Evaluating the Similarity of XML Documents Based on Frequent Label Sequences

Yijun Bei, Xiaolin Zheng, Zhen Lin

1 Corresponding Author
College of Software and Technology, Zhejiang University, byj@zju.edu.cn
2 College of Computer Science, Zhejiang University, xlzheng@zju.edu.cn
3 College of Computer Science, Zhejiang University, nblin@163.com

Abstract
To efficiently and yet accurately cluster XML documents is of great interests to web users. Among different methods to address the problem, clustering XML documents based on the frequent patterns in the documents seems to be a novel, interesting one. The intuition of the clustering criterion is that documents within the same cluster share more common sequences, while those belonging to different clusters share fewer or nothing. In this paper, we introduce the method using the frequent label sequences (FLS) as features to represent the XML documents and propose a new similarity metric to calculate the differences between documents based on the tag sequences. A framework of clustering algorithm for XML documents using frequent sequences is also presented. Experimental results show that our similarity metric outperforms the previous method CXP and can discriminate documents from different clusters effectively.

Keywords: XML Documents, Clustering, Frequent Label Sequences

1. Introduction
In recent years, XML documents have become ubiquitous with the rapid increasing in both number and scale of applications such as XML database systems, business transactions, XML middleware systems, and so on. Discovering knowledge from XML structural data have become more significant with the exponential growing of XML documents available on the Web. Among various approaches, XML document clustering is one of the useful mining approaches for knowledge discovery [4-10, 15, 22, 23, 26-27]. The objective of XML document clustering is to group XML documents with similar characteristics together, which can be used in broad applications including web mining, information retrieval, querying, and storage compression [1-3].

To group together similar XML documents, some clustering approaches have been proposed, which evaluate structural and content similarities of each pair using tree-based or vector-based representations of XML documents [4-10, 22, 23]. XML documents are considered in the same cluster if they are close according to a given distance model. However, in general, computing tree-edit distance or vector distance turns out to be unpractical, as it requires a quadratic number of comparisons among documents. Suppose, there are n documents, then n(n-1)/2 total times are required to compute distances among documents. Moreover, in some situations, tree-based distance model is unsuitable. For example, in Figure 1, we see the document T1 is more similar to T3 than that to T4. However, the distance between document T1 and T4 is closer than that between T1 and T3 when using tree-edit distance strategy.

Compared to tree-edit based clustering technique, frequent pattern based technique is also leveraged to evaluate the distance among XML documents [11-14, 28]. A frequent pattern mining approach is used to discover common occurred patterns from the XML dataset. These mined patterns are considered as the features of XML documents and leveraged to evaluate the similarities of documents. To further consider the structural levels of data in the documents, a document distance evaluation technique based on common path is proposed [5, 15]. Frequent common XPath (CXP) mined from the XML documents are used to form the feature vectors and to calculate the documents’ distance.

However, the similarity evaluation method proposed in PBClustering [5, 15] is not able to handle the hierarchical clustering problem of the documents in Figure 1. Considering the CXP results shown
in Table 1, we will find all documents in Figure 1 have the same frequent common XPath. This means we are not able to discriminate these four documents and have to put them into the same group. In fact, the document $T_4$ is more dissimilar from the others, and should be clustered into a different group. The differences of nodes at low level have small influence on the structural similarity for XML documents.

The failure of the PBClustering approach to clustering documents in Figure 1 is because the model is too simple to represent the documents. It just takes the containment between the documents and the sequences into consideration while ignoring the sequence characteristics, such as length, frequency, and continuity of the sequence in the original documents. In this paper, we study the characteristics between the sequences and documents and present a new similarity evaluation method FLS (Frequent Label Sequences). A clustering approach FSXC (Frequent Sequence-based XML Clustering) based on the vector model using frequent label sequences is also proposed. The contributions of our work are as follows:

More efficient and scalable clustering. Distance-based clustering approaches require a quadratic number of comparisons among documents. Furthermore, it is a time-consuming method to compute distance among XML documents. FSXC makes use of the frequent sequences to determine the clusters of documents. Experiments show that FSXC is more efficient and scalable than the traditional tree-edit based method.

High clustering accuracy. Maximal frequent label sequences are used to represent document features. When evaluating the document distances, information such as sequence length, frequency and continuity are considered and leveraged. Experimental results demonstrate that our approach outperforms common path-based clustering approach PBClustering in term of clustering quality.

The rest of the paper is organized as follows. In Section 2 we discuss previous works related to XML document clustering. In Section 3, we prepare basic definitions and present the framework for frequent sequence-based clustering algorithm for XML document set. In Section 4 we describe the document similarity evaluation strategy based on frequent sequences. Section 5 shows results of experiments and we make a conclusion about our work in Section 6.

