Clustering Large Scale of XML Documents

Tong Wang[1], Da-Xin Liu[1], Xuan-Zuo Lin[2], Wei Sun[1], and Gufran Ahmad[1]

[1] Department of Computer Science and Technology, Harbin Engineering University, China
Wangtong@hrbeu.edu.cn
[2] Northeast Agriculture University, Harbin, China
xuanzuolin@sina.com

Abstract. Clustering is able to facilitate Information Retrieval. This paper addresses the issue of clustering a large number of XML documents. We propose ICX algorithm with a novel similarity metric based on quantitative path. In our approach, each document is firstly represented by path sequences extracted from XML trees. Then these sequences are mapped into quantitative path, by which the distance between documents can be computed with low complexity. Finally, the desired clusters are constructed by utilizing ICX method with literal local search. Experimental results, based on XML documents obtained from DBLP, show the effectiveness and good performance of the proposed techniques.

1 Introduction

Since XML is becoming the pervasive web data exchange format, much research effort is currently devoted to support the storage and retrieval of large collections of such documents. Our research is driven by the hypothesis that closely associated documents tend to be relevant to the same requests, so that grouping similar documents accelerates the search [1]. However, traditional text clustering approaches [2] didn't take the structural information of XML into account. This paper considers the structure of XML and extracts paths from documents.

Many researchers [3][4][5] measure structural similarity using the "edit distance" between tree structures. However, the edit distance between two documents has time complexity at least $O(n^2)$ and the algorithm requires computing the distance for each document-pair. Thus, it is unsuitable for a collection of large documents.

In this paper, we introduce a novel indirect clustering method ICX for XML documents. The contributions are as follows: ① a novel distance metric is proposed based on quantitative path sequence, called QX_path. This metric calculation is simple with low computational complexity, which is fit for clustering high-volume XML documents. ② Based on the metric above, an improved C-means ICX method is proposed. This method solves local optima problem and the experiment compared with C-means shows its efficiency.

The remaining of the paper is organized as follows: section 2 is the feature extraction and similarity metric; section 3 introduced ICX clustering method; section 4 shows experiment evaluation. We conclude in section 5.

Y.-C. Chung and J.E. Moreira (Eds.): GPC 2006, LNCS 3947, pp. 447 – 455, 2006.

2 Model Representation and Similarity Metric

Compared to traditional Vector Space Model (VSM), we use a different metric, called QX_path. In this model, each document is expressed by path sequences, and then is transformed to QX_path according to tag-mapping table. Finally, we define the similarity calculation, which has a lower time complexity.

2.1 Document Representation

XML document can be viewed as a labeled tree. In our case, we define here document model tree D_T.

Definition 1. XML document tree. Suppose a countable infinite set E of element labels (tags), a countable infinite set A of attribute names. An *XML document tree* is defined to be $d = (V, lab, ele, att, v_r)$ where V is a finite set of nodes in d; lab is a function from V to $E \bigcup A$; *ele* is a partial function from V to a sequence of V nodes such that for any $v \in V$, if *ele(v)* is defined then $lab(v) \in E$; *att* is a partial function from $V \times A$ to V such that for any $v \in V$ and $l \in A$, if $att(v,l) = v_1$, then $lab(v) \in E$ and $lab(v_1) = l$; v_r is a distinguished node in V called root of d, $lab(v_r) = root$.

Figure 1 shows an example of XML document tree. The model is a rooted, directed, and unordered tree. A path in D_T is sequence of nodes $v_1, v_2, v_3, ..., v_n$, through which we can traverse step by step in D_T. In addition, there exists one and only one path from node v_i to node v_j for each v_i and v_j, $v_i \neq v_j$. Before we formally define QX_path, we first give the definition of X_path.

