Volume No 03, Special Issue No. 01, March 2015 ISSN (online): 2348 – 7550

# A SUPERVISED LEARNING METHOD TO CLUSTER XML DOCUMENTS WITH REDUCED COMPLEXITY

**Aditya Kumar Mishra[1], K. Vinay Kumar[2]**

[1,2] *Department of Computer Science And Engineering, National Institute of Technology, Karnataka,(India)*

**ABSTRACT**

*This paper presents an efficient methodology for clustering of XML documents. Because of enormous amount of XML documents available, it is necessary to find out the method for clustering [Antoine Doucet, 2002] of XML documents. This method groups the XML documents on the basis of the structure and content of XML documents. For Homogeneous XML documents [Nierman, 2002] and Heterogeneous XML documents [Graupmann, 2005], this method to be proved as efficient.*

*Since there are lots of DTD (Document Type Definition) available for XML documents. So those documents which are of same DTD are called as Homogeneous XML documents and which are of different DTD are called as Heterogeneous XML documents. This method considers both structure and content of documents and find out the cluster to which the document belongs to. Structure similarity is calculated by mining of XML* [Denoyer, 2007*] tags in the document and content similarity is calculated by using mining of data included in the document.*

***Keywords: Clustering, Content, Heterogeneous XML documents, Homogeneous XML documents, Similarity, Structure.***

## I. I NTRODUCTION

Since the web is the effective source of exchanging of information. These information is exponentially increasing day by day. The form in which it exists that is extensible Markup Language (XML) documents. Because XML documents describe itself and also it is very flexible as it can consist of user defined tags. Web includes Yahoo, Google, EBay, Wikipedia, social networking, government departments and many more. So the web contains a large amount of Heterogeneous XML documents .

Several tools and algorithms developed for storing, mining, retrieving and delivering XML data. However, they also require the efficient method to manage the XML data in such a way that storing of XML documents, retrieving content from XML documents, and delivering XML data.

XML documents are in the form of tree, i.e. they have both starting and ending tags, tag with their attributes, elements and contents. The tag defines the structural property of documents and content defines the content of XML documents. Tags, Attributes and Elements represent the internal nodes of the tree and the content represents the leaves of the tree. So to find out the structural similarity of documents, it is required to consider internal nodes only and for content similarity it is required to consider the leaves also. After finding out the

similarity between the documents, it is required to cluster these documents according to some algorithms so that the result for the query gives satisfaction to the user.

In the recent years, there are many algorithms developed to cluster these XML documents. But the problem arises in heterogeneous documents clustering. It also increases the complexity as moves from homogeneous documents to heterogeneous documents.

In the recent years, there are many algorithms developed to cluster these XML documents. But the problem arises in heterogeneous documents clustering. It also increases the complexity as moves from homogeneous documents to heterogeneous documents.

For heterogeneous documents, the tree structure is different, but the content may be same or different for example user queries for the documents i.e. either text or images or videos. So the aim is to search all the useful links or documents regarding the query no matter what is the structure. Therefore, it is a big problem where it is required to search for the content and structure both.

In homogeneous XML documents only the contents is our concern, so the clusters mainly depends on the content of XML documents.

Clustering of XML documents are different from text data and also complicated. Several approaches have been developed for clustering of XML documents based on content or structure or both, but there are still some problems in the terms of complexity and efficiency. Since, there are many challenges in finding out the similarity in Heterogeneous XML documents.

This paper proposes a different strategy to cluster Homogeneous and Heterogeneous XML documents depends on both structure and content. First, the XML documents converted into tree form in which tags are nodes of the tree. For structure similarity, the editing in trees is calculated with the least number of operations and for content similarity, the similarity factor calculated using mathematical formula which will be explained later in paper. In this way, the documents are grouped.

This methodology describes the unsupervised learning of system, in which the system first trained with large number of documents to form clusters and then other documents are tested using this system which gives an efficient way of cluster.

This paper contains the following section. The next section discusses about XML documents with example and also differentiates between Homogeneous and Heterogeneous XML documents. Section 3 consists of the literature survey. Section 4 describes the algorithm for clustering of XML documents. Section 5 consists of System Implementation which describes development of system. Section 6 consists of Mathematical evaluation of algorithm developed. Section 7 consists of results and last section describes the conclusion.

