International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

CLUSTERING OF HIERARCHICAL DOCUMENTS BASED ON THE SIMILARITY DEDUCTION OF STRUCTURE

Prof. S. Sahunthala[1], Nandhini S[2], Poojasree K[3]

[1,2,3] *Department of Information Technology, Anand Institute of Higher Technology, Tamilnadu, India*

ABSTRACT - *Day by day the usage of hierarchical documents is increased in the internet in terms of data transmission standard because it is reliable and easier to expand or upgrade to operating systems, have new applications or a new web browsers without losing data. Clustering hierarchical document has become a crucial issue in hierarchical data mining because of its tree like structure. The existing LSPX approach (Level Structure with Parent node information of XML data) and cluster core is based on level structure consume more time. This model is to cluster the data. We propose a method that overcomes the problem that presents in the cluster core and LSPX method by reducing the number of computation needed to find the similarity calculation. This HSS approach (Hierarchical Structural Similarity) uses X-Path to determine the nodes of a document for the structural similarity calculation. It will provide a high time efficiency in finding the structure similarity of multiple hierarchical documents. The main focus of this method is to improve the clustering efficiency, complexity, performance and scalability while dealing with very big datasets which will be helpful in query processing, web mining and information retrieval.*

KEYWORDS: Hierarchical, Clustering, Mining, Structure and Similarity.

I. INTRODUCTION:

Hierarchical documents organize the information as a tree like structure. It provides the structure and content. Different domain uses hierarchy structure which will be useful to organize large number of data. The hierarchical document is the collection of elements and values where the elements are represented as a node and values are represented as a node value. Clustering hierarchical documents with respect to their structure help us to identify the different documents providing a similar kind of information. Various clustering approaches for hierarchical documents particularly for XML (Extensible Markup Language) has been proposed by considering their structure, content or a distance. Document clustering will be useful in the area of information retrieval, web mining and so on. In this paper we concentrate on XML document clustering based on the structural similarity between the documents. XML is widely used as data representation model so the documents need to be organized based on the structure and contents on the similarity between the documents. The most of the available clustering algorithms are not accurate with low clustering efficiency and scalability.

In this paper we focus on clustering hierarchical document by using X-Path is a W3C recommendation. Where X-Path is a major element in XSLT standard in query which represents the syntax for defining parts of an XML document. It retrieves path of each node in the hierarchical document which will be useful in finding the structural similarities among the documents the with their structural similarity probability values and to increase a clustering efficiency and scalability. Section II describes about the Literature survey that are mentioned below.

II. LITERATURE SURVEY:

There are many methods for clustering hierarchical (XML) documents are available based on the structure. Most of the clustering algorithm uses distance and incremental methods and X-Path are studied. This Section takes into account for the literature survey that are described below.

A.SUMMARIZATION:

This technique is referred from the article[3]. It is based on the tree edit distance of the summarization method. The tree edit distance between ordered labeled trees is the **minimal-cost sequence of node edit operations** that transforms one tree into another. The general algorithm for the tree edit distance uses a path strategy to compute the tree edit distance. For two given trees, AA and BB, a single path function computes the distances between each subtree of AA rooted in the chosen path and all subtrees of BB. The distances are stored in a distance matrix for later reuse. The precondition is that the distances between every relevant subtree of AA(according to the chosen path) and all subtrees of BB have been computed beforehand. It performs the following steps.(i)For a given pair of trees, (A,B)(A,B), look up the path in the path strategy.(ii)If the path is in AAdo the following steps, otherwise run the algorithm for (B,A)(B,A).(iii)Run the algorithm for every relevant subtree A′A′ in AA and the tree BB.(iv)Compute the single-path function for (A,B)(A,B) according to the path's type. In which the original structure is maintained even after removing the redundancies of xml tree. It consist of two phase. The first phase proposes a GENERATE-CLUSTER algorithm and the second phase proposes PARTITION-CLUSTER algorithm. To generate a new cluster (I) select a candidate node to split. (II) Further it is divided into 2 child cluster. (III) Comparing the quality of the new partition with the original partition which is obtained by splitting. (IV) If

the quality is better than the original partition, the new cluster is updated else the clusters are discarded and a new cluster is formed by splitting.