2. Related works

XML document clustering approaches guarantee that documents in the same cluster have similar characteristics. One widely adopted technique is to evaluate the similarity between two documents by exploiting a tree-edit distance metric. The basic idea of tree-edit distance algorithm is to find the
sequence of editing operations that can transform one tree into another with minimum cost. The editing operations used in the tree editing problem include relabeling, deleting, inserting a node and deleting, inserting a subtree. Based on different editing operations, various approaches [16-19] have been proposed to compute tree distances employing dynamic programming techniques.

The tree edit distance can be easily used to cluster XML documents based on traditional clustering algorithms [6, 19]. Dalamagas et al. compute similarities on extracted structural summaries from documents by nesting and repeated reducing, then apply clustering algorithms using distances [6]. The insert-subtree and delete-subtree operations are also utilized to evaluate tree edit distance as well as inserting, deleting, and relabeling operations [19]. Besides tree edit distance, several other methods are also introduced to compute similarity among XML documents. Authors categorize XML documents into different clusters via the vector-space model based on three different feature sets: text features, tag features and combination of both [7]. A hierarchical algorithm S-GRACE is proposed using structure graph which summarizes the XML structure and supports efficient distance metric computation among documents [4]. Leung et al. employ a novel structural representation called CXP to form the feature vectors for XML documents and group documents according to CXP [5, 15].

Recently, some feature extracting approaches are proposed [23-27]. In [23], a novel method of representing the XML documents in Tensor Space Model is introduced and utilized for clustering. In [24], a simple approach is proposed to extract XML structural aspects from the documents. As a result, the flat data format about XML data is obtained and a clustering method is able to apply directly. Frozen structures in XML documents are mined to represent documents in [25]. Then, a novel algorithm called weighted cosine measure (WCM) is proposed in [25] via frozen structures. Authors in [27] firstly discovers frequently changing structures from a sequence of versions of dynamic XML documents and then use these frequent results to cluster dynamic XML documents.

3. Sequence-based XML clustering framework

3.1. Definitions

Basically, an XML document can be modeled as a rooted tree $T = \langle N, E, r, \sum, l \rangle$, including a node set $N$, an edge set $E$, a root $r$, a node label set $\sum$ and a node labeling function $l: N \rightarrow \sum$. Let $D = \{T_1, T_2, \ldots, T_n\}$ denotes all documents in an XML document set. XML clustering is the task of grouping XML documents in $D$ into clusters taking into consideration the similarities of documents. The problem of XML document clustering is presented as follows: Given an XML document set $D$ and group count $k$, clustering all documents in $D$ into $k$ groups such that documents are similar in the same group while dissimilar in different groups.

Definition 1 (Path). A path $\langle n_1, n_2, \ldots, n_l \rangle$ is a sequence of nodes in an XML document, where $n_i$ is the parent node of $n_{i+1}$ ($1 \leq i \leq l-1$) and $l$ is the length of the path. The sequence of node labels is considered as the label sequence.

Definition 2 (Rooted Path). A rooted path is a path with the first node as the root $r$.

Definition 3 (Repeated Path Node). Given two nodes $u$ and $v$ in the same level, if node $v$ has the same rooted path as node $u$, then node $v$ is considered as a repeated node.

Definition 4 (Nested Child Node). Given node $u$ and its child node $v$, if node $v$ has the same label as node $u$, then node $v$ is considered as a nested child node.

\[ \text{Figure 2. XML document reduction} \]
For example, in Figure 2, a path (1, 2, 6, 10, 13) is a rooted path with its label sequence A-B-A-A-C. Node 10 is a nested child node, while node 3 is a repeated path node compared to node 2.

3.2. XML clustering framework

Figure 3 shows the framework for XML document clustering, which partitions XML documents according to frequent label sequences. Before mining frequent label sequences from documents, we firstly perform a document reduction process, including nesting reduction and repetition reduction. We then use the sequence mining approach to mine maximal frequent label sequences from the reduced documents. Based on the discovered label sequences, sequence vectors representing the features of the documents are constructed. We then evaluate the document similarities by leveraging the feature vectors and employ the traditional clustering algorithm such as K-Means or hierarchical algorithms to cluster XML documents.

