**XCLS++: A new algorithm to improve XCLS+ for clustering XML documents**

Ahmad khodayar\(^1\) and Hassan naderi\(^2\)

\(^1\)Department of Computer Engineering, Shabestar Islamic Azad University of Science and Technology, Shabestar, Iran  
\texttt{ahmad_kho2000@yahoo.com}

\(^2\)Department of Computer Engineering, Iran University of Science and Technology, Resalat st., Tehran, Iran  
\texttt{naderi@iust.ac.ir}

**Abstract**

The purpose of this paper is to offer a method for clustering of XML documents. Many different ways of clustering were discussed for clustering documents which can be divided to structure, content and combination of structure and content. One method that has been proposed to solve this problem is XCLS which be improved with XCLS+ later. XCLS+ is efficient algorithm for clustering XML documents and Exposure into structure based on clustering category. The conducted survey showed which XCLS+ method has problem that makes away from its optimal value too. This paper presents a method with name XCLS++ which has not related problem XCLS+ and its efficiency are diagnosed more than XCLS+.

**Keywords**

Clustering, XML documents, XCLS+ algorithm, Content similarity of document, Structure similarity of document

1. Introduction

The XML documents are currently devoted for exchanging the largest volume of textual information on the internet. Interesting characteristics of XML, including ability to describe the information (which makes information store in any computer), be its textual (been processed with any operating system and software) and have been understood with machines and humans, have caused particular popularity of this format in the IT community. On the other hand, because this technology is relatively new, high-performance systems that enable to process it, are still being developed. XML documents easy management and access to content with giving structure to content. So because XML document is composed of content and structure therefore in processing of XML documents should are considered these two parts.

With increased spread of information, for its storage, retrieval and transmission through internet, we need to manage that information basically. Clustering is very useful in above processes and for summarizing and reducing size of documents. Clustering XML documents also not an exception and as many applications are faced with problem of clustering. For example, with representative of each cluster obtained with an efficient clustering algorithm can very large volume set of documents to store more briefly. Formula for calculating is the first appliances required for clustering set of documents.
Methods to determine similarity two XML documents are generally divided into three categories: content [10], structure [2] [5] [11] and combination of mentioned methods [1] [3] [4] [8] [9]. Clustering XML documents base on content is very similar to clustering textual documents which a lot of works have been performed in this field. On the other words in this kind of clustering, document tags omitted or considered as content, and clustered with planned algorithms easily. In this way, criterion for clustering is content and if documents have content similarity then has been laid in same cluster. Content based on clustering used for searching textual documents with Google and AltaVista engines. Clustering base on structure tries to find documents which have structure similarity and puts them in similar cluster. Combined clustering uses advantages two types of content and structural clustering.

One of the methods of structural clustering XML document that has good efficiency is XCLS [5]. Performance of this method is to find similar tag name of nodes in levels of tree correspondence to document and then calculating similar number. But fundamental problem XCLS is neglect relationship FATHER-NODE that makes order of nodes is not preserved. XCLS+ method are created for improving and solving the problem mentioned XCLS method [11]. XCLS+ method has tried by adding a factor as relationship FATHER-NODE in XCLS related formula and solved the problem and improved previous method. But with careful study have been observed which the XCLS+ method because not considering duplicate nodes has problem and is way from optimality. Therefore this article tries to improve similar formula calculation XCLS+ and a new method called XCLS++ is established. In section2 tree model in XML documents will be reviewed. Section3 discusses previous works and then in section4 detail analysis of the XCLS+ method will be expressed. In section5 the relative bug of XCLS+ method will be offered. The proposed method will be offered in section6 for solving the bug XCLS+ method. Then in next section, results of proposed method are presented and compared with the XCLS+ method and will be fond that optimality of the proposed method is good in comparison with XCLS+ method. In final section summary and conclusions will be brought.

2. Tree model of XML documents

As the XML documents are the tree format, so documents can to model in tree case. In this case, problem of clustering the XML documents will decrease into clustering trees. The structural methods benefit of these trees. The XCLS+ method that uses of tree model is an example which is successful in clustering the XML documents. The XCLS+ method optimized the XCLS method. Despite improvements with XCLS+ on XCLS, but XCLS+ has bug too. The main reason this method which is not optimal (from now on we talk only about XCLS+), neglecting duplicate nodes in the tree levels which in this paper tried to fix it.