## II. HOMOGENEOUS AND HETEROGENEOUS XML DOCUMENTS

XML emerged in 1997 by W3C and it proved to be a powerful language for modeling the information of web. Conversion of web data in the form of XML documents reduces the complexity for different types of applications and after that, a large amount of data are being converted into XML documents in many areas of programming and internet.

Volume No 03, Special Issue No. 01, March 2015 ISSN (online): 2348 – 7550

```
<?xml version='1.0' ?>
<!DOCTYPE root SYSTEM "http://www.cs.washington.edu/research/projects">
<root>
                    <listing>
   <seller_info>
       <seller_name>537_sb_3 </seller_name>
       <seller_rating> 0</seller_rating>

   <payment_types>Visa, Mastercard
   </payment_types>
   <shipping_info>siteonly, Buyer Pays Shipping Costs
   </shipping_info>
   <buyer_protection_info>
   </buyer_protection_info>
   </seller_info>
   </listing>
</root>
```

**Fig. 1: fig1.xml**

XML schema describes the XML documents consists of constraint in structure and content both. These schemas define document type descriptor (DTD) of XML documents. XML documents are of different DTD's. It consists of different tags and each tag has its end tag also. These tags are considered as elements. XML documents of same DTD are called as Homogeneous XML documents. XML documents of different DTD are called Heterogeneous XML documents.

```
<?xml version='1.0' ?>
<!DOCTYPE root SYSTEM "http://www.cs.washington.edu/research/projects">
<root>
                    <listing>
   <seller_info>
       <seller_name> cubsfantony</seller_name>
       <seller_rating> 848</seller_rating>

   <payment_types>Visa/MasterCard
   </payment_types>
   <shipping_info>Buyer pays fixed shipping
   </shipping_info>
   <buyer_protection_info>
   </buyer_protection_info>
   </seller_info>
   </listing>
</root>
```

**Fig. 2: fig2.xml**

```
<?xml version="1.0" ?>
<!DOCTYPE root SYSTEM "http://www.cs.washington.edu/research/projects">
<root>
<course>
   <reg_num>10577</reg_num>
   <subj>ANTH</subj>
   <crse>211</crse>
   <sect>F01</sect>
   <title>Introduction to Anthropology</title>
   <units>1.0</units>
   <instructor>Brightman</instructor>
   <days>M-W</days>
   <time>
       <start_time>03:10PM</start_time>
       <end_time>04:30</end_time>
   </time>
   <place>
       <building>ELIOT</building>
       <room>414</room>
   </place>
</course>
</root>
```

**Fig. 3: fig3.xml**

fig1.xml and fig2.xml are considered as Homogeneous XML documents and fig1.xml, fig3.xml and fig2.xml, fig3.xml are considered as Heterogeneous XML documents. fig1.xml and fig2.xml are of same DTD and consist of same tag and elements but content are different whereas fig1.xml and fig3.xml are of different tags, elements and content.

Since, XML documents consist of opening and closing tags. So it can be denoted in the form of tree where the nodes represent the tags of XML documents.

## III. LITERATURE SURVEY

[Kaizhong Zhang and Dennis Shasha, 1989] have proposed the algorithms for editing distance between trees and related problems with the quadratic complexity. In this, one tree is compared with other tree and finding the number of operations required to make both trees identical. The operations are insertion, deletion and modification. If one tree has not one node then it is to be inserted in the tree. If one tree has one node additional at some level then it is to be deleted from the tree and if one tree has same number of nodes on a level then it can be modified. So in this way numbers of operations are calculated. Dynamic programming is used to find out the distance between varieties of trees with same complexit

[Yoon, Jong P., Vijay Raghavan, and Venu Chakilam, 2001] have proposed the BitCube – a three dimensional bitmap indexing form XML documents in which matrix construction time is noticeable. According to this method, the documents are in the form of 2-D bitmap. Then finding out the similarity using variance and mean in bitmap of documents and by using the similarity factors, the documents are being partitioned into clusters. Bit operation takes less time but matrix construction time for large documents is very high.

[Sergio Flesca, Giuseppe Manco, 2002] have proposed Fourier transformation techniques which use the time series representation. Different number of occurrences of an element or small shifts in its position has no effect on similarity estimation. The structure of documents (XML) converted to time series representation in which time noted according to the occurrences of tags and their appearances order in the document. The tags are

extracted and converted into the sequence of time frames. Then by applying the Discrete Fourier Transformation (DFT), they can have the sequence of frequencies. Encoding schemes are being used to find out the signal samples for the documents. So this method is as effective as tree edit distance.