4. Document similarity evaluation

4.1. Node reduction

To evaluate similarities of XML documents, sequence vectors are used to represent the features of documents. However, repeated path node and nested child node must be preprocessed in order to improving clustering quality. This is because these two types of nodes do not change basic structures or fill additional contents with XML documents essentially. Otherwise, the clustering process will become more complicated if we do not delete them. Therefore, a reduction on repeated path node and nested child node is performed using the following strategy: If node $v$ is considered as a nested child node of $u$, then we remove node $v$ and set node $u$ as new parent node of all child nodes of $v$. If node $v$ is considered as a repeated path node compared to $u$, then we remove node $v$ and set node $u$ as new parent node of all child nodes of $v$. To perform the reducing process, a top-down method is adopted to remove those redundant nodes. Nodes are visited using breadth-first iterating manner.

For example, in Figure 2, document $T_1$ can be reduced to document $T_2$ by adopting the nesting reduction and $T_4$ can be reduced to document $T_3$ by removing repeated nodes. As a result, document $T_1$ is changed to $T_3$ through the reduction process.

After the process of document reduction, maximal sequence mining approaches such as [20, 21] are adopted to obtain maximal frequent label sequences from XML document set. In this way, features of XML documents are obtained using frequent sequences.

4.2. Similarity evaluation

Given a set of frequent sequences $S = \{s_1, s_2, ..., s_n\}$, each document can be represented by the sequence set $S$ using a n-dimensional vector according to the containment of each frequent sequence with respect to the document, where $n$ is the number of discovered sequences. The Score$(d, s_i) \ (1 \leq i \leq n)$ measures the goodness of a sequence $s_i$ for an XML document $d$. If document $d$ does not contain the $i$-th sequence, then the Score value is set to 0, otherwise the value is computed as follows:
where $|s_i|$ represents the length of the sequence $s_i$, $\text{Max}(\{s \mid s \in S \land \text{contain}(d, s_i)\})$ represents the maximal length of all sequences in $S$ which are contained by document $d$, and $SP(d, s_i)$ denotes the shortest continuous sequence in document $d$ containing the frequent sequence $s_i$. $w_1$, $w_2$, and $w_3$ are three weighted coefficients. The Score of a document with respect to a frequent sequence contains three aspects: 1) the base value; 2) a value considering the length of the sequence; 3) a value considering the continuity of the sequence.

For example, consider the minimum support 0.75 used in Figure 1. All the sequences A-B-D, A-B-E and A-C are the maximal sequences. For all maximal frequent sequences contained in document $T_4$, the sequence A-B-E is the longest. Thus we have $\text{Max}(\{s \mid s \in S \land \text{contain}(T_4, s_i)\}) = 3$. However, it is not a continuous sequence in the document $T_4$. As a result, in term of sequence continuity, we should adopt a lower impact in documents.

Using our proposed method, we can construct the sequence vectors in Table 2 from the documents in Figure 1. From the table we see that document $T_4$ can be easily discriminated from the other three documents, while we are not able to discriminate them if we use the CXP method.

### 4.3. Clustering algorithm

**Algorithm FSXC(XMLDB, minsup, k)**

**Input**: XML document set XMLDB, minimum support minsup, clustering count $k$

**Output**: $k$ groups

1. $RDB = \text{ReduceXMLDB}(XMLDB)$;
2. $MFS = \text{MineMaxSequence}(RDB)$;
3. $DV = \text{ConstructVector}(RDB, MFS)$;
4. Assign initial centroid for the $k$ clusters
5. Repeat
6. For each document features $dv$ in $DV$
7. Assign document $dv$ to the cluster which has the closet mean
8. Calculate new centroid of each cluster;
9. Until centroids do not change
10. Return the $k$ clusters

**Figure 4. XML document clustering algorithm**
In Figure 4 we show the XML document clustering algorithm using the FLS as the features of documents. The K-Means clustering strategy is leveraged to perform the clustering. As previously stated, XML documents are reduced and maximal frequent sequences are mined (Lines 1-2). Then the feature vectors of documents are constructed (Line 3). Traditional clustering algorithm K-Means are employed to cluster documents (Line 4-10). Using the state-of-the-art frequent mining approaches, only one iteration of XML documents are required to discover those sequential features. The time complexity of the K-Means algorithm is $O(\text{tkn})$, where $t$ is iterating number of XML document vectors, $k$ is the targeting count of groups and $n$ is the total number of XML documents.

5. Experiments

We compare our algorithms with the common XPath (CXP) proposed in [5, 15]. We also present experimental results of our FSXC clustering algorithm compared to previous distance-based algorithms. We implement both FSXC algorithm and tree-edit distance based algorithm in Java language (JDK1.7) and carry out all experiments in the environment of 2.0GHz Pentium dual machine with 4G memory running Ubuntu 11.10. For the tree-edit distance based XML clustering, we adopt the computing method of tree-edit distance similar to the algorithm presented by Dalamagas [6]. Experiments are analyzed from the aspects of clustering quality and efficiency.