3. Previous works

Clustering criteria is to find similar documents. Similarity of documents should be found with a certain similarity and clustering is done base on similarity. As mentioned above three methods for finding similarities has been suggested and are: 1- content [10] 2- structure [2] [5] [11] 3- content with structure [1] [3] [4] [8] [9]. As we know, each XML document can be changed to tree and clustering operations can be done with these trees. The structural methods considered only the tree structure and do not work with content. XCLS+ is the structural approach which in this article has focused on this type clustering. As previous mentioned, this method have been created to optimize the XCLS method. Experiments view that the XCLS+ method despite good performance compared to the XCLS method has a basic problem that decrease efficiencies in some circumstances. with more studies was concluded that main reason for low efficiency of this method is to neglect duplicate similar nodes in trees which in this paper tried to present solution
could solve bug and improve the XCLS+ algorithm. More details in this article will be brought in the next. In this paper, first XCLS+ have been studied then proposal algorithm offered. Finally the results compared with proposed method and have been seen which optimality of the proposed method is better than the XCLS+ method.

4. XCLS+ method

This method uses a similar structure and acts in more detail base on tag similarity between two levels of tree nodes related to document tags. In this way the incoming XML document compared with clustered documents and if there is acceptable similar number between those, new XML document is placed beside relevant clustered document. Acceptable new incoming document combined with document in cluster and form a new tree. In other words, in each cluster there is a single tree. If similarity value is not acceptable in this case a new cluster has created and incoming document is placed in it. This practice will continue until entrance final document. The similarity calculation is done based on Formula1. This formula is created for improving the XCLS method. In this way operations taken when document arrival and how execute it include: first root node of incoming document has compared with root node of clustered documents then if there are similar, factors the Formula1 is calculated and comparing levels nodes in each document have been continued in a level down for similarity calculation. In the absence of similarity, the next comparison is done between node in the clustered document in a level down and the same node of the incoming document. This compares has been continued to the leaves levels of document. As previously noted, similarity value have been calculated with:

\[
\text{sim base on XCLS+} = \frac{0.5 \times \sum_{i=0}^{l-1} (CN^i + CP^i) \times r^{l-i-1} + 0.5 \times \sum_{j=0}^{l-1} (CN^j + CP^j) \times r^{l-j-1}}{(\sum_{k=0}^{l-1} N^k \times (r)^{l-k-1}) \times z + 0.5 \times (\sum_{i=0}^{l-1} CP^i \times r^{l-i-1} + \sum_{j=0}^{l-1} CP^j \times r^{l-j-1})}
\]

Formula1. The XCLS+ method based on formula

Numeric Value Formula1 is variable between zero and one. This value will change relatively on similar.

The variables in the above formula are:

1- Z cluster size or in other words, the number of documents within cluster.

2- CN^i is the total of similar nodes between level i of the new incoming document and level j of the clustered documents.

3- CN^j is the total of similar nodes between level j of the clustered documents and level i of the new incoming document.

4- CP^i is the total of similar nodes between level i of the new incoming document and level j of the clustered documents as have the same father.

5- CP^j is the total of similar nodes between in level j of the clustered documents and in level i of the new incoming document as have the same father.

6- N in N^k is the number of elements in level k of the incoming tree.

7- l is high of tree in the each document.
8- \( i, j \) are desired number of level.

9- \( r \) is the incremental factor, which is considered number 2.

10- \( k \) equal with 2.

The above formula is active in every jump between levels document and the similarity Value calculated is sum of obtained factors. This practice continues until Level one of documents ended. Overall algorithm the XCLS+ method is given in Section 4.1.

**4.1. XCLS+ method algorithm**

1- Start to look same in node two tree node. If a node is found then does the calculation of formula then move to step2, otherwise go to Step3.

2- If depth of trees are move toward the lower level in the both trees. Search the same node as step1. If there is same node, calculate the formula and repeat step2, otherwise go to step3.

3- If depth of tree is (usually in clustered document), move down level in the clustered document and stay in the same level of the new incoming document. Search again the same nodes. If there is the same node, calculate the formula And then repeat step2, otherwise repeat step3.

