IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A Novel FP-Tree Algorithm for Large XML Data Mining

Amit Kumar Mishra[*1] and Hitesh Gupta[2]

[*1]Research Scholar, Department of Computer Science & Engineering, P.C.S.T., Bhopal, M.P., India

[2]H.O.D, Department of Computer Science & Engineering, P.C.S.T., Bhopal, M.P., India

amit.sist1@gmail.com

Abstract

The goal of data mining is to extract or mine" knowledge from large amounts of data. XML has become very popular for representing semi structured data and a standard for data exchange over the web. Mining XML data from the web is becoming increasingly important. The ever increasing demand of finding pattern from large XML data enhances the association rule mining. To date, the famous Apriori algorithm to mine any XML document for association rules without any pre-processing or post-processing has been implemented. But the algorithm only can mine the set of items that can be written a path expression for. However, the structure of the XML data can be more complex and irregular than that. Among the existing techniques, the frequent pattern growth (FP-growth) method is the most efficient and scalable approach. We propose an improved technique that extracting association rules from XML documents without any preprocessing or post processing. Our proposed improved algorithm, for mining the complete set of frequent patterns by pattern fragment growth. First Frequent Pattern-tree based mining adopts a pattern fragment growth method to avoid the costly generation of a large number of candidate sets and a partition-based, divide-and-conquer method is used. We propose an association data mining tool for XML data mining. It will increases the mining efficiency and also takes less memory.

Keywords: Data mining, Association mining, XML, XSTL, FP-Growth

Introduction

Data mining[1], which is also referred to as knowledge discovery in databases, means a process of nontrivial extraction of implicit, previously unknown and potentially useful information (such as knowledge rules, constraints, regularities) from data in databases. Other terms for data mining are knowledge mining from databases, knowledge extraction, data archaeology, data dredging, data analysis, etc. By knowledge discovery in databases, interesting knowledge, regularities, or high-level information can be extracted from the relevant sets of data in databases and be investigated from different angles, and large databases thereby serve as rich and reliable sources for knowledge generation and verification. Mining information and knowledge from large databases has been recognized by many researchers as a key research topic in database systems and machine learning[1], and by many industrial companies as an important area with an opportunity of major revenues. The discovered knowledge[2] can be applied to information management, query processing, decision making, process control, and many other applications. Re-searchers in many different fields, including database systems, knowledge-base systems, artificial intelligence, machine learning, knowledge acquisition, statistics, and spatial databases have shown great interest in data mining.

Nowadays, we have huge amount of data stored in each institute, university, and company. However, data could not tell us anything without processing. We often flooded by data but lack of the information. Data mining has attracted increasing interests in recent years trying to find the underlying models or patterns of the data, and making use of the found models and patterns. The data mining process is a rather complex procedure involving many candidate algorithms serving for various tasks for different types of data. For a real data mining problem, we often need both of the background knowledge from users and data miners. In one hand, the user's background knowledge is important. This background knowledge can be incorporated with the induction algorithm and used for evaluating the mined results.

Data mining has been used in a broad range of applications [2]. More and more leading-edge organizations are realizing that data mining provide them the ability to reach their goals in customer relationship management, risk management, fraud and abuse detection, and e-business[2] etc.

XML[3] is a Standard, flexible syntax for data exchanging Regular, structured data. As a platform-independent solution, XML is going to be used in many environments such as application integration and Web Services. With the continuous growth in XML data sources, the ability to manage collections of XML documents and discover knowledge from them for decision support becomes increasingly important. Mining of XML documents significantly differs from structured data mining and text mining. 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. Element tags and their nesting therein dictate the structure of an XML document. Due to the inherent flexibility of XML, in both structure and semantics, discovering knowledge from XML data is faced with new challenges as well as benefits.

Various data mining techniques[4] such as, decision trees, association rules[5], and neural networks are already proposed and become the point of attention for several years. Association rule mining technique is the most effective data mining technique to discover hidden or desired pattern among the large amount of data[6]. It is responsible to find correlation relationships among different data attributes in a large set of items in a database.

In 2000, Han et al[7] proposed the FP-growth algorithm—the first pattern-growth concept algorithm. FP-growth constructs an FP-tree structure and mines frequent patterns by traversing the constructed FPtree. The FP-tree structure is an extended prefix-tree structure involving crucial condensed information of frequent patterns. FP-tree structure: The FP-tree structure has sufficient information to mine complete frequent patterns. It consists of a prefix tree of frequent 1-itemset and a frequent-item header table. Each node in the prefix-tree has three fields: item-name, count, and node-link.