In Generate-cluster the Quality (Q) of the cluster is calculated as follows,

$$Quality(R) = \sum_{Q \in R} Pr(Q)\ Quality(Q)$$

The Quality(R) of the Partition-cluster is calculated as follows,

$$Quality\ (Q) = [Pr(Q) \sum_{f \in sc} [Pr(f|Q)^2 - Pr(f|E)^2]]$$

Where $Pr(f|Q)^2$ corresponds to relative strength of f within q
Pr(c) represents relative strength of c
D dataset

The main advantage of this method is that the structured components are addressed at the level of their hierarchy. It has the problem because the structural components are in the pre-specified form so the effectiveness of clustering algorithm is reduced.

B.CHAWATHE'SALGORITHM:

Chawathe's technique is referred from the aricle[6] and it is based on structure summaries. In this the pre-order traversal is used to traverse a tree to produce a structural summaries. The pre-order tress are transformed into a special sequence called ab-pairs. It is defined as ab-pair representation of a node is defined as the pair (a,b) where a and b are the label node and the depth of the tree. The label and the depth of an a tree is represented as p.a and p.b –pair node p respectively.

In this it is easy to assign a new incoming document to the already identified cluster. Calculating a distance for each element in xml document is not efficient process but it reduces the process of Reclustering.

C. XCLSC ALGORITHM:

XCLSC is referred from the article [4]. They have enhanced a search performance consisting of two phases. The first phase consist of Enhanced Clustering algorithm (EXCLS)" which is applied to a cluster of XML documents. This approach is an Incremental algorithm used for grouping the documents which is extended by XCLS+.

XML documents threshold is given as an input and output is obtained as a set of clusters. The procedure is (i) To cluster all the XML document.(ii) Read the next document.(iii) Find the similarity between the existing cluster and document.(iv) If the similarity is found between two objects and similarity is greater than threshold .(v) Assign the document to the existing cluster.(vi) Otherwise form a new cluster.

In the Second phase, a search process is done to form clusters. This leads to enhancement of clustering results and the search performance without the need for summarization. It improves the quality and efficiency of the clustering process but has more computations.

D. X PATTERN AND PATH XP:

In this article[6] they have used patterns to cluster the XML documents without pairwise comparisons. They put up a new pattern based clustering framework called XPattern. XPattern includes four steps that is(i.e),(i) Data transformation: It transforms input data into a representation for pattern mining according to pattern definition.(ii) Pattern mining: The transformed data is mined for patterns. (iii) Pattern clustering: The mined patterns are clustered into groups called profiles. (iv)Document assignment: Clusters to the document are assigned and represented by profiles. It also implements an algorithm called pathXP. The feature of a pathXP is XML Path. In PathXP, the feature is called as pattern if it is a maximum frequent path, if it contain a least minisup of document A, where minisup is a parameter which is defined by user. The similarity is calculated using

$$Sim(a_1,a_2) = \frac{\sum_{d \in D} [\min\{m(p_1,d),m(p_2,d)\}]}{[\sum_{d \in D} m(p_1,d)+m(p_2,d)- \min\{m(p_1,d),m(p_2,d)\}]}$$

Where, p= path
d=distance

It mines maximal frequent paths and group them into profiles detecting and excluding the outliers. Clustering of imbalanced datasets are not addressed in this approach.

let B be (N+1)×(P+1) matrix such that B[x,y] is the shortest path from (0,0) to (x.y) in the edit graph. Matrix B is called Distance matrix for G. Distance matrix is computed as, B[x, y], where 0 <x ≤ N and 0 <y ≤ P

$$B[x, y] = \min(m1,m2,m3)$$

E. CLUSTER CORE AND LSPX:

CO-LSPX is an clustering method based on cluster core, LSPX model and incremental clustering which is reffered from the article[2]. It avoids the problem of similarity calculation method, in which it calculates the similarity pair by pair using the formula,

$$Sim_{1\rightarrow 2} = \frac{[0.5 \times IW_1 + IW_2]}{[MW_1 \times Z_1]}$$

Where IW=Identweight,
MW=Modelweight.

The Cluster core is formed by (i) Extracting the centre.(ii) Generation of the cluster core.(iii) Cluster core guidance specification. It reduce the XML data clustering process time overhead but shields the disadvantage of incremental clustering. It increases the efficiency and quality of clustering results. It takes more time for clustering large datasets.

III. PROPOSED MODEL

The hierarchical documents are generally in the form of tree structure. The nodes in the document are present at the different layer of the tree structure. In the proposed model we are going to use hierarchical (XML) documents. In a directory we had provided the dataset which is a XML document. Then we retrieved the X-Path of XML document on the combination basis to compare the structure similarity where X-Path returns the exact path of each node from the start path.