**4.2. Example of XCLS+ method**

For understanding the algorithm an example is given in this section. Goal is to find similarity between tree1 and tree2 base on the XCLS+ method. It should be noted that the tree1 referred to the incoming document and the tree2 referred to the clustered documents. Arrows of up to down are indicative the order of execution of algorithm. In this example in the Figure1 for facility the variable values are calculated and placed on cut arrows

![Diagram of tree1 and tree2](image)

**Figure1. An example for showing how operation of the XCLS+ method**

After obtaining above factors, the similarity value base on the XCLS+method is:

\[
\text{sim base on XCLS}^+ = 0.5 \times \frac{0.5 \times \left( (0+0) \times 2^2 + (1+0) \times 2^2 + (2+2) \times 2^0 \right) + 0.5 \times \left( (0+0) \times 2^2 + (1+0) \times 2^1 + (2+2) \times 2^0 \right)}{(1 \times 2^2 + 3 \times 2^0) + 0.5 \times (0 \times 2^2 + 0 \times 2^4 + 2 \times 2^0)} = 0.85
\]
5. Problem for XCLS+ method

With more detail study of algorithms and testing various examples observed that the XCLS+ formula has problem because neglecting repeated nodes. For more clarify presented an example in the Figure2, 3.

After calculating above factors similar number is:

\[
\text{sim base on XCLS+} = \frac{0.5 \sum_{k=0}^{L-1} (CN^1 + CP^1)^{2^{L-k}-1} + 0.5 \sum_{k=0}^{L-1} (CN^1 + CP^1)^{2^{L-k}-1}}{\sum_{k=0}^{L-1} N^k \times 2^{N-k} + 0.5 \sum_{k=0}^{L-1} CP^1 \times r^{L-k} + 0.5 \sum_{k=0}^{L-1} CP^1 \times r^{L-k}} = 0.85
\]

For another example in the Figure3 similar number calculated based on the XCLS+ method is:

\[
\text{sim base on XCLS+} = \frac{0.5 \sum_{k=0}^{L-1} (CN^1 + CP^1)^{2^{L-k}-1} + 0.5 \sum_{k=0}^{L-1} (CN^1 + CP^1)^{2^{L-k}-1}}{\sum_{k=0}^{L-1} N^k \times 2^{N-k} + 0.5 \sum_{k=0}^{L-1} CP^1 \times r^{L-k} + 0.5 \sum_{k=0}^{L-1} CP^1 \times r^{L-k}} = 0.85
\]

As a result calculated for the Figure2 also for trees in the Figure3 similar numbers are equal and how calculation similar number for trees in the Figure3 is same similar number for trees in the Figure2. Factors Figure3 for calculating similar number on arrows in Figure3 written. As see all factors are same with factors in Figure2. Despite significant difference between In the Figure3, 2 have been seen which variable values are equal. So calculated similarity numbers with the XCLS+ method is equal in Figure2, 3. By comparing two above examples, have been concluded that similar numbers of first instance should be greater than second example. The results have obtained with XCLS+ are away from logic and should be different. The primary cause of this problem is ignoring repeated nodes in original formula with the XCLS+ method. Considering cp factor regardless of repeated nodes causes order of modified nodes in levels changed and for different trees, same similarity number be earned. So should method have been proposed which repeated cases are also included. This paper proposed a method called XCLS++ which tries with adding a new factor into the formula1 and solving problem the XCLS+ method in finding similar number be more effective. In continuation proposed method is mentioned in this article.
Figure 3. Same nodes of two trees are not equal in hierarchies

6. XCLS++: proposed method

As mentioned above the XCLS+ method has fundamental problem in calculating similarity number. This method calculates equal similarity number for trees in the Figure 2, 3. So should a method have been proposed as their calculated similarity numbers be different. In this article tried a method proposed for solving mentioned problem and gaining optimized results compared with the XCLS+ method. In this article tried the problem will solve with two steps. Reason of unreal calculation for same trees in Figure 2, 3 is duplicate nodes. This reason cases knocked hierarchy of tree nodes. Therefore must factor added to formula which preserves order of nodes. In this paper, in first step an incremental change are added to the original formula and then in second step replacement change are apply. Finally the proposed formula will be optimized formula in this paper. As will see final formula obtained is more reasonable and more realistic and have good optimality in compare with the XCLS+ method. Steps are mentioned bring in continue.

