International Journal of Intelligent Information and Management Science
ISSN: 2307-0692, Volume 6, Issue 3, June, 2017

SFCS Algorithm of legal Data Mining in Cyberspace

Huifeng LONG
Hunan City University, Yiyang Hunan 413000, CHINA

Abstract: For FCS does not have the ability to find the trend of the sub-structure space, this article proposes the SFCS algorithm. SFCS maintains the frequency of version changes at the time point, and also it maintains the frequency of vertical degree and the breadth changes at the space point. SFCS mining solves the problem that FCS unable to solve the changes of sub-structure spatial extent, which can be applied in application area, which concerns the spatially frequent changes of the sub-structure of the semi-structured data. Experimental results show that: the proposed algorithm has good scalability and can maintain the frequency of mining results.

Keywords: Data Exchange; Internet; Documents; Dynamic Variability

1. Introduction

The generation and rapid development of extensible Markup Language (XML) bring new vigor and vitality to the network. XML can be used directly on the Internet, and it can transparently interact with HTTP and other major already existed protocols; it supports a wide variety of applications; it is evolved by Standard Generalized Markup Language (SGML), also it is compatible with SGML, which can separate the "abstract" data concepts with specific expression method to achieve “write once, several spreads”. XML is a set of rules to definite the semantic tags; these tags separate the documents into many parts and identified them. It is also the meta-markup language that defines the syntax language, which is used to definite the markup language which is other domain-specific related, semantic and structured. Users can define their tags needed, which has more advantages than HTML[1-3].

Once XML technology appeared, it shows a significant advantage in the field of data storage, data exchange, and others. It quickly became the unified data format standards independent of vendor and platform, so it has been widely used in e-commerce, personalized publishing, mobile communications, online education fields, electronic document exchange and so on[4-6]. Accordingly, there are some problems in these areas like how to effectively store, manage and operate the XML data. In this paper, the application of order information processing for e-commerce products is used as an example. Through the research of XML Document Object Model (DOM) technology, the manipulation and processing of XML data are achieved.

The query problems of continuously uncertain XML data can be divided into query node matching and calculation of uncertainty probability. Currently the researchers have proposed many structural join algorithms based on merge or non-merger to deal with the inquiry problem on the matching stage, such as the direct merging algorithm MPMGJN. Due to a large number of intermediate results generated by repeatedly scanning the list of documents nodes, the algorithm has a larger time and space cost. Twig Stack algorithm and Twig 2 Stack algorithm realized the completely and partial whole twig pattern matching, which reduces the connection of binary relations, but it still need to be merged and opens the large storage space. LIU qin et al make improvement on the research basis of Twig 2 Stack and proposed the Twig List algorithm. They use the linked to replace the level of stack to reduce the memory consumption and improve the efficiency of the query match, but the algorithm is more complex. On the stage calculating the uncertainty probability, Scholte T et al proposed the integration method (IM) and rectangle approximation method (RA).

The mining of frequent substructure is one of important research topics in mining of the semi-structured data. The main idea is extracting pointed sub-structures with frequent features from the given XML documents, which can be divided as static XML frequent structure mining and dynamic XML frequent structure mining. Rusu et al proposed the method, which concentrated mines the association rules in the dynamic XML documents; Bifet et al proposed the method Inc Tree Nat, which mines the frequent sub-trees in a dynamic tree data stream. Zhao et al proposed the H-DOM model, the mining frequently changing structure FCS based on method H-DOM model and the method mining frequent changing semantic structure FASST. Zhao et al proposed some new knowledge of incremental mining for XML, including frequent

changes structure FCS, frozen structure FS, association rules, changing patterns etc. Also they introduced the incremental mining of XML and the related applications. The mining of frequent changes structure FCS refers to the sub-structure which mines frequently and has significant in the history of XML document changing, while FCS mining has no types of changes and trends of changes for binding XML. It mines the FCS in highly dynamic data set; the mining results include a large number of sub-structures, which are inconvenient for users to use and analysis. Chen et al proposed the approach mining the maximum frequent changes substructure FCSP and on this basis proposed the method to mine the frequent changes consistent substructure FRACTURE. Frequent changes consistent substructure FRACTURE is the frequent changes in the frequent changes structure set of the history changes of XML documents and having frequent changes simultaneously; it is the subset of FCS. Chen et al achieved the mining methods of FRACTURE based on Apriori algorithm and FP growth algorithm. Chen et al also proposed the concept of Maximal FRACTURE to eliminate a lot of redundancies of FRACTURE mining results.