Where the X-Path of all the nodes in the XML document are taken and it will be compared with X-Path of the XML nodes in the other document. This process will be continued until all the combinations of the document is compared. The structural similarity value between the documents are stored and based on the value returned by the similarity calculation and threshold limit the clustering of hierarchical documents on the structural basis is done.

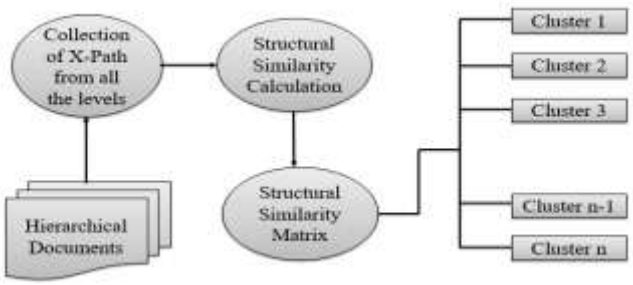

Fig -1: Hierarchical Structural Similarity Architecture

In this it first collect the X-path of first file and it collect X-path of the next file and compares it with the first collected list of X-path of file and perform structural similarity calculation then it replaces the second file with third, four and so on.

Figure 1: Hierarchical Structural similarity architecture.

$$StructSim = \frac{[XPath\ of\ doc(i) \cap XPath\ of\ doc(j)\ ]}{Total\ number\ of\ XPath\ in\ doc(i)}$$

Where i = 0 to Total number of files-1
j = 1 to Total number of files

This method of structural similarity calculation reduces the computation time by reducing the multiplicative operation that present in the cluster core and LSPX method. Where both return the same results.

After completing each structural similarity calculation we form a similarity matrix in the form mentioned below.

Table-1: Similarity matrix format

	D1	D2	D3	D4	D5	D6
D1	1	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
D2	S_{21}	1	S_{23}	S_{24}	S_{25}	S_{26}
D3	S_{31}	S_{32}	1	S_{34}	S_{35}	S_{36}
D4	S_{41}	S_{42}	S_{43}	1	S_{45}	S_{46}
D5	S_{51}	S_{52}	S_{53}	S_{54}	1	S_{56}
D6	S_{61}	S_{62}	S_{63}	S_{64}	S_{65}	1

Where D1, D2.. are doc1, doc2,...
S are th documents to do the experimentation of Hierarchical Structural Clustering

ALGORITHM:

The Structural Similarity algorithm is shown below. Where the algorithm states the method to calculate the structural similarity of any number files and add it to the matrix for clustering.

In this it will collect X-Path of files and compare it to find structural similarity between the documents and add it to structural similarity matrix with respect to the file index. This process will be continued until all the combinations of the document is compared.

```
Input     : Hierarchical(say XML) Files
Output    : Clusters based on structural similarity
Method :  StructuralSimilarity(Input, Total Number
of files n)
 {
for i=0; i<n-1; i++
   XPath1[ ]=getXPath(doc(i))
     for j=1; j<n ; j++
       XPath2[ ]=getXPath(doc(j))
       compare  XPath1, XPath2
       StructSim =  [XPath of doc(i) ∩ XPath of doc(j) ]
                    Total number of XPath in doc(i)
        Add StructSim[i][j] to Matrix
        end
       end
  clusters=HierarchicalCLustering(StructSim);
  return clusters
 }
```

Figure 2: Structural Similarity Algorithm

The below fig-3 algorithm for hierarchical clustering forms a clusters using the structural similarity matrix where the matrix is formed using the above algorithm.

```
Method : Hierarchical Clustering(StructSim);
{
  for i =0; i < n - 1; i++
    for j = 0 ; j < n ; j++
      if (i= = j)
       break;
       else if(StructSim[i][j]>=Threshold)
         Add it to the same cluster where doc(i) is cluster
        else
              Form a different cluster
     end
end
}
```

Fig-3: Hierarchical Clustering algorithm

V. EXPERIMENTATION:

Consider the following set of XML documents to perform hierarchical clustering. Where we take n number of documents to explain how the proposed hierarchical clustering works. Initially we are providing a dataset in a repository with full of XML documents.