[Theodore Dalamagas, 2006] have proposed hierarchical algorithm for clustering documents which has quadratic complexity for single link and product of quadratic and logarithmic for complete link. The methodology is used to cluster XML documents by structure for similar documents by structure and also improve the performance of the edit distances.

[Jianwu Yang and Songlin Wang, 2010] have proposed partitioned clustering algorithm for clustering with linear time complexity but the problem arises in sensitivity to initial center points and the assumption of knowing number of clusters. Authors used k-means algorithm to find out clustering of XML documents based on text but improving the quality by using the tags. For finding similarity, vector space model is used and k-means for clustering.

[Bin Zhao, 2008] have proposed combination algorithms for clustering which make them robust than single partitioned algorithms. Authors combined the single link partition algorithm and hierarchical clustering algorithm and eliminate the problems in single link clustering algorithm. XML documents are represented in the form of n-D vector.

[Elaheh Asghari, 2013] have proposed the multilevel clustering algorithm by applying different degrees of importance on different levels of elements in the tree make them efficient methods for clustering XML documents. Author proposed the algorithm to apply different clustering algorithm on different levels so it is very difficult to determine which algorithm to apply on which level. So the time complexity and the quality of cluster depend on level selected and algorithm selected.

## IV. ALGORITHM

The algorithm developed has four steps to cluster the XML documents. These four steps are Document Representation, Structure Similarity, Content Similarity, and finally to find out the cluster. In the first step, representation of document is described. The second step describes the method to find out the similarity between the structure and the third step describes the method to find out the similarity between the content of documents. Finally the last step is used to find out the cluster of the documents to which it belongs.

### 4.1 Document Representation

XML documents are represented in the form of tree (bracket notation) in which tags represent nodes of trees. Structure of document is stored in the file of .tree extension and content is stored in text file. Since the XML documents must have the end tag for each and every start tag so it can be transformed in the form of tree. For example fig1.xml can be written in fig1.tree, fig2.xml represented in fig2.tree and fig3.xml represented in fig3.tree. So in this way the XML document are being represented as in the form of tree. Nodes in fig1.tree and

fig2.tree are similar so called as homogeneous XML documents whereas fig3.tree and fig1.tree or fig3.tree and fig2.tree are heterogeneous XML documents.

```
{root
 {listing
  {seller_info
   {seller_name}
   {seller_rating}
   {payment_types}
   {shipping_info}
   {buyer_protection_info}
  }
 }
}
```

**Fig. 4: fig1.tree**

```
{root
 {listing
  {seller_info
   {seller_name}
   {seller_rating}
   {payment_types}
   {shipping_info}
   {buyer_protection_info}
  }
 }
}
```

**Fig. 5: fig2.tree**

```
{root
 {course
   {reg_num}
   {subj}
   {crse}
   {sect}
   {title}
   {units}
   {instructor}
   {days}
   {time
    {start_time}
    {end_time}
   }
   {place
    {building}
    {room}
   }
  }
 }
```

**Fig. 6: fig3.tree**

The tree format is extracted by using stack in which for every start tag, one '{' and the tag is pushed in the stack and for every end tag, one '}' pushed into stack. Finally in this way the tree form of XML document extracted.

### 4.2 Structure Similarity

The structure similarity between two trees t1 and t2 depends on difference of nodes in trees. String comparison operation is required in comparison of two nodes. If it is similar then no need to increase the value of count but if it is different then increase the count by 1. So in this way, finally get the result of number of different nodes in two trees.

Now the number of operation for every tree pair can be calculated but the reciprocal of this value is to be considered as ES[ij](equality in structure of tree i and tree j).

]If the number of operations required is zero then the reciprocal considered as 1 i.e, the most similar tree.

### 4.3 Content Similarity

Content similarity means similarity on the basis of content. Content is the text between the nodes of the tree. So for each and every tree there is a document file which consists of the content of XML document. To find the similarity of content between tree pairs, following steps to follow.

Collection of documents are pre-processed and represented as term document matrix.

Entry in matrix corresponds to the weight of a term in a document.

Zero means term has no significance in document.

f[ij]: frequency of term i in document j.

term frequency(tf[ij])= f[ij]/max(f[ij]) where max(f[ij]) is frequency of most common word in document.