We use XML generator provided by IBM\(^1\) to generate documents from the DTDs supplied by ACM SIGMOD\(^2\): SigmodRecord.dtd (A), OrdinaryIssuePage.dtd (B), HomePage.dtd (C). For XML generator, we set the parameters with MaxRepeats=5 and NumberLevels=12, where MaxRepeats represents the maximal occurring number of nodes and NumberLevels indicates the level of documents.

We generate 1000 documents from each DTD and obtain an XML dataset totally containing 3000 documents. Documents from the same DTD are considered homogenous while documents from different DTDs are heterogeneous. We set the parameters with $w_1=0.5$, $w_2=0.25$, $w_3=0.25$ and minimum support 0.33.

![Figure 5. Evaluation of document distances](image)

**Clustering Quality.** In Figure 5, the similarities among documents from three DTDs are analyzed. We use the Euclidean distance to evaluate the similarity among these documents. For example, (A, A) denotes the average distance of documents generated by the SigmodRecord.dtd, while (A, C) represents the average distance between documents generated by the SigmodRecord.dtd and HomePage.dtd. The means and variances of the distances for the evaluation methods of FLS and CXP are evaluated respectively. Figure 5(a) shows that the distance value obtained by the FLS are quite similar to CXP. However, distance value is slightly smaller than the value computed by CXP. The reason is we use a different similarity metrics to calculate the XML document similarity. However, our method has lower variance than CXP as shown in Figure 5(b). This indicates that our method achieves

\(^1\) http://www.alphaworks.ibm.com
\(^2\) http://www.sigmod.org/record/xml/
a better discrimination among documents from heterogeneous sources. Meanwhile, documents from the same DTD have very low distances as well as variances.

We analyze the clustering quality of XML documents using the following method:

\[
\text{precision} = \frac{\sum \text{RightCluster}(j)}{n}
\]

where RightCluster(j) means the document j is assigned to a right cluster.

Figure 6. Evaluation of clustering quality

In Figure 6(a), the clustering accuracy is presented on the different size of document sets from 6,000 to 42,000. As all the documents are generated only from three DTDs, there should be three groups in the final clustering result. From Figure 5, we find FLS-based algorithm FSXC outperforms CXP-based algorithm PBClustering in term of clustering quality. This is consistency with the evaluating result on document distances. When employing FLS to represent document features, not only sequence length, but also sequence continuity in the original document is considered. In addition, as the size of the data set increases, the number of document features, i.e. maximal frequent sequences keep stable.

In Figure 6(b), the clustering accuracy is also tested on the various minimum supports. When the minimum support is set to 0.3, the best clustering quality is obtained. Clustering quality is decreased whenever the value becomes large or small. This is because all the documents are generated by three different DTDs. When the support value decreases, the number of frequent sequences increases. This means discrimination of the same document is also increased. As a result, we may put documents from the same DTD into different groups. On the contrary, when support value increases, the discovered frequent sequences may be contained by all documents. Consequently, these frequent sequences are not able to differentiate documents and become poor features.

Figure 7. Evaluation of clustering performance

Clustering Performance. In Figure 7(a) we present the performance our algorithm with different sizes of datasets from 300 to 15,000 on different parameters for document generation. Here, the method TDXC stands for the tree-edit distance based XML clustering algorithm. From the figure we can find FSXC algorithm is faster than TDXC for clustering XML documents, especially for the large dataset.
Consumed time by FSXC algorithm mainly depends on the mining process. However, many existed mining algorithms are efficient and linear to the size of dataset set. For the TDXC algorithm, computing tree distances have large cost. Suppose a dataset with $n$ documents, TDXC needs $n(n-1)/2$ times of comparisons among documents, which leads to an inefficient clustering process and a low scalability. When the dataset grows, the running time spent on TDXC increases drastically.

In Figure 7(b), the experiment shows that our algorithm presents good scalability. When dataset grows, the running time doesn’t rise fast. As mentioned earlier, the consumed time using our algorithm is determined by the mining process which is proportional to the size of dataset. Therefore, good scalability can be obtained using FSXC compared to TDXC.

6. Conclusion

In this paper, we use the frequent label sequences to represent XML documents as features and present the effective similarity metric to evaluate the distance among documents. Unlike CXP method, we consider sequence characteristics, such as length, frequency and continuity of sequences in the original documents when evaluating document distances. We also present a framework for the clustering algorithm for XML documents using sequence features. Experimental results demonstrate that our similarity metric can discriminate documents from heterogeneous sources effectively and have good scalability.

7. Acknowledgments

The work was supported in part by National Natural Science Foundation of China [grant numbers 60903038 and 61003254]; the Fundamental Research Funds for the Central Universities.

8. References