The XML document structural similarity are calculated with different combinations of combinations like doc1 with doc2, doc1 with doc3 and so on this will continue up to comparing file (n-1) with file (n) using the proposed formula. Then the cluster is formed by using the similarity matrix and threshold values. The threshold value is minimum limit for the similarity between the documents. When the similarity value is greater than the threshold then it is added to the cluster.

Fig-4: Sample XML documents

To do the experiment we used python language to provide the implementation and used python package element tree to read the XML file, xlsxwriter for storing the similarity matrix, lxml API for retrieving. Then retrieved X-Path from the xml file to compare a different documents.

The below table 2 consists of a structural similarity value of the above XML files.

Table 2: Similarity Matrix

Doc Name	Doc1	Doc2	Doc3	Doc4	Doc5	Doc6
Doc1	1	0.833	0.5	0	0	0
Doc2	0.833	1	0.5	0	0	0
Doc3	0.428	0.428	1	0	0	0
Doc4	0	0	0	1	0.714	0.142
Doc5	0	0	0	0.714	1	0.142
Doc6	0	0	0	0.2	0.2	1

Let consider the threshold value as 0.6(say). The cluster formed from the above similarity matrix are:

Cluster1: doc1, doc2
Cluster2: doc3
Cluster3: doc4, doc5
Cluster4: doc6

How the structural similarity value is calculated is Shown for the above sample XML documents.

$$StructSim_{12} = \frac{\text{X-Path doc(1)} \cap \text{X-Path doc(2)}}{\text{Total Number of X-Path in doc(1)}}$$

$$S_{12} = 5/6 = 0.833$$

Similarly calculate Structural Similarity for all other documents until all combinations of documents are completed

VI. CONCLUSION AND FUTURE ENHANCEMENTS:

In this approach, we have proposed an algorithm named as Hierarchical Structural Similarity (HSS) which uses a large number of XML documents as dataset. It uses X-Path of the nodes in a each XML file to compare the X-Path of the nodes in a different documents and forms a structure-based Hierarchical Document Clusters. It improves the efficiency of clustering, performance and scalability when it deals with large datasets used in Query Processing, Web mining, data transmission and Information retrieval. In the future process, in order to provide a the full valid cluster which considers both the structure and content similarity to cluster

efficiency the clustering is done using content-based XML document clusters.

REFERENCES

[1]. Akshay agrawal, Anand KR, Srivastava, "**Review on Clustering Techniques in XML Documents**" in International Journal of Advances in Electronics and Computer Science, 2014

[2]. Di Zhao, HaiDong Fu, Hui Ren, Mengxue Wei and Jie Chu, **"XML Documents Clustering Algorithm Based on Cluster Core And LSPX"**. in Institute of Electrical and Electronic Engineers, 2017**.**

[3]. Gianni Costa, Giuseppe Manco, Riccardo Ortale and Ettore Ritacco, **"Hierarchical clustering of XML documents focused on structural components"** in Elsevier Publication, 2012.

[4]. Karima Bessine, Hadda Cherroun and Attia Nehar," **XCLSC: Structure and Content-based clustering of XML" documents"** in Institute of Electrical and Electronic Engineers, 2015.

[5]. Maciej Piernik, Dariusz Brzezinski, Tadeusz Morzy, **"Clustering XML documents by Patterns".** in Association of Computing Machinery , 2016.

[6]. Mehdi G.Duaimi and YasirAbdAlhamed, **"Mining XML Document Based on Structure"** in International Journal of Advanced Research in Computer Science, 2013.

[7]. Mary Posonia A and Dr V Ljyothi **"Structural- based Clustering Technique 0f XML Documents"** in Institute of Electrical and Electronic Engineers, 2013.

[8]. Rehab Shalabi and Ahmed Elfatatry **"Towards improving XML search by using structure clustering technique"** in Journal of Information Science, 2014.

[9]. Suma D, U. Dinesh Acharya and GeethaM, Raviraja Holla M, "**XML Information Retrieval:An overview**" in International Global Journal For Engineering Research– Volume 10 Issue 1 –2014

BIOGRAPHIES

S.SAHUNTHALA, M.E,
pursuing Ph.D in Data Mining.
ASSISTANT PROFESSOR,
INFORMATION TECHNOLOGY,
ANAND INSTITUTE OF HIGHER TECHNOLOGY

NANDHINI S,
Student, Information Technology,
ANAND INSTITUTE OF HIGHER TECHNOLOGY

POOJASREE K,
Student, Information Technology,
ANAND INSTITUTE OF HIGHER TECHNOLOGY