6-1. step1

In step1 in New mentioned method for solving problem factor FATHER_CHILD_NODE added in the formula1. This factor in proposed method formula is cc and proposed formula mentioned in formula2 is:

\[
\text{sim base on XCLS + +} = 0.5 * \sum_{i=0}^{I-1} (CN^1_i + CP^1_i + CC^1_i) \cdot r^{l-i-1} + 0.5 * \sum_{i=0}^{I-1} (CN^I_i + CP^I_i + CC^I_i) \cdot r^{l-i-1} / (\sum_{k=0}^{I-1} N^k \cdot (r)^{l-k-1}) \cdot z + 0.5 * (\sum_{i=0}^{I-1} (CP^1_i + CC^1_i) \cdot r^{l-i-1} + \sum_{j=0}^{I-1} (CP^I_i + CC^I_i) \cdot r^{l-i-1})
\]

formula2. Step1 stage formula of XCLS++ proposed method

In the above formula, all variables equal with variables in Formula1 and only cc is new factor. New Factor represents number of same nodes that have same father and same children. In neglecting this factor in ways that is XCLS+ causes to grow tree toward duplicate child node and create unreasonable results. Considering above mentioned factor causes problem solved. Because in this case growing tree of repeated node was determined and proportionate value of cc is placed. For proving base on optimality of proposed method, similarity numbers for examples in Figure 2, 3 obtained again with proposed formula provided in formula2 and shown that proposed method for two above mentioned examples which have significant difference, give better results than the XCLS+ method. The similarity of trees Figure 2 and Figure 3 is calculated again.
sequentially in Figures 4, 5. Result are seen after related Figures. Note which factors without () belong for both trees.

Figure 4. Calculating similarity number for trees based on the step 1 optimization

\[ \text{sim base on XCLS} + = \frac{0.5 \times \sum_{i=1}^{n} (CN_1 + CP_1 + CC_1) \times 2^{l-i-1} + 0.5 \times \sum_{i=1}^{n} (CN_2 + CP_2 + CC_2) \times 2^{l-i-1}}{2^{n+2+k-1} + 0.5 \times \sum_{i=1}^{n} (CP_1 + CC_1) \times 2^{l-i-1} + 0.5 \times \sum_{i=1}^{n} (CP_1 + CC_1) \times 2^{l-i-1}} = 0.85 \]

Figure 5. Calculating similarity number for trees based on the step 1 optimization

\[ \text{sim base on XCLS} + = \frac{0.5 \times \sum_{i=1}^{n} (CN_1 + CP_1 + CC_1) \times 2^{l-i-1} + 0.5 \times \sum_{i=1}^{n} (CN_2 + CP_2 + CC_2) \times 2^{l-i-1}}{2^{n+2+k-1} + 0.5 \times \sum_{i=1}^{n} (CP_1 + CC_1) \times 2^{l-i-1} + 0.5 \times \sum_{i=1}^{n} (CP_1 + CC_1) \times 2^{l-i-1}} = 0.83 \]

With comparing the two observed numbers contrary the XCLS+ method, the proposed method able to distinct difference between above mentioned cases. So factor cc able to solve problem XCLS+. But with next examples have been seen which this factor has problem too and were not solved problem completely. In continue mentioned examples will be brought.
6-2. problem for step1

Also step1 able to solve problem of xcls+ but for some examples are problem too. For example for trees in figure6, 7 similarity numbers calculated with step1 are 0.96, 0.95 sequentially. Result are seen after related Figures.