Currently, there is no frequent changing structure method according to the changing situation of the the structure of the XML sub-space, so the research of mining method for XML space frequent changing structure is meaningful. The FCS does not have the ability to find out the developing trends of sub-structure space. For the existing problem of FCS, this thesis proposes the SFCS algorithm. SFCS is the sub-structure of the XML documents when the time dimension meets the frequently changing conditions and the trends of space dimension can achieve the users' desire.

2. Data Model of XML Document

As early as 20 years ago, e-commerce based on Electronic Data Interchange (EDI) began to appear between the large enterprises and their business partners, but because of its high costs caused by high complexity, it is hindered. The good performance of XML like good data storage format, scalable, highly structured, network features and ease of programming solved the problems encountered by EDI, especially in logistics, warehousing and other applications. XML provides a fully automated solution. Figure 1 is the instance of the codes' product orders with XML format (including order number, customer name, time, address and order contents), and the file name of XML document is set as: ordeLxml. As can be seen from the above example that XML document is a segment hierarchical plain text file constituted by nodes, and its content and form apply the separation and support the Chinese encoding, and also it allows users to customize the structure. Logical structure includes the statements, processing instructions, elements, attributes, comments, etc.; user information is generally stored in the text and attributes of the element. Elements are used to identify different segments of the XML file. The use of elements makes the operation like data express, storage and transfer can be automatically processed; property is also a mechanism added described information for the elements.

2.1. Ordered Tag Tree

This article uses the symbol "T" to represent the ordered tag tree corresponding to XML document; "layer (node)" defines the number of layers of the tree T's nodes, that is, the path length (the number of root layers is specified as 0); "n _ num $(T, 1)$" is the number of nodes of the tree T's first layer; "child (v)" is the direct number of children of node v, i.e. the degree of node v. The symbol "depth (T)" and "width (T)" are used to separately denote the depth width of the tree T. The depth of tree is defined as the maximum path length of the leaf nodes; depth (T) = max {layer (leaf i) | lea f i as tree T's leaf node}; the width of the tree is defined as the maximum value of the tree node's layers; $width(T)=\max\{n\_num(T,i) \mid 0<idepth(T)\}$. The "Merge structure" of tag tree's T1 and T2 can be expressed with the symbol "T 1 ∪ T 2"; the node set of T1 ∪ T2 is the union set of T 1 and T 2; the edge set of T 1 ∪ T 2 is the union set of T 1 and T 2, which is as shown in Figure 2.

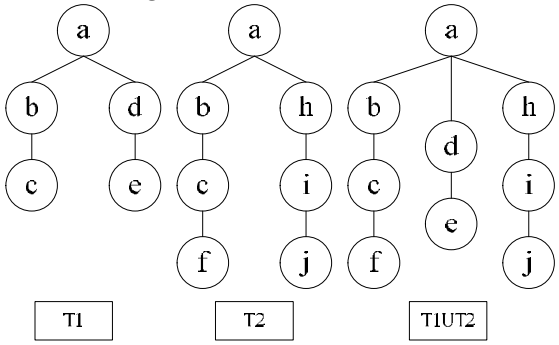

Figure 1. Two ordered tag trees and its consolidated structure

2.2. Attribute Definition of Spaces

The ordered rag tree of the XML document provides leaf insertion, deletion leaves, insertion of inside, inner deletion, updating, moving, inserting sub-tree and deleting sub-trees eight. Wherein the first five operations are the basic editing operations and the last three operations are the composite editing. Complex editing operations are consisting of several basic editing operations.

Definition1. Vertical degrees of space. The tree T's vertical degree of space VD (T) comprehensively represents the tree's vertical depth extent. Let the tree T's collection

of leaf nodes as {leaf 1, leaf 2, ..., leaf n}; the tree T's vertical degree of space VD (T) is defined as

$$VD(T)=\frac{1}{n}\times\sum_{i=1}^{n}\frac{layer(leaf_i)}{depth(T)}\times\Delta(depth(T)) \quad (1)$$

$$\frac{1}{n}\times\sum_{i=1}^{n}\frac{layer(leaf_i)}{depth(T)} \quad (2)$$

As can be seen from the definition 1, VD (T) is consisted by the product of the two parts; formula (2) partially presents the degree of closing of the tree and tree depth. The number of branches with their length is close with depth (T) are increasing and the vertical space of tree T is larger. Since this section does not reflect the effect of the depth of the tree depth (T) itself to the vertical degree of space, the followings will occurs: Let $T1$ and $T2$ are two XML documents corresponding ordered tag tree models as shown in Figure 2. Tree $T1$ has two leaf nodes and its path lengths is 2; tree T2 also has two leaf nodes, and its path length is 3, and then the first part of vertical space of tree $T1$ and $T2$ is :