XML data can be more complex and irregular than that. Among the existing techniques, the frequent pattern growth (FP-growth) method is the most efficient and scalable approach. We propose an improved technique that extracting association rules from XML documents without any preprocessing or postprocessing. Our proposed improved algorithm, for mining the complete set of frequent patterns by pattern fragment growth. First Frequent Pattern-tree based mining adopts a pattern fragment growth method to avoid the costly generation of a large number of candidate sets and a partition-based, divide-and-conquer method is used. We propose an association data mining tool for XML data mining. It will increases the mining efficiency and also takes less memory

The rest of the paper is divided as follows. Section 2 present the data mining related work. Section 3 covers the proposed algorithm. Section 4 concludes the paper and also present future work.

Related Work

Algorithms for mining association rules[8] from relational data have been well developed. Several query languages have been proposed, to assist association rule mining. The topic of mining XML data has received little attention, as the data mining community has focused on the development of techniques for extracting common structure from heterogeneous XML data. For instance, [3] has proposed an algorithm to construct a frequent tree by finding common subtrees embedded in the heterogeneous XML data. [9] proposed a new and improved FP tree with a table and a new algorithm for mining association rules. The author proposed an efficient association rule mining technique with help of improved frequent pattern tree (FP-tree) and a mining frequent item set (MFI) algorithm. This algorithm mines all possible frequent item set without generating the conditional FP tree. It also provides the frequency of frequent items, which is used to estimate the desired association rules.

Algorithms for mining association rules from relational data have been implemented since long before. Association rule mining was first introduced at 1993 by R. Agrawal, T. Imielinski, and A. Swami [10]. The Apriori algorithm [11] uses a bottom-up breadth-first approach to find the large item set. As it was proposed to grip the relational data this algorithm cannot be applied directly to mine complex data. Another well-known algorithm is FP growth algorithm. It adopts divide-and-conquer approach. First it computes the frequent items and characterizes the frequent items in a tree called frequent-pattern tree. This tree can also utilize as a compressed database. The association rule mining is performed on the compressed database with the help of this FP tree. This indicates that the dataset needs to be inspecting once. Also this algorithm does not require the candidate item set generation. So, in comparison with Apriori algorithm, it is much better in terms of efficiency [10] .But like other algorithms it also have certain disadvantages, it generates a large number of conditional FP trees. It generates this FP trees

recursively as a procedure of mining. So the efficiency of the FP growth algorithm is not reasonable. But in proposed improved FP tree and MFI algorithm no need to generate conditional FP tree[11] because the recursive element is separately stored in a different table. That reduces the existing bottleneck of the FP growth algorithm.
Many modified algorithm and technique has been proposed by different authors. Such as FP- tree and COFI based approach is proposed for multilevel association rules. Here except the FPtree, a new type of tree called COFI- tree is proposed [7] .An Apriori based data mining technique is described at [3] .we use that example as the input of our proposed MFI algorithm and it is easily understandable that the new approach collect the association rule more efficiently.

There are other related work in XML security. Alban Gabillon et al. [16] applies XSLT transformation technology to generate user's view of required XML document. A priority number is used to solve permission confliction. The authorization rule is in the format of (subjects, objects, access, priority). Subjects are presented in XML Subject Sheet, a XML document. The objects in this model are based on XML instance document.

In 2000, Han et al. proposed the FP-growth algorithm—the first pattern-growth concept algorithm. FP-growth constructs an FP-tree structure and mines frequent patterns by traversing the constructed FPtree. The FP-tree structure is an extended prefix-tree structure involving crucial condensed information of frequent patterns.
FP-tree structure: The FP-tree structure has sufficient information to mine complete frequent patterns. It consists of a prefix tree of frequent 1-itemset and a frequent-item header table. Each node in the prefix-tree has three fields: item-name, count, and node-link.
Construction of FP-tree: FP-growth has to scan the TDB *(*Transactional Database*)* twice to construct an FP-tree. The first scan of TDB retrieves a set of frequent items from the TDB . Then, the retrieved frequent items are ordered by descending order of their supports. The ordered list is called an F-list. In the second scan, a tree *T* whose root node *R* labeled with "null" is created. Then, the following steps are applied to every transaction in the TDB . Here, let a transaction represent [p\P] where *p* is the first item of the transaction and *P* is the remaining items.
In each transaction, infrequent items are discarded.
Then, only the frequent items are sorted by the same order of F-list.