Figure6. calculating similarity number for trees base on step1 optimization

\[
sim \text{ base on XCLS } = \frac{0.5 \times 2^{-1} \left( C_{11} + C_{10} + C_{01} \right) + 2^{-L-1} + 0.5 \times 2^{-L} \left( C_{11} + C_{10} + C_{01} \right) + 2^{-L-1}}{\sum_{k=0}^{L} 2^{-k} + 0.5 \times \left( \sum_{l=0}^{L-1} \left( C_{11} + C_{10} + C_{01} \right) + 2^{-l-1} + \sum_{k=0}^{L-1} \left( C_{11} + C_{10} + C_{01} \right) + 2^{-k-1} \right)} = 0.96
\]

Figure7. calculating similarity number for others trees base on the step1 optimization

\[
sim \text{ base on XCLS } = \frac{0.5 \times 2^{-1} \left( C_{11} + C_{10} + C_{01} \right) + 2^{-L-1} + 0.5 \times 2^{-L} \left( C_{11} + C_{10} + C_{01} \right) + 2^{-L-1}}{\sum_{k=0}^{L} 2^{-k} + 0.5 \times \left( \sum_{l=0}^{L-1} \left( C_{11} + C_{10} + C_{01} \right) + 2^{-l-1} + \sum_{k=0}^{L-1} \left( C_{11} + C_{10} + C_{01} \right) + 2^{-k-1} \right)} = 0.95
\]

As see two up examples show step1 only do not ables to solve xcls+ problem completely and for tow group’s trees calculates unreasonable similarity numbers. So must changed another factor in formula for earning optimal similarity numbers. This work will be brought in step2.
6-3. step2

For latest optimizing proposed method must a factor changed too. Factor which cases unreason result is same father. In this step rather than it, factor same brother replaced. Latest optimized formula is:

$$\text{sim base on XCLS + + } = \frac{0.5 \cdot \sum_{k=0}^{4-1} (CN^2_k + CB^1_k + CC^1_k) \cdot 2^{4-k-1} + 0.5 \cdot \sum_{i=0}^{4-1} (CN^1_i + CB^1_i + CC^1_i) \cdot 2^{4-i-1}}{\sum_{k=0}^{4-1} N^k \cdot 2^{4-k-1} + 0.5 \cdot \left( \sum_{i=0}^{4-1} (CB^1_i + CC^1_i) \cdot r^{4-i-1} + \sum_{i=0}^{4-1} (CN^1_i + CB^1_i + CC^1_i) \cdot r^{4-i-1} \right)}$$

formula3. Latest formula of XCLS++ proposed method

Now for trees in Figure 6, 7 similarity numbers again calculated with formula3 in Figures 8, 9. Results of calculated are after related Figures:

Figure 8. Calculating similarity number for Figure 6 trees base on step 2 optimization

Figure 9. Calculating similarity number for Figure 7 trees base on step 2 optimization
As see results calculated with latest optimized formula are reasonable. For clarifying effectiveness the above factors the XCLS++ and the XCLS+ algorithms implemented and have been seen that how much this factor plays a fundamental role in applied environment. This work will be taken in next sections.

7. Evaluating algorithms and comparing methods

As was shown in above examples the XCLS++ approach in comparison with the XCLS+ method is good. For proving above sentence in this section the XCLS+ and XCLS++ algorithms have been implemented. Both of them were implemented with C language in DOS environment on a machine with 2.4 GHZ Intel Celeron CPU and 512 MB of RAM. The evolution criteria were implemented too in same conditions for evaluating xml files in different types. The results of experiments like above examples, confirm optimality of the proposed algorithm and efficiency of the XCLS++ algorithm is higher than the XCLS+ method.

For evaluating similarity diagnostic algorithms and clustering XML documents, must first set of the incoming files have been specified and after clustering accuracy of algorithms will be calculated with existent criteria.

7.1. Set data

For evaluating, files divided in two categories for best analysis and both algorithms executed on same files in same conditions for efficiency comparing. Two mentioned categories are homogeneous files (from one type DTD) [6] and heterogeneous file (multi type of DTD) [7]. The results for both categories will be shown separately.

