proposed a model which integrates the database, customer queries, transactions, and all other specifications used in ERP systems, then use data mining techniques to integrate decision making and forecast flows. By using ERP's characteristics and background they gathered the data from central database in cluster format which is based on the action taken against the queries generated by the customers. Furthermore, they used the clustered data by Apriori Algorithm to extract new rules and patterns for the enhancement of an organization. That is a complete implementation of data mining applications on ERP framework to predict the solution of upcoming queries. This will make the best association between the customers and organization, and customer will always satisfied with company's policies.

3. XML Background

The Extensible Markup Language (XML) is a subset of SGML that is completely described in this document. Its goal is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML. A data object is an XML document if it is well-formed, as defined in this specification. In addition, the XML document is valid if it meets certain further constraints.

Each XML document has both a logical and a physical structure. Physically, the document is composed of units called entities. An entity may refer to other entities to cause their inclusion in the document. A document begins in a "root" or document entity. Logically, the document is composed of declarations, elements, comments, character references, and processing instructions, all of which are indicated in the document by explicit markup. The logical and physical structures MUST nest properly.

3.1. Well-Formed XML Documents

A textual object is a well-formed XML document if:

I. Taken as a whole, it matches the production labeled document.

II. It meets all the well-formedness constraints given in this specification.

III. Each of the parsed entities which is referenced directly or indirectly within the document is well-formed.

Document: document:= (prolog element Misc*) - (Char* RestrictedChar Char*)

Matching the document production implies that:

I. It contains one or more elements.

II. There is exactly one element, called the *root*, or document element, no part of which appears in the content of any other element. For all other elements, if the start-tag is in the content of another element, the end-tag is in the content of the same element. More simply stated, the elements, delimited by start-and end-tags, nest properly within each other.

As a consequence of this, for each non-root element C in the document, there is one other element P in the document such that C is in the content of P, but is not in the content of any other element that is in the content of P. P is referred to as the parent of C, and C as a child of P. Figure 1 is a well-formed XML document, and its tree presentation is in Figure 2.

```
<bookstore>
<book category="COOKING">
  <title lang="en">Everyday Italian</title>
  <author>Giada De Laurentiis</author>
  <year>2005</year>
  <price>30.00</price>
</book>
<book category="CHILDREN">
  <title lang="en">Harry Potter</title>
  <author>J K. Rowling</author>
  <year>2005</year>
  <price>29.99</price>
</book>
<book category="WEB">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>
```

Figure 1. A well-formed XML document

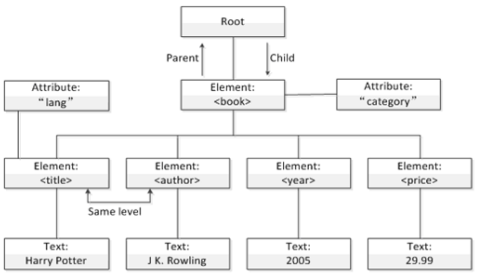


Figure 2. A well-formed XML document

4. Model Overall Designing

A diagram of the architecture of our system is shown in Figure 3. It is similar in spirit to the architecture of research prototypes (*e.g.*, [16]), with the distinguishing factor that our system is built upon an XML data model.

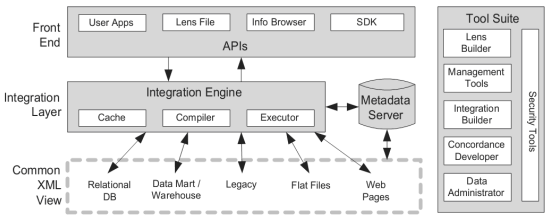


Figure 3. Architecture of XML data mining model

The query languages supported by our system are XQuery, XSLT and XML-QL [17]. Ultimately, we plan to adopt the standard query language recommended by the W3C Query Working Group. Users and applications interact with the system using a set of mediated schemas. These schemas are essentially definitions of views over the schemas of the data sources (similar to the global-as-view approach [18]). It is important to notice that these schemas are built in a hierarchical technology. That is, we can define successive schemas as views over other underlying schemas. This provides important flexibility when there are going to be multiple classes of users and applications using the system, and also facilitates defining the integration of the data sources, because it can be done in an incremental fashion.