Vertical space's first part value of Tree

$$T_1:\frac{1}{2}\times\left(\frac{2}{2}+\frac{2}{2}\right)=1 \quad (3)$$

Vertical space's first part value of Tree

$$T_2:\frac{1}{2}\times\left(\frac{3}{3}+\frac{3}{3}\right)=1 \quad (4)$$

To reflect the impact of the depth of the tree to the vertical space of the tree-degree, the coefficient Δ proportional to the tree depth in the definition of VD (T) is introduced, i.e., Δ (depth (T)). Δ should meet the following two conditions:

Condition 1.0 <Δ (depth (T)) <1;

Definition3. The degree of changes of the version space. Version spatial variation measures the special changing frequency in the history process of substructure S from the time the angle, i.e., it is used to present n versions in the XML documents and the ratio of special changes of sub-structure S. The degree of version changes is defined as follows:

$$VSCD(S)=\frac{\sum_{i=1}^{n-1}v_i}{n-1}$$
$$v_i=\begin{cases}1, SSCD_i(S)\neq 0\\ 0, SSCD_i(S)=0\end{cases} \quad (5)$$

Definition6. The degree of spatial variability. Spatial variation presents the ratio of the number of users concentration of the substructure S's SSCD to the number of changes in S space in the changing process of XML. Spatial variation is defined as follows:

$$SCD(S,a)=\frac{\sum_{i=1}^{n-1}d_i}{(n-1)\times VSCD(S)} \quad (6)$$
$$d_i=\begin{cases}1, SSCD(S)\geq\alpha\\ 0, SSCD(S)<\alpha\end{cases}$$

2.3. Spatial Variation SC-DOM

Spatial variation SC-DOM is the tree model with the tree's horizontal extent and vertical extent, and it is an extension of DOM model. SC-DOM can record the dynamic changing process of XML space, which can avoid the high cost of XML documents when comparing the differences and facilitate the space changing structure SFCS.

SC-DOM is a quintuple (V, E, SISs, f, r). Wherein: V is the set of tree nodes; E is the set of edges; E = {(x, y) | x, y ∈ V} means that x is the father node of y; SISs is the collection of spatial information structure (SIS) and each special information structure SIS (v) stores the width and depth information of sub-tree with v as the root; f is the mapping f: v → SIS (v) of node v to SIS (v), wherein v ∈ V, SIS (v) ∈ SISs; r represents the root node. Spatial information structure SIS (v) stores the width and depth information of the tree when sub-tree changing with the v as the root nodes. SIS (v) is defined as the six-tuple: (V, N, P, PL, D, W), wherein V is the version number; N is the total number of nodes of the sub-tree (does not including the root nodes); P is the number of paths of sub-tree, that is, the number of the sub-tree's leaf nodes; PL is the sum of sub-tree's path length of each path.

$$PL=\sum_{i=1}^{P}layer(leaf_i) \quad (7)$$

3. SFCS Algorithm

SFCS discovery algorithm based SC-DOM traverses the ordered tag tree of XML once, and it's complexity of the algorithm is O (| T |), and the algorithm is as follows (C + + description).

Take the SC-DOM shown in Figure 3 for example, set the threshold λ = 0.5, α = 0.05, β = 0.05, γ = 0.6, the SFCS mining process based on SC-DOM is shown as follows: firstly, each node of SC-DOM is traversed and they are iterated n-1 times. This paper takes node a and the node c as instance, and then ask vertical degree VD of each node versions, the width of the space HD space and the corresponding longitudinal increments ΔVD and breadth increments ΔHD. According to the 5-10 line of the SFCS mining algorithm, the longitudinal degree and their increments and width of space and their increments of node a and node c are shown in Table 1.

void SFCSMining($\lambda,\alpha,\beta,\gamma$){

1. for(XML all nodes of labeled ordered tree v){
2. int $SCDCnt$=0
3. double $VSCD$=0
4. for(int $i=1; i<n; ++i$){
5. double $VDi1 = PL_i / (D_i \times P_i) + \Delta(D_i)$;
6. double $HDi1 = N_i / (D_i \times W_i) + \Delta(W_i)$;
7. double $VDi2 = Pl_i + 1 / (D_i + 1 \times P_i + 1) + \Delta(D_i + 1)$;
8. double $HDi2 = N_i + 1 / (D_i + 1 \times W_i + 1) + \Delta(W_i + 1)$;
9. double $DeltaVD = VDi2 = VDi1$;
10. double $DeltaHD = HDi2 - HDi1$;
11. if ($DeltaVD > 0$) || $DeltaHD > 0$)
12. $VSCD += 1/(n-1)$;
13. if (($DeltaVD * \lambda + DeltaHD * (1-\lambda)) >= \alpha$
14. $SCDCnt++$;
15. }
16. if ($VSCD >= \beta$ && $SCDCnt / ((n-1) * VSCD) >= \gamma$){
17. SFCS Taking v as subtree of root.
18. }
19. }

