A Novel Coverage Pattern Mining Method for Unordered Tree

Ying Xia[1a], Hong-Xu Li[2b]

[1] Research Center of Spatial Information System Chongqing University of Posts and Telecommunications Chongqing, China
[2]Research Center of Spatial Information System
Chongqing University of Posts and Telecommunications Chongqing, China
[a] xiaying@cqupt.edu.cn, [b]565268915@qq.com

Abstract: Unordered tree is widely used for semi-structured data modeling, coverage pattern mining on it has benefit for finding frequent subtrees without redundant information, so that improve the efficiency of subsequent knowledge discovering. A coverage pattern mining method MCRP is proposed. Firstly, all candidate subtrees are generated on the basis of maximum prefix coding and edge extension.Then coverage patterns are output by introducing δ'- coverage concept. Compared with traditional algorithms such as mining frequent closed tree patterns and maximal frequent tree patterns, the proposed method can output fewer frequent subtrees in the case of preserving all the frequent subtree information, and has a certain superiority in processing efficiency.

1. Introduction

With the development of Internet, a large number semi-structured data such as Web pages and XML documents continuous emerged. Modeling these data with unordered tree and mining frequent subtrees can effectively find the hidden knowledge. At present, many frequent subtree mining algorithms have been proposed, such as FP-tree tree [1] and PFP-tree[2] mine frequent subtrees based on the idea of projection, while EvoMiner [3] and MCFP-tree[4] adopt the ideas of Apriori. However, these algorithms usually output large scale frequent subtrees with redundant information which affect the efficiency of subsequent processing. In order to reduce the output scale of frequent subtrees, FBMiner[5] is proposed to mine frequent closed tree and MFTM[6] is presented to mine maximal frequent subtree. Although these algorithms can effectively reduce the output scale of frequent subtrees, they lose some structure information of frequent subtrees.

Aiming at the shortcomings of traditional methods, this paper proposes MCRP algorithm for mining the coverage pattern. MCRP first encodes an unordered tree by code rule based on width and number of children (WANOC), which will improves the computational efficiency of subtree support degree. On the basis of WANOC, MCRP uses edge extension method based on maximum prefix coding sequence to generate candidate subtrees, which ensures that all the frequent subtrees can be found without information omitted. In addition, a new concept of δ'- coverage is proposed based on δ-coverage[7] to judge whether a subtree is satisfied the requirement of coverage pattern for output.

The paper is outlined as follows: Section 1 introduces encoding rule of WANOC. Section 2 gives detailed description of MCRP algorithm. Section 3 presents the experimental results. Conclude is drawn in Section 4.

2. Encoding Rule of Wanoc

Defination1. Tree Support Degree And Frequent Subtree

Suppose that $\mathrm{D}=\{\mathrm{T1},\mathrm{T2}...\mathrm{Tn}\}$ is a set of unordered tree. Given a unordered tree T, the support degree of T is as follows:

$$\mathrm{Dsupp(T)}=\frac{|\{\mathrm{T'}|\mathrm{T}\subseteq\mathrm{T'},\mathrm{T'}\in\mathrm{D}\}|}{|\mathrm{D}|}$$

|D| represents the size of the tree set. Given a minimum support threshold $\mathrm{min\_sup}\in[0,1]$, if $\mathrm{Dsupp(T)}\geq\mathrm{min\_sup}$, then T is a frequent subtree on D.

In order to improve the computational efficiency of subtree support degree, an encoding rule named WANOC (the code of width and number of children) is proposed based on traditional string coding[8]. In the rule, each tree node is denoted by a triplet (index(v), range(v), tag(v)), index(v) denotes the position of node v in the width-first sequence of a tree, range(v) denotes the position range of the children of node v, tag(v) denotes the label of node v. By breadth-first traveling of tree, we can get the code of width and number of children of a tree.