The system front end is flexible, offering multiple layers of access. For example, a *lens* is an object that contains a set of XML queries, parameters, XSL formatting, and authentication information. Result formatting can be targeted to specific devices (*e.g.*, web interface, wireless device). Customers who wish to use a lower-level interface to the integration engine are also supported.

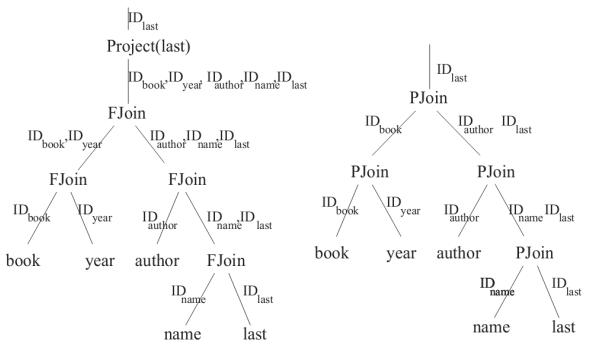
When a query is posed to the integration engine, it is parsed and broken into multiple fragments based on the target data sources. The compiler translates each fragment into the appropriate query language for the destination source; for example, if an RDB is being queried, then the compiler generates SQL. Note that the compiler considers both the type of the underlying source, information concerning the layout of the data within the sources, and the presence of indices on the data. The metadata server contains the mappings that allow the query language to be split apart and translated appropriately; mappings are set via the management tools. Load balancing is provided; multiple instances of the integration engine can be run simultaneously on one or more servers.

5. Examples of Multi-level Data Mining Method

In this section, we introduce our multi-level data mining method. XML documents are all stored in the form of trees, and thus query of tree is different from that of relational database. We consider several operators that could follow a pattern tree selection: projection, intersection, union, and set difference. For each of these four operators, we consider the benefit of merging it with the pattern tree selection(s) that it follows. At the macro level, merging operators seems intuitive and simple. But since both the micro level operators that compose the selection and the mapping between the macro selection and the micro containment joins will get affected, the merging process is no longer as simple.

Example 1: Consider pattern tree presented in Figure 4(a). The query seeks books with author last name “Bernstein” and year greater than 1995. We will alter it a bit to return *last*. Figure 4(b) is a plan that can be used to evaluate the query. To evaluate this query, we would first generate lists of candidate nodes that match individual nodes in the pattern (*e.g.*, “book” nodes, “author” nodes, and so on). Then we will compute a sequence of containment joins,

one for each edge in the pattern. For instance, suppose the first join is name-last. The result of this join is a set of pairs of nodes that jointly satisfy the relevant portion of the pattern. The next containment join, say between author and name actually joins a set of name-last pairs with a set of author pairs to produce a set of author-name-last triples. Finally, after all containment joins have been evaluated, we have a set of 5-tuples, which are the results of the pattern match selection. A projection operator is then used to focus on the last nodes and eliminate the others.



(a) Full Join Plan (b) Partial Join Plan

Figure 4. Difference between Full and Partial containment joins

Example 2: Figure 5 presents a pattern to be matched. PL, the projection list, is specified to be *book* and *year*. The set *S* has nodes *name* in it twice because it has three edges incident. *book* is not in *S* since it is in PL. The other three (leaf) nodes have only one edge incident each. A possible sequence of partial binary containment joins to evaluate this query is shown, along with the manipulations of the set *S*. In the first step, *name* is joined with *first*. *first* is not retained in the result since it is neither in *S* nor in PL. *name* is retained since it is in *S*, but one occurrence of name is removed from *S*. In the last step, *book* and *year* are retained in the result since both are in PL.