Advantages and Disadvantages

This method is advantageous because, it doesn't generate any candidate items. It is disadvantageous because, it suffers from the issues of special and temporal locality issues.

A) FP-Tree (Frequent Pattern Tree)

A tree structure[14] in which all items are arranged in descending order of their frequency or support count. After constructing the tree, the frequent items can be mined using FP-growth.

(a) Creation of FP-Tree

First Iteration: Consider a transactional database which consists of set of transactions with their transaction id and list of items in the transaction. Then scan the entire database. Collect the count of the items present in the database. Then sort the items in decreasing order based on their frequencies (no. of occurrences).

B) *Second Iteration*

Now, once again scan the transactional database. The FP-tree is constructed as follows. Start with an empty root node. Add the transactions one after another as prefix subtrees of the root node. Repeat this process until all the transactions have been included in the FP-tree. Then construct a header table which consists of the items, counts and their head-of-node links.

C) *Finding Frequent Patterns from FP-Tree*

After the construction of FP-tree, the frequent patterns can be mined using an iterative approach FP-growth. This approach looks up the header table and selects the items that support the minimum support. It removes the infrequent items from the prefix-path of an existing node and the remaining items are considered as the frequent itemsets of the specified item.

Advantages and Disadvantages

This method is advantageous because, it doesn't generate any candidate items. It is disadvantageous because, it suffers from the issues of special and temporal locality issues.

Our Approach

Proposed Improved Algorithm

A) FP-tree structure

The FP-tree structure has sufficient information to mine complete frequent patterns. It consists of a prefixtree of frequent 1-itemset and a frequent-item header table. Each node in the prefix-tree has three fields: *item-name*, *count*, and *node-link*.

- ***item-name*** is the name of the item.
- ***count*** is the number of transactions that consist of the frequent 1-items on the path from root to this node.
- ***node-link*** is the link to the next same itemname node in the FP-tree. Each entry in

the frequent-item header table has two fields: *item-name* and *head of node-link*.

- ***item-name*** is the name of the item.
- ***head of node-link*** is the link to the first same item-name node in the prefix-tree.

B) Construction of FP-tree

FP-growth[16] has to scan the *TDB* twice to construct an FP-tree. The first scan of *TDB* retrieves a set of frequent items from the *TDB* . Then, the retrieved frequent items are ordered by descending order of their supports. The ordered list is called an F-list. In the second scan, a tree *T* whose root node *R* labeled with "null" is created. Then, the following steps are applied to every transaction in the *TDB* . Here, let a transaction represent [p\P] where *p* is the first item of the transaction and *P* is the remaining items.

The function insert_tree (p\P,R) appends a transaction [p\P] to the root node *R* of the tree *T* . Pseudo code of the function insert_tree (p\P,R) is shown.

Transaction ID	Items	Frequent Items
100	A, B, C	A, B, E
200	B, D	B, D
300	B, C	B, C
400	A, B, D	A, B, D
500	B, C	B, C
600	A, B, C, E	A, B, C, E
700	A, B, C	A, B, C

B	7
A	4
C	4
D	2
E	2

```
Input: Transactional database
Output: Improved FPTree
Create the root of tree R
Initially R=NULL
-In each transaction, infrequent items are discarded.
-Then, only the frequent items are sorted by the
same order of F-list.
let N be a direct child node of R, such that N 's
item-name = p 's item-name.
if ( R has a direct child node N ) {
increment N 's count by 1.
}
else{
create a new node M linked under the R .
set M 's item-name equal to p .
set M 's count equal to 1.
}
Recursively call insert_tree ( P ,N).
}

for each item i in FP-tree where (i! = R) do
        if supprt is equal to frequency of item then
                frequency of item set = S
        generate item set P with the frequency of
item set
else if support is greater than frequency then
        frequent item=frequency + count
else
        Generate item set all possible combination
of item and node in FPTree.
End for
End
```

Conclusion and Future Work

Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data sets. In recent years, XML is a widely used data representation and storage format over the web and hence the issues of data quality and the task of data mining processes are getting significant attention to the database community. Mining XML data from the web is becoming increasingly important. XML can be used to represent unstructured, structured and complex data. To date, the famous Apriori algorithm to mine any XML document for association rules without any pre-processing or post-processing has been implemented using only the XQuery language which is costly. But the algorithm only can mine the set of items that can be written a path expression for. This paper propose that extracting association rules from XML documents without any preprocessing or postprocessing using XML query language XQuery is possible and analyze the XQuery implementation of the efficient First Frequent method-tree based mining method, First Frequent Pattern-growth, for mining the complete set of frequent patterns by pattern fragment growth. First Frequent Pattern-tree based mining adopts a pattern fragment growth method to avoid the costly generation of a large number of candidate sets and a partition-based, divide-and-conquer method is used. It will increases the mining efficiency and also takes less memory.