Due to the disorder between the brothers in an unordered tree, we order nodes with sibling relationship by the dictionary order of their tag in order to make WANOC more standardized. For the sample tree T and t in Figure I, their WANOC codes are shown in TABLE I. By matching the WANOC code of given subtree with all existing trees, it is easy to count the frequency of subtree, so that calculate the subtree support degree. For example, when detecting wether tree t is included in T, we firstly find node A in WANOC code of T, and then search wether node C and D are included in range(A) , it is unnecessary to match each node one by one.

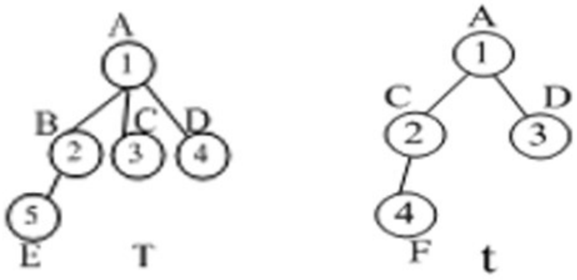

Figure I. Sample trees

Table 1 The wanoc code of sample trees

Tree	the WANOC code
T	(1,(2,4),A)-(2,(5),B)-(3,(),C)-(4,(),D)-(5,(),E)
t	(1,(2,3),A)-(2,(4),C)-(3,(),D)-(4,(),F)

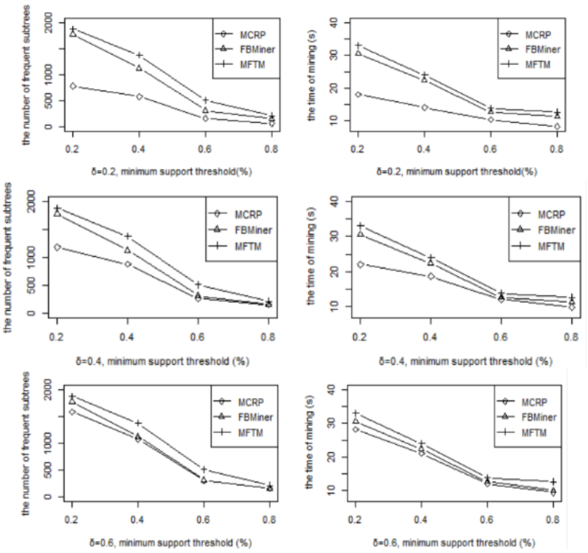

Figure 2. Experiment result

3. Mcrp Algorithm

Defination 2. Maximum Prefix Coding

Suppose an unordered tree is encoded by WANOC, the code of remaining nodes after the last one is removed is called maximum prefix encoding of the tree.

Defination 3. δ'-Coverag

Assuming that T and T' are both frequent tree, stipulating T is δ'-covered by T' if $T \subseteq T'$, the Jaccard distance [9] between T and T' is less than δ', $H(T') \leq H(T) + 1$ and $W(T') \leq W(T) + 1$. H (T) and W (T) denote the height and width of T respectively. The Jaccard distance formula is as follows:

$$D(T, T') = 1 - \frac{|S(T) \cap S(T')|}{|S(T) \cup S(T')|}$$

S(T) represents the numbers of T in D.

The algorithm includes two core steps. The first step is to generate the candidate subtrees using the edge extension method based on the maximum prefix encoding and to determine whether the candidate subtrees are frequent subtrees. In the second step, the algorithm outputs coverage patterns that are satisfied with the condition of δ'-coverageon the basis of the frequent subtrees.

2.1 Edge Extension Based On Maximum Prefix Coding

Generating candidate subtrees usually adopts the edge extension method. Traditional methods generate candidate subtrees based on the most right path to expand [10] need to find the most right path in advance, which will affect the efficiency. According to edge extension based on maximal prefix coding rule, two subtrees with the same maximum prefix coding can be combined. The more candidate subtrees are generated, the more original data information is retained. It is useful for some real applications to generate all candidate subtrees. Edge extension based on maximum prefix coding can generate all candidate subtrees. And, it can improve the time efficiency and reduce the omission of the information.