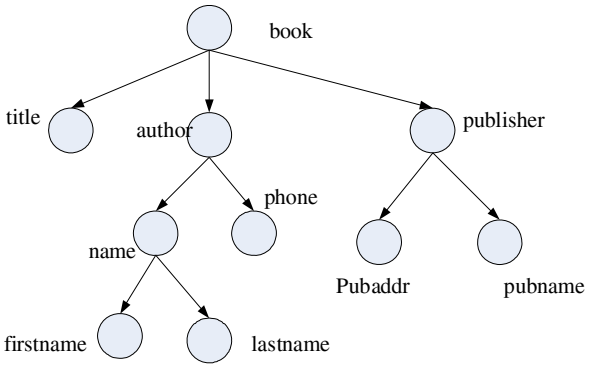

Fig. 1. XML document tree

Definition 2. Path Sequence. Given v_i and D_T, the path sequence of v_i is defined as an ordered sequence of tag names from *root* to v_i, denoted as

$$X\_path_{v_i}^{D_T} = \{v_0, v_1, ..., v_m\} \text{ where } v_k \in \sum_{D_r} \quad ,k \in [1...m]$$

Given node v_i and v_p, we define v_i is *nested* in v_p w.r.t. $i < p \wedge v_i, v_p \in X\_path_{v_i}^{D_T}$. Note that X_path describes hierarchical and structural information of the XML document D_{Tk}, for it shows how v_i is nested in D_{Tk}. Furthermore, in XML document collection, an XML document D_{Tk} consists of many $X\_path_{v_i}^{D_{Tk}}$ sequences, denoted as follows.

$$D_{Ti} = \{X\_path_{v1}^{D_{Tk}}, X\_path_{v2}^{D_{Tk}}, ..., X\_path_{vn}^{D_{Tk}}\} . \quad (1)$$

2.2 Similarity Metric

For clustering methods, similarity metric is foremost and related directly to the computing complexity. This paper proposes a novel distance metric based on quantitative QX_path with low complexity. And our idea is inspired by numeric transformation of liner space [10]: Let the size of the alphabet be c, with an established order on the symbols in the alphabet. Choose an integer d>2c. Let a string of length n be $S_1, S_2, ..., S_n$, with each symbol S_i mapped to an integer t_i between 1 and c. t_i depicts the symbol position in the string, where $1<t_i<c$. Thus, the string can be mapped to linear expression $t_1/d + t_2/d^2 + ... + t_n/d^n$.

In our approach, we map X_path to numeric QX_path, by which we compute the similarity metric between each documents. Given document collection D_{Tall}, occurring tags collection T_{all} and size of tags collection $|T_{all}|$, X_paths can be translated to QX_paths according to the tag-mapping table. When the mapping table is constructed, WordNet Java API [7] is employed to consider semantics of *tag* (v_i) in each document and determine whether two tags are synonyms.

Table 1. Tag-mapping table of Fig.1

tag	position	tag	position
book	1	phone	6
title	2	pubaddr	7
author	3	pubname	8
publisher	4	firstname	9
name	5	lastname	10

Definition 3. Quantitative XML path sequence (QX_path). Given $N_{D_{Tk}}(v_i)$, the mapping function from v_i of D_{Tk} to numeric representation, we define the quantitative path as follows:

$$QX\_path_{v_i}^{D_{Tk}} = N_{D_{Tk}}(v_1)/d + N_{D_{Tk}}(v_2)/d^2 + ... + N_{D_{Tk}}(v_i)/d^i . \quad (2)$$

where $d=2|T_{all}|+1$.

The method has the ability to preserve "prefix " properties: the nearer the node to the root, the more discriminable contribution the node has to the structure. That is to say, we needn't represent all the paths from root. When the distance of paths is computed, the root-to-leaf paths can represent almost all the structural information extracted from the XML tree.

Note that equation 1 can be turned into equation 3. In this process, the structural information can be mapped into rational numbers, which can reduce the expense of similarity calculation subsequently.

$$D_{Ti} = \{QX\_path_{v1}^{D_{Tk}}, QX\_path_{v2}^{D_{Tk}}, ..., QX\_path_{vn}^{D_{Tk}}\} \ . \tag{3}$$

Example1. As is shown in table1, T_{all} ={book, title, author, publisher, name, phone, pubaddr, name, firstname, lastname}, $|T_{all}|$=10. Let's take some paths as an example:

$X\_path_{firstname}^{D_T} = \{book, author, name, firstname\}$,

$X\_path_{title}^{D_T} = \{book, title\}$.