In future work we will implement the algorithm and develop the data mining tool for XML data mining.

References

[1] Arun K Pujai "Data Mining techniques". University Press (India) Pvt. Ltd. 2001

[2] J. Han and M. Kamber.Data Mining: Concepts and Techniques. Morgan Kaufman, San Francisco, CA,2001.

[3] Qin Ding and gnanasekaran Sundaraj, " Association rule mining from XML data", Proceedings of the conference on data mining.DMIN'06

[4] Jayalakshmi.S, Dr k. Nageswara Rao, "Mining Association rules for Large Transactions using New Support and Confidence Measures", Journal of Theoretical and applied Information Technology, 2005.

[5] R Srikant, Qouc Vu and R Agrrawal. "Mining Association Rules with Item Constrains". IBM Research Centre, San Jose, CA 95120, USA.

[6] Ashok Savasere, E. Omiecins ki and Shamkant Navathe"An Efficient Algorithm for Mining Association Rules in a Large Databases". Proceedingsof the 21st VLDB conference Zurich, Swizerland,1995.

[7] J. Han, J. Pei, and Y. Yin, "Mining Frequent Patterns without Candidate Generation", Proceedings of the ACM SIGMOD, Dallas, TX, May 2000, pp. 1-12.

[8] C. Silverstein, S. Brin, and R. Generalizing Association Rules to Dependence Rules," Data Mining and Knowledge Discovery, 2(1), 1998, pp 39–68

[9] A.B.M.Rezbaul Islam, Tae-Sun Chung, An Improved Frequent Pattern Tree Based Association Rule Mining Technique, 2011 IEEE, pp- 978-985

[10] R. Agrawal, T. Imielinski, and A. Swami.. "Mining association rules between sets of items in large databases". In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pages 207-216, Washington, DC, May 26-281993.

[11] Qihua Lan,Defu Zhang, Bo Wo, , "A new algorithm for frequent itemset mining based on apriori and FP-tree", Global Congres on Intelligent System,2009.

[12] Jiawei Han, jian pei, and Yiwen Yin, "Mining frequent patterns without candidate generation" paper id : 1 196, SIGMOD '2000.

[13] A.B.M. Rezbaul Islam, Tae- Sun Chung, „ An improved Frequent Pattern Tree

[14] Muthaimenul Adnan and Reda Alhajj, "A Bounded and Adaptive Memory-Based Approach to Mine Frequent Patterns From Very Large Databases" IEEE Transactions on Systems Management and Cybernetics-Vol.41,No. 1,February 2011

[15] W. Cheung and O. R. Zaiane, "Incremental mining of frequent patterns without candidate generation or support constraint," in Proc. IEEE Int.Conf. Database Eng. Appl., Los Alamitos, CA, 2003, pp. 111–116.

[16] J. S. Park, M.-S. Chen, and P. S. Yu, "An effective Hash-Based Algorithm for Mining Association Rules",Proceedings of the ACM SIGMOD, San Jose, CA, May1995, pp. 175-186.

[17] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, "Dynamic Itemset Counting and Implication Rules for Market Basket Data", Proceedings of the ACM SIGMOD, Tucson, AZ, May 1997, pp. 255-264.

[18] Virendrakumar Shrivastava, Dr.p arveenkumar and DR. K.R.pardasani, " FP-Tree and COFI Based Approach for Mining of Multiple Level Association Rules in Large database" , IJCSIS, International Journal of Computer Science and Information Security,Vol.7 No. 2,2010

[19] S. Brin, R. Motwani, J. D. Ull lman, and S. Tsur, "Dynamic Itemset Counting and Implication Rules for Market Basket Data," GMOD International Conference on Management of Data, Tucson, AZ Proceedings of the ACM SI Z, USA, May 1997, pp. 255–264.