The topologies of two tree with the same maximal prefix coding may be the same or different. There are two cases for generating candidate subtrees. In the first case, the last node of two trees with the same topologies are on the same layer, if combining them, the last node of each tree might be sibling or parent-child relationship in the candidate subtree. In the second case, the last node of two trees with the different topologies might be on the same layer or not, if combine them, overlapping maximum prefix coding of two trees and the last node of the two tree keep the relative position unchanged.

2.2 Output Coverage Pattern

In order to reduce the output scale of frequent subtrees and improve the efficiency of subsequent operations, all frequent subtree patterns can summarized by a small set of coverage patterns. Domain experts only need to analyze this small set of coverage patterns to gain all information in the data set. If tree T is δ'-covered by tree T', the algorithm will only output T' instead of T.

The coverage pattern is on the basis of concept of δ'-coverage. In order to make the topology of coverage pattern as similar as possible, stipulated that the absolute value of the difference of node number between the coverage pattern and its corresponding covered pattern can not exceed 1. If the height of T' is greater than the height of T by 1, the algorithm adds the label 'H' to the front of the coding of T', this means that T' covered a tree with the height of 1 smaller than its own. Domain experts can restore the T that is not exported by removing the last node of the last layer of the

T'. If the weight of T' is greater than the weight of T by 1, the algorithm adds the label ‘W’ to the front of the coding of T', this means that T' covered a tree with the weight of 1 smaller than its own. Domain experts can restore the T that is not exported by removing the most right node of T'. In the same way, if the width and height of T' was larger than that of T by 1, we label on the front of the coding of T' with ‘HW’.

2.3 Pseudo Code Of Algorithm Mcrp

```
Algorithm MCRP(D, min_sup, $\delta'$)
Input: data set D, min_sup, Jaccard distance threshold $\delta'$
Output: coverage pattern set of frequent subtrees CS
//$S^k$ denotes a set of the frequent subtrees with k nodes
//$L(S^1)$ denotes the size of $S^1$
//$S^1(i)$ denotes the i-th tree of $S^1$
//countNum(T) is used to count the frequency of T
Traverse the data set D, count the frequency of each node and M which denotes the max node number of tree in D, put frequent nodes in S1.
for(int i=0; i<L(S1)-1; i++)   // get all 2-frequdent subtrees
    for(int j=i; j<L(S1)-1; j++)
        the candidate subtree T obtained by combining S1 (i) and S1 (j)
        if $(\mathrm{countNum}(T)/|D| \geq \mathrm{min\_sup})$
            S2←T;
    end for
end for
for (int i=2; i<=M; i++)   // get all frequent subtrees
    CS←GenerateCorageTree(Si, $\delta'$);
end for
return CS;
```

```
Function GenerateCorageTree(Sk, $\delta'$)
if (L(Sk)>= 2)
    for (int i=0; i<L(Sk)-1; i++)
        for (int j=i+1; j<L(Sk); j++)
            if (Sk(i) and Sk(j) have the same maximum prefix coding)
                if (Sk(i) and Sk(j) have the same topologies)
                    the candidate subtree T obtained by combining Sk (i) and Sk(j)
                else
                    the candidate subtree T obtained by overlaping the maximum prefix coding of S1(i) and S1(j)
                end if
                if $(\mathrm{countNum}(T)/|D| \geq \mathrm{min\_sup})$
                    Sk+1←T;
                    if $(1-|\{Tr|(T \subseteq Tr)\ \&(Sk(e) \subseteq Tr), Tr \in D\}|/|\{Tr|(T \subseteq Tr)\ |\ (Sk(e) \subseteq Tr), Tr \in D\}| \geq \delta')$ //e=i, j
                        if(H(T)>H(Sk(e)))
                            CSK←Tag(‘H’, T);   // Tag(‘H’,T) denotes that the result of add the label ‘H’ in the front of the code(T)
                        else if(W(T)>W(Sk(e)))
                            CSK←Tag(‘W’, T);
                        else if(H(T)>H(Sk(e)) && W(T)>W(Sk(e)))
                            CSK←Tag(‘HW’, T);
                        end if
                    end if
        end for
    end for
return CSK;
```

The most time-consuming part of algorithm MCRP is the process of generating frequent subtrees and outputting the coverage pattern. In order to check all the possible generating of candidate subtrees, function GenerateCorageTree (S^i,δ') needs to traverse k-frequent subtrees when generated (k+1)-candidate subtrees. But in practice, the time complexity of MCRP algorithm is not so high because that not any two k-frequent subtrees satisfies the conditional to generate (k+1)-candidate subtrees.