}

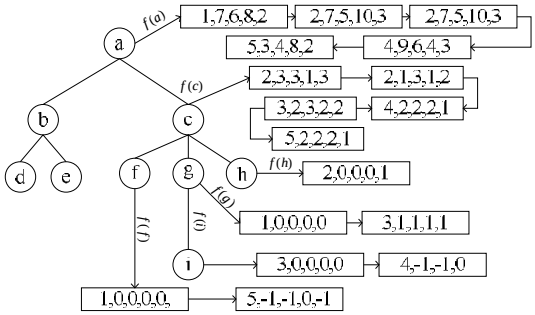

Figure 2. Dynamically updated results of SC-DOM

4. Experimental Simulation and Analysis

4.1. Experimental Environment and Setup

Experiments is completed in Intel Core2 Quad Q9550 3.83GHz (4cores) CPU, 2048MB RAM, Windows 7 PC operating system, and they are completed by using the Java language in the Eclipse integrated development environment. In the experiment Dom4J open API library is used for parsing XML documents and DOM establishment.

Experimental data comes from the artificially generated XML data sets. Firstly human edits a standardized DTD document, which contains duplicate elements in the DTD document ("+" element, "*" element and "?" element). According to this DTD, the space size of some substructure of generated XML documents can be changed. And then the IBM XML Generator is used to verify the correctness of DTD and generate several initial XML document based on the DTD document. Finally, the version generator XML File Gen of the artificial written XML is used to generate several versions (XML File Gen randomly change the changeable elements in XML documents according to specific DTD document) according to different experimental purposes.

4.2. Results Analysis

The number of Changing version $NV = 5$ of document is unchanged and when the threshold α, β and γ values are 0.1, the performance analysis of the number of changing nodes between the adjacent versions changes are shown in Figure 4.

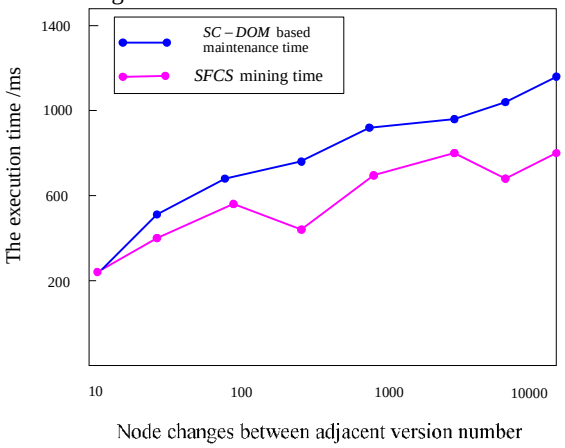

Figure 3. Impact of the number of changing nodes between adjacent versions to SC-DOM

With the increases of CR, the generation time of SC-DOM model increases, and the mining time of SFCS structure is also correspondingly increased. The building and maintenance of SC-DOM model, the calculating scale of the SFCS mining is proportional to the changing level of XML. When CR increases, the size of the model rises and the time traversing the model also is increased. With the increases of CR, the proportion of SFCS mining time to the total execution time increases. When CR = 10 000, the mining time of SFCS takes 1/3 of the total execution time; when CR = 30000, mining time of the SFCS is substantially equal to the time of building and maintenance model, i.e. it takes 1/2 of the total execution time.

Figure 5 is a comparison of the running time of the SFCS algorithms and FCS algorithms. It can be seen that when the minimum support equal to 3%, the running speed of

SFCS algorithm is faster than the FCS algorithm. However, when the minimum support is less or equal to 1%, the running speed of SFCS algorithm is faster than the FCS algorithm. Moreover, with the reduction of the support, the difference becomes bigger.