7.2. Evaluation criteria

There are three items for calculating accuracy clustering algorithms: 1-entropy 2-purity 3-fscore

7.2.1 Entropy

Entropy is sum documents which located in the cluster i which are of the class r. The entropy formula is:

\[ Entropy = \sum_{i=1}^{k} \frac{n_i}{N} E(C_i) \]

\[ as \]

\[ E(C_i) = \frac{1}{log_k} \sum_{\ell=1}^{k} \frac{n_{i}^\ell}{n_i} log \frac{n_{i}^\ell}{n_i} \]

In above formulas, \(C_i\), \(N\), \(k\), \(n_i\) and \(n_{i}^\ell\) are respectively ith cluster, total number of incoming documents, number of clusters, number clustered documents in cluster i and number clustered documents in cluster i of class r. The entropy value if be closer to zero is better and has good efficiency.
7.2.2 Purity

Purity is sum maximum documents which located in the cluster i which are of the class r. The purity formula is:

\[ Purity = \sum_{i=1}^{k} \frac{n_i}{N} P(C_i) \]

As

\[ P(C_i) = \frac{1}{n_i} \max(n_i^r) \]

The entropy value if be closer to one is better and has good efficiency.

7.2.3 Fscore

Fscore is another item created by combination of above two items and is:

\[ FScore = \frac{\sum_{r=1}^{k} n_r F(Z_r, C_i)}{N} \]

As

\[ F(Z_r, C_i) = \frac{P(Z_r, C_i) * r(Z_r, C_i)}{P(Z_r, C_i) + r(Z_r, C_i)} = \frac{2 * n_i^r}{n_i + n_r} \]

\[ r(Z_r, C_i) = \frac{n_i^r}{n_r} \]

\[ P(Z_r, C_i) = \frac{n_i^r}{n_i} \]

The fscore value if be closer to one is better and has good efficiency.

In this paper, after implementation algorithms and above criteria for both categories, results are calculated and compared for analyzing. For testing implemented program, XML files consists of 100 different classes, such as medical files, colleges, shops, cars, insurance, etc... have been considered. As above mentioned, input files divided into two parts and with the XCLS+ and XCLS++ algorithms were evaluated separately. The results of algorithm on the homogeneous files are in Table 1 and include:
Table 1. Results of algorithms executed on homogeneous files

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Threshold</th>
<th>Entropy</th>
<th>Purity</th>
<th>FScore</th>
</tr>
</thead>
<tbody>
<tr>
<td>XCLS++</td>
<td>0.7</td>
<td>0.20</td>
<td>0.85</td>
<td>0.88</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.30</td>
<td>0.75</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.18</td>
<td>0.91</td>
<td>0.99</td>
</tr>
</tbody>
</table>

The results of algorithm on the heterogeneous files are in Table 2 and include:

Table 2. Results of algorithms executed on heterogeneous files

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Threshold</th>
<th>Entropy</th>
<th>Purity</th>
<th>FScore</th>
</tr>
</thead>
<tbody>
<tr>
<td>XCLS++</td>
<td>0.7</td>
<td>0.20</td>
<td>0.80</td>
<td>0.90</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>0.30</td>
<td>0.81</td>
<td>0.79</td>
</tr>
<tr>
<td></td>
<td>0.9</td>
<td>0.25</td>
<td>0.85</td>
<td>0.92</td>
</tr>
</tbody>
</table>

As previous section results, the results obtained in this section shows which the XCLS++ method has higher efficiency than the XCLS+ method. Also results show which proposed method has good efficiency for homogeneous files in comparison with heterogeneous files. Because probably for existence repeated node in homogeneous file is up.

8. Result and conclusion

The purpose of this paper is classification XML documents in order to expedite search and others benefits that classification has them. Criteria for classifying is structural or content and structure with content. The XCLS+ method is a method of classification methods which criteria for classification is done base on structure. Despite good performance for the XCLS+ method in compared with the XCLS method, ignoring repeated nodes in some documents causes it be inefficient. Therefore, this paper proposed \textsc{Father-Node-Chaild} and \textsc{Brother-Node} factors to the XCLS+ formula for achieving good efficiency. The results proposed idea as entropy, purity and Fscore show which the proposed method works better than the previous method. In future weight of levels will be changed for obtaining similarity number actually and better than proposed method too. Also the new method will be exam on much more documents in future and with comparing those, will be able to obtain better results.

9. References


10. Authors

Ahmad Khodayar received master science of computer engineering in 2010 from Islamic Azad University of Shabestar, Iran. His current research area is data mining and specially clustering; now he is working on a project about new way in clustering.

Hassan Naderi received his PhD degree in 2006 from INSA-LYON university of France. His current research areas are text mining, search engine and massive data processing.