4. Experiment Result

In the experiment, we compared MCRP algorithm to FBMiner that is used to mining requent closed tree and MFTM that is used to mining maximal frequent subtrees respectively. Experimental data comes from the website (http://more.datatang.com/data/12122), which includes 5000 XML documents. All experiments were done on the environment ,which is shown in TABLE II. As what shown in Figure 2, the left column of the following experimental chart shows the number of frequent subtrees that the above three algorithms outputs in case of different support thresholds min_sup when the Jaccard distance threshold δ' is 0.2 , 0.4 and 0.6 respectively. The right column corresponds to the calculation time of three algorithms on the basis of the different δ'. Experimental results show that both the output scale of frequent subtrees and consuming time of MCRP algorithm are less than FBMiner and MFTM.

Table2 Experimental environment

Operating Ssytem	Windows 8.1
CPU	Intel Core I7
RAM/Hard Disk	8GB/1TB
DB/Drawing Tool	MYSQL/R
XML Parsing Tool	DOM4J
Programming Langue	JAVA

5. Conclusion

This paper focuses on the method of mining frequent subtrees on unordered trees. Firstly, an encoding method named WANOC is proposed, which keeps the position information of nodes and their children's nodes, so that the efficiency is improved when the subtree support degree is calculated. And then, the maximum prefix coding is proposed to generate all the candidate subtrees on the basis of WANOC, which avoids the omission of frequent subtree information. In the end, in order to avoid outputing frequent subtrees in large scale and affecting the efficiency of subsequent operations , the coverage patterns is output on the basis of δ'coverage concept. Experimental results show that the MCRP algorithm is effective in the frequent subtree output scale and time cost.

References

[1] Yakop, M.A.M., S. Mutalib and S. Abdul-Rahman, Data Projection Effects in Frequent Itemsets Mining. 2015: Springer Singapore. 23-32.

[2] Malviya, J., A. Singh and D. Singh, An FP Tree based Approach for Extracting Frequent Pattern from Large Database by Applying Parallel and Partition Projection. International Journal of Computer Applications, 2015. 114(18): p. 1-5.

[3] Deepak, A., et al., EvoMiner: frequent subtree mining in phylogenetic databases. Knowledge and Information Systems, 2014. 41(3): p. 559-590.

[4] Wu Qian , Luo Jianxu, An Improved Search Algorithm for Compressed FP-Tree.Computer Engineering and Design, 2015 (7): 1771-1777.

[5] Feng, B., et al. A new method of mining frequent closed trees in data streams. in International Conference on Fuzzy Systems and Knowledge Discovery, Fskd 2010, 10-12 August 2010, Yantai, Shandong, China. 2010.

[6] Yang Pei , Tan Qi, Maximal Frequent Subtree Mining and its Application, in Computer Science, 2008.35 (2): 150-153.

[7] D. Xin, J. Han, X. Yan, et al. Mining Compressed Frequent Pattern Sets[C]//31th International Conference on Very Large Data Bases (VLDB). Santiago de Chile,Chile:VLDB Endowment,2005:709–720.

[8] Liu, L. and J. Liu. Mining Frequent Embedded Subtree From Tree-Like Databases. in International Conference on Internet Computing & Information Services. 2011.

[9] A. K. Jain, R. C. Dubes. Algorithms for Clustering Data[M]. 1st ed., Prentice Hall,1998:366–367

[10] Han, K., et al. Constrained Frequent Subtree Mining Method. in International Conference on Digital Home. 2014.