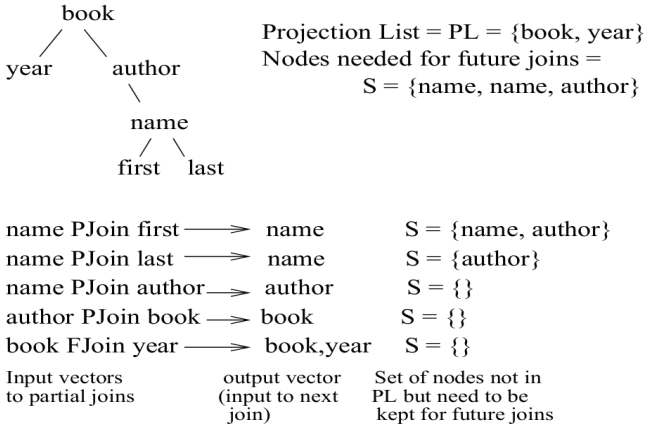


Figure 5. Difference between Full and Partial containment joins

6. Research issues

In this section, we present the key research issues that need to be addressed to realize these various mining efforts.

6.1. Duration of real data collection

In order to make the extracted knowledge convincible and accurate, the data collection is expected to go on for a significant time period. However, finding appropriate time duration for data collection is a challenging task as it depends on several parameters such as characteristics of source and application, frequency of changes, number of available versions, type of knowledge to be discovered, etc. It is an open problem to estimate the appropriate duration for different applications. Such estimation will reduce the resource consumption of the data mining process and maximize the quality of results generated by different XML structural delta mining techniques.

6.2. Determining schedules for structural delta generation

XML data mining research has largely assumed that XML documents are static. However, in reality the documents are rarely static. XML structural delta mining aims to extract knowledge by analyzing the structural evolution pattern of history of XML documents. One key issue is to generate a sequence of structural deltas that can be fed to the mining engine for pattern extraction. To improve the accuracy of the mining process, ideally we should be able to harness the complete set of structural deltas during a particular time period. As these documents often reside in autonomous and remote sources, it is not realistic to assume that structural deltas will be automatically propagated to the mining engine. Hence, finding appropriate schedules for change detection is important so that set of structural deltas used for the mining task is complete and contain sufficient data to guarantee reliable and accurate mining results. Defining schedules for structural delta detection is a challenging task, because the rate of change of the XML document may vary drastically from document to document. Note that the naive solution of polling the sources periodically is not an efficient process for two reasons. First, we may miss some of the intermediate structural deltas as frequency of change of a particular document/source may not match the polling frequency. Second, this approach may unnecessarily overload the mining process by attempting to detect changes when the documents have not changed. A more efficient way of solving this problem is to predict the rate of change of documents by analyzing the change history.

6.3. Scalable and efficient mining algorithms

One of the most important concerns in designing a data mining algorithm is the scalability and efficiency. Similarly, for our XML structural delta mining, we need to design efficient and scalable algorithms that can handle very long sequence of historical XML documents that are very large. In the context of XML structural delta mining, the algorithm should be scalable with respect to not only the size of the XML delta sequence, but also to various values of different dynamic metrics such as degree of dynamic , structure dynamic , and version dynamic in discovering the various types of interesting structures, association rules, and classifications. To address the scalability, a relational database-based approach may be more competitive than a memory based one. However, in terms of efficiency, a memory-based approach is more competitive than the relational data-base-based approach. To sum up, there should be a trade-off between the scalability and the efficiency. In addition, another issue should be considered. Since the dataset is accumulated over time, the characteristics and

knowledge hidden behind the data collection may vary accordingly. This feature makes it desirable to design an incremental data mining algorithm that can use the previously discovered results and the current changes to update the extracted knowledge.

6.4. Unified mining framework

Since data from many fields, such as Web log data and biological data, can be represented in XML format and there are different types of knowledge that can be extracted, it is desirable to design a unified mining framework that is suitable for different applications in different fields and can discover various types of knowledge. For example, different users may be interested in different types of knowledge. The unified mining frame-work should be able to provide different types of knowledge to different users with regards to their requirements. That is, for the same data, the mining framework should contain different data mining techniques so that it can provide personalized knowledge for all types of possible users. Similarly, for applications in different areas, the background knowledge will be different. The unified mining framework should be able to incorporate the corresponding background knowledge into the mining process.

7. Conclusion

As a standard of Web 2.0, XML is touted as the breakthrough in data exchange. As XML documents become more and more, the ability to mine knowledge from XML sources becomes more and more important. Thus, there is a great need to apply data mining techniques to XML data. This paper suggests a XML data mining model based on multi-level technique, proposes a multi-level XML data mining method, and describes some research issues in the implementation of XML data mining systems.