Terms that appears in many different documents.

df[i]= document frequency of term i i.e number of documents containing term i.

idf[i]= inverse document frequency of term i

= lg(N/df[i]) where N is total number of documents.

Log used to dampen the effect relative to term frequency.

w[ij]= tf[ij]*idf[i] where w[ij] is the weight of term i in document j.

EC[i,j] is the equality in content between two documents which is calculated as summation of the product of weight of common words in both trees and normalized using square root of the square of common words.

### 4.4 Clustering

Clustering means to collect the objects which are similar between them belongs to one group and dissimilar objects belong to other group. Cluster for homogeneous XML document, based on equality in structure whereas for heterogeneous XML documents based on equality in content. There are number of groups selected and for every group there are some documents belong to those groups which can best describe the group and also the corpus developed for every group which consists of the content of all documents which are available in the group. Now for every group, one document selected as the center for that group which can be used to calculate the ES for the document and the corpus used to calculate EC. In this way, determination of cluster for XML documents accomplished.

## V. SYSTEM IMPLEMENTATION

The system developed for implementing this algorithm, uses the supervised learning in which training document defined for every group and based on these documents, other documents are tested to find out their groups. There are five groups which are Books Shopping, Music, Astrology, Colleges and Science. Each group consists of large number of documents which are related to their groups. In every group, there is a central document selected which has highest ES with maximum number of documents in the group. The content of all documents in the group stored in a document file and then the file is pre-processed. There is a threshold value to be decided for making the decision of the group for XML document, if ES smaller than that value then the document can not belong to that group and if Es is greater than that value then the document can belong to that group. So now,

the system is ready for testing. The group of XML document determined on the basis ES and EC. The ES calculated for an XML document with central document of every group and the document belongs to the group with which it has highest value of ES but the value must be greater than threshold value. But if the ES value with every group is smaller than threshold value then calculate the value of EC with the document file of every group. There is other threshold value decided for content so if the value of EC is smaller than that value, document can not belong to that group but if value is higher than document may belong to that group. Now the document belong to the group with which it has the highest value and greater than the threshold value. If document has both ES value and EC value less than the threshold value decided with each and every group then it does not belong to any of the group and it is placed in additional group.

## VI. EVALUATION

The analysis of complexity of clustering of homogeneous and heterogeneous XML documents in the following manner.

Let the center of group consists of n nodes and XML document whose group is to be decided has m nodes. So n[1] is number of nodes for group 1, n[2] for group 2 etc. Let the number of groups is k. Let the number of words in document file of group has w words and document file for new XML document has x words. So w[1], w[2],...,w[k] is number of words in document file for group 1, 2, ..., k.

Time required for extracting tags and extracting content to respective file is very less as compared to other factors if the speed of traversing a file is very less as compared to other factors. So it can be ignored.

Now complexity for clustering of homogeneous XML document is max(m,n) where max indicates maximum and it is required for every group. max(m,n[1])+ max(m,n[2])+...+ max(m,n[k]) is the total complexity. Since the number of groups are very less as compared to number of documents and number of nodes. So it is very less than quadratic.

Now complexity for clustering of heterogeneous XML documents is (max(m,n[1])+ max(m,n[2])+...+ max(m,n[k]) + x*log(w[1]) + x*log(w[2]) + ... + x*log(w[k]))where log(w[1]) is the searching time of one word in file of group 1 by using universal hashing data structure. Since the number of groups is very less as compared to number of nodes and numbers of words in document file so the complexity is less than product of quadratic and logarithmic function which is previously defined in literature survey. In this way the complexity for clustering for both homogeneous and heterogeneous XML document reduces by reasonable amount.

Now these comparison of new documents with central documents can be done in parallel way then max(m,n[1])+ max(m,n[2])+...+ max(m,n[k]) can approach to the max(m,N) where N is the number of nodes which is maximum among all the central documents and x*log(w[1]) + x*log(w[2]) + ... + x*log(w[k]) can approach to x*log(W) where W is the number of words in largest document file of group among all document file of groups . Finally the complexity for clustering of homogeneous XML documents reduces to max(m,N) and for heterogeneous XML document reduces to max(m,N)+x*log(W).