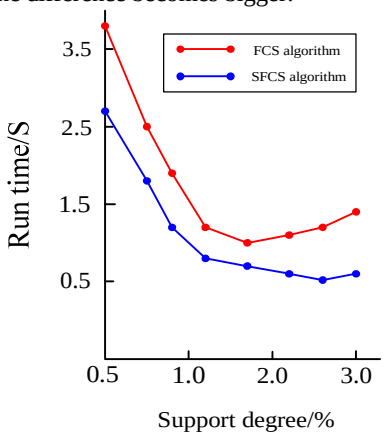

Figure 4. Running time of SFCS algorithm and FCS algorithm

When the number of version NV = 5 is the same, the number of changing nodes between the adjacent versions CR = 100 are unchanged, the relationship of the threshold α of structure spatial variation, threshold β of version spatial variation and threshold γ of spatial variation with the number of SFCS mining is shown in Figure 6.

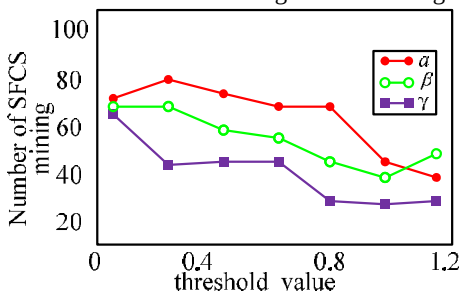

Figure 5. Relationship between three thresholds of changes with the number SFCS mining

When β = 0.1, γ = 0.1 and the structure special variation a takes different values, the number of frequently changing structure S F C S in mining space is inversely proportional to the number of threshold α; When α = 0.7, the number of mining SFCS is as 0. When α = 0.1 and γ = 0.1, the number of mining SFCS is inversely proportional to the threshold β; the number of mining SFCS decreases with the increase of threshold β. When α = 0.1, β = 0.1, the number of SFCS mining is inversely proportional to the threshold γ, which indicate that the results of the SFCS mining number are inversely proportional to the three thresholds α, β, and γ, so in practical application, users need to make a trade-off between the number of SFCS mining and the threshold α, β, and γ. The above experiments are performed in the time interval [141ms, 172ms], which shows that the changes of threshold α, β and γ do not affect building and maintenance of SC-DOM and the the running time of SFCS mining.

The time of SFCS mining is proportional to scale of the data set, and it is basically the same with the time of FCS mining, so the SC-DOM and H-DOM have good scalability. For the same data set, the number of mining SFCS is fewer than the number of FCS and the FCS set is the proper subset of set SFCS, which indicate that the SFCS itself is the FCS. The definition does not change the frequently changing structure of SFCS.

To sum up, establishment and maintenance costs of SC-DOM and the cost of SFCS mining are proportional to the size of XML documents and the number of changing nodes between adjacent versions in XML documents. Threshold α, β, and γ affect the number of SFCS mining, and the specific relationship is that the number of SFCS mining is inversely proportional to the thresholdα, β and γ. The changes of each threshold do not affect the time of SFCS mining. Algorithm has good scalability and it maintains frequency of the mining results.

5. Conclusion

SFCS algorithm maintains the frequency of version changes from the perspective of time, also it maintains the frequency of the vertical and wide changes from the perspective of space. SFCS mining solves the problem of FCS unable to deal with the level of the sub-structure's spatial changes, which can be applied to concern frequent changes of sub-structure space of semi-structured data. The experiments verify the feasibility of the algorithm.

References

[1] Muhammad J. Mirza, Nadeem Anjum. Association of Moving Objects Across Visual Sensor Networks. Journal of Multimedia, Vol 7, No 1 (2012) pp. 2-8

[2] Haiping Huang, Hao Chen, Ruchuan Wang, Qian Mao, Renyuan Cheng.(t, n) Secret Sharing Scheme Based on Cylinder Model in Wireless Sensor Networks. Journal of Networks, Vol 7, No 7 (2012) pp. 1009-1016

[3] N. P. Ramaiah, "De-duplication of Photograph Images Using Histogram Refinement", Recent Advances in Intelligent Computational Systems, 2011, pp.391-395

[4] A. V. Sreedhanya and K. P. Soman, "Secrecy of cryptography with compressed sensing," International Conference on Advances in Computing and Communications, pp. 207-210, 2012.

[5] C. Wengert, M. Douze, H. Jegou. "Bag-of-colors for Improved Image Search", In Proc. of the 19th ACM international conference on Multimedia, 2011, pp.1437-1440

[6] CHEN T W., SHEN G W., XU B H., et al.H-Code : A Hybird MDS Array Code to Optimize Partial Stripe Writes in RAID-6.// Proceedings of the 2011 IEEE International Parallel & Distributed Processing Symposium, Anchorage, Alaska, USA, IEEE Press,2011:782-793.