Acknowledgements

Chongqing municipal education commission funded projects: based on semantic understanding of intelligent video monitoring system key technology research, Numbers: KJ111504.

References

[1] B. Liu, "Web Data Mining: Exploring Hyperlinks, Contents and Usage Data", Springer Verlag, **(2007)**.
[2] S. Madria, S. Bhowmick, W. Ng and E. Lim, "Research issues in web data mining", Data Warehousing and Knowledge Discovery, **(1999)**, pp. 805.
[3] R. Grossman, S. Bailey, A. Ramu, B. Malhi, P. Hallstrom, *et al.*, "The management and mining of multiple predictive models using the predictive modeling markup language", Information And Software Technology, vol. 41, no. 9, **(1999)**, pp. 589-595.
[4] H. Wang and C. Zaniolo, "Atlas: A native extension of sql for data mining", **(2003)**.
[5] S. Boag, D. Chamberlin, M. F. Fern, A. Ndez, D. Florescu, J. Robie, *et al.*, "XQuery 1.0: An XML query language", W3C working draft, vol. 12, **(2003)**.
[6] M. Kay, "XSLT programmer's reference", Wrox Press Ltd., **(2001)**.
[7] J. Boulicaut and C. Masson, "Data mining query languages", Data Mining and Knowledge Discovery Handbook, **(2005)**, pp. 715-726.
[8] T. Calders, L. V. Lakshmanan, R. T. Ng and J. Paredaens, "Expressive power of an algebra for data mining", ACM Transactions on Database Systems (TODS), vol. 31, no. 4, **(2006)**, pp. 1169-1214.
[9] L. Yuxiang, Z. Dewen, C. Lijun, Y. Liqun and Y. Zhanghong, "XML Text Classification Based on Genetic-SVM Classifier", JCIT: Journal of Convergence Information Technology, vol. 8, no. 4, **(2013)**, pp. 347-353.
[10] C. Zhang, G. Zeng, H. Wang and X. Tu, "Analysis on Data Mining Model Objected to Internet of Things", IJACT: International Journal of Advancements in Computing Technology, vol. 4, no. 21, **(2012)**, pp. 615-622.
[11] S. Dan, J. Kun and Z. FuYan, "The Study on Data Mining Method for Distributed Databases", IJACT:

International Journal of Advancements in Computing Technology, vol. 4, no. 22, **(2012)**, pp. 664-670.
[12] T. -P. Chang, "XCode: a Novel Encoding Scheme of Frequent XML Query Pattern Mining", IJACT: International Journal of Advancements in Computing Technology, vol. 4, no. 12, **(2012)**, pp. 171-181.
[13] Z. Yinghui, Z. Yawei and M. Tinghuai, "XML Data Cube Based on Tree Structure", JCIT: Journal of Convergence Information Technology, vol. 8, no. 4, **(2013)**, pp. 423-430.
[14] V. Bogorny, C. Heuser and L. Alvares, "A conceptual data model for trajectory data mining", Geographic Information Science, **(2010)**, pp. 1-15.
[15] F. A. A. S. Al-Mudimigh, S. B. Z. Ullah and T. C. F. Saleem, "A Framework of an Automated Data Mining Systems Using ERP Model", International journal of computer and Electrical Engineering, vol. 1, no. 5, **(2009)**.
[16] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Ra-jaraman, Y. Sagiv, J. Ullman and J. Widom, "The TSIMMIS project: Integration of heterogeneous information sources", Journal of Intelligent Information Systems, vol. 8, no. 2, **(1997)** March, pp. 117–132.
[17] A. Deutsch, M. Fernandez, D. Florescu, A. Levy and D. Suciu, "A query language for XML", http://www.research.att.com/ mff/xml/w3c-note.html, **(1998)**.
[18] D. Florescu, A. Levy and A. Mendelzon, "Database tech-niques for the world-wide web: A survey", SIGMOD Record, vol. 27, no. 3, **(1998)** September, pp. 59–74.

Author

Jie Ma. She received her M.Eng. in Computer (2007) from University. Now she is devoted to research Data mining and e-commercial business. She has published almost 10 papers about data mining. Her current research interests include different aspects of computer application technology.