## VII. RESULTS

In the above section, we have described about the complexity to find out the cluster for an XML document by using supervised way. We have seen that the complexity for clustering of homogeneous XML document and heterogeneous XML document reduced. Table 1 indicates the results shown for homogeneous XML documents. Let the time unit(tu) for one insertion or modification or comparison in a tree is equal to 1 tu. This is indicated in time unit as because this value depends on the processor speed and memory size.

**Table1**

| Category | Central Doc. Nodes | Tested Doc. Nodes | Time(in tu) |
|---|---|---|---|
| Book Shopping | 145 | 150 | 209 |
| Music | 175 | 163 | 209 |
| Astrology | 134 | 135 | 209 |
| Colleges | 209 | 207 | 209 |
| Science | 178 | 180 | 209 |

Table 2 shows the results for Heterogeneous XML documents. CentralNodes represent the number of nodes in central document, TestedNodes represent the number of nodes in tested document, Category (words) and Tested (words) represent the number of words in category or group document and tested document respectively.

**Table2**

| Category | CentralNodes | TestedNodes | Category(words) | Tested(words | Time(tu) |
|---|---|---|---|---|---|
| Book Shopping | 145 | 175 | 510 | 475 | 1622.5 |
| Music | 175 | 135 | 310 | 305 | 1116.6 |
| Astrology | 134 | 125 | 754 | 510 | 1905.3 |
| Colleges | 209 | 150 | 524 | 564 | 1887.4 |
| Science | 178 | 135 | 94 | 839 | 2705.8 |

The results are tested for about 500 homogeneous XML documents and 400 Heterogeneous XML documents. Trained documents are 100 for every group out of which one is selected as central document and for every group there is one document file maintained.

## VIII. CONCLUSION

We have defined the methodology for clustering of homogeneous XML documents and heterogeneous XML documents with reduced complexity by considering both structure and content of XML documents. For structure, we have converted every XML document in the form of tree and the compared two XML document to find structure similarity and for content similarity, we have formed the document file for every document which consists of its content then find out the similarity by using term frequency and document frequency. So this algorithm gives the result according to the reduced complexity.

## IX. ACKNOWLEDGEMENT

We would like to thank to the Department of Computer Science and Engineering of National Institute Of Technology, Karnataka, Surathkal for providing their best resources for developing and implementing this system.

## REFERENCES

[1] Antoine Doucet, Helena Ahonen-Myka, et al. Naive clustering of a large xml document collection. 2002:81{87, 2002}.

[2] Nierman, Andrew, and H. V. Jagadish. "Evaluating Structural Similarity in XML Documents." WebDB. Vol. 2. 2002.

[3] Graupmann, Jens, Ralf Schenkel, and Gerhard Weikum. "The spheresearch engine for unified ranked retrieval of heterogeneous XML and web documents."Proceedings of the 31st international conference on Very large data bases. VLDB Endowment, 2005.

[4] Denoyer, Ludovic, and Patrick Gallinari. "Report on the xml mining track at inex 2005 and inex 2006: categorization and clustering of xml documents." ACM SIGIR Forum. Vol. 41. No. 1. ACM, 2007.

[5] Kaizhong Zhang and Dennis Shasha. Simple fast algorithms for the editing distance between trees and related problems. SIAM journal on computing, 18(6):1245{1262, 1989.

[6] Yoon, Jong P., Vijay Raghavan, and Venu Chakilam. "BitCube: A three-dimensional bitmap indexing for XML documents." Scientific and Statistical Database Management, 2001. SSDBM 2001. Proceedings. Thirteenth International Conference on. IEEE, 2001.

[7] Sergio Flesca, Giuseppe Manco, Elio Masciari, Luigi Pontieri, and Andrea Pugliese. De-tecting structural similarities between xml documents. 2:55{60, 2002.

[8] Theodore Dalamagas, Tao Cheng, Klaas-Jan Winkel, and Timos Sellis. A methodology for clustering xml documents by structure. Information Systems, 31(3):187{228, 2006.

[9] Jianwu Yang and Songlin Wang. Extended vsm for xml document classification using frequent subtrees. pages 441{448, 2010.

[10] Bin Zhao, Yong-Sheng Zhang, and Hua-Xiang Zhang. A robust clustering method for xml documents. 1:19{23, 2008.

[11] Elaheh Asghari and MohammadReza KeyvanPour. Xml document clustering: techniques and challenges. Artificial Intelligence Review, pages 1{20, 2013.