The corresponding rational number will be:

$QX\_path_{firstname}^{D_T} = N_{D_T}(book)/|T_{all}| + N_{D_T}(author)/|T_{all}|^2 +$

$N_{D_T}(name)/|T_{all}|^3 + N_{D_T}(firstname)/|T_{all}|^4 = 1/21 + 3/21^2 + 4/21^4 + 5/21^5$

$QX\_path_{title}^{D_T} = N_{D_T}(book)/|T_{all}| + N_{D_T}(title)/|T_{all}|^2 = 1/21 + 2/21^2$

Based on the above feature extraction, we define the distance metric between D_{Tx} and D_{Ty} .

$$Dist(D_{Tx}, D_{Ty}) = \sqrt{\sum_{v_i \in D_{Tx}} \sum_{v_j \in D_{Ty}} |QX\_path_{v_i}^{D_{Tx}} - QX\_path_{v_j}^{D_{Ty}}|} \ . \tag{4}$$

The time complexity of the metric calculation is satisfactory. Traditional distance metrics often requires mapping features into vectors and dealing with them in the high-dimensional Vector Space Model (VSM) [11][12], in which time complexity is expensive. While the Quantitative path is an indirect distance method, for it only measures the documents through paths they contain.

Meanwhile, as mentioned above, tree edit distance [3] is unfit for large XML documents, because the computational expense of metric is $O(n^2)$. In our case, the distance between two documents has time complexity $O(p^2)$, where p denotes the scales of QX_path s and is far less than the scale of document collections.

3 ICX Cluster Technique

Document clustering is to categorize the documents based on similarity without the prior knowledge on the taxonomy. And it has two beneficial aspects: efforts in integrating XML documents with different structures and semantics can be alleviated because reconciling analogous and relatively small document collection is easier. Besides, ranges of queries can be dramatically decreased to applicable documents after relevant documents are aggregated together.

Section 2 introduces the distance metric of the clustering. In this chapter, we at first introduce the basic C-means clustering method briefly. Then, we present the **I**mproved **C**-means methods for **X**ML document, called ICX.

3.1 Standard C-Means Clustering

C-means is a partitional clustering algorithm based on the firm foundation of analysis of variances. It clusters a group of data objects into a predefined number of clusters. It starts with randomly initial cluster centroids and keeps reassigning the data objects in the dataset to centroids based on the similarity between the data object and the centroids. The reassignment procedure will not stop until a convergence criterion is met (e.g., the fixed iteration number, or the cluster result does not change after a certain number of iterations). The C-means algorithm can be summarized as:

1. Randomly select cluster centroids to set an initial dataset partition.
2. Assign each data object to the closest cluster centroids.
3. Recalculate the cluster centroid $c_j = \frac{1}{n_j} \sum_{\forall d_j \in S_j} d_j$

   where d_j denotes the data object that belong to cluster S_j; c_j stands for the centroid; n_j is the number of data object that belong to cluster S_j.
4. Repeat step 2 and 3 until the convergence is achieved.

3.2 ICX Method

C-means algorithm is efficient, with time complexity $O(ntk)$, where n is the size of dataset, k is the clusters and t is the circle time. Recent studies have shown that partitional clustering algorithms are more suitable for clustering large datasets [6].

However, It is well known that the main drawback of the C-means algorithm is that the result is sensitive to the selection of the initial cluster centroids and may converge to the local optima [14]. To solve the problem, ICX method is proposed. The main idea is that when a solution can be no more improved the algorithm makes the next iteration after an appropriate disturbance on the local minimum solution. Thus the algorithm can skip out of the local minimum and in the meanwhile, reach the whole search space.

Algorithm ICX
Input: n: number of XML collection; k: number of clusters
Output: k cluster

1. Randomly select one initial partition $P_k = \{C_i\}, (i = 1, ..., k)$
2. Initialize current best partition
3. Give terminate condition of algorithm; $\varepsilon > 0$; maximum iterant times n of object function;
4. do {
5. Search locally in P_k to get a local minimum $f_{local-opt}$ and its corresponding partition P_k^*;

6. do
7. { $P_k = P_k^*$; $f_{local-opt} = f_{opti}$;}
8. until ($f_{local-opt} > f_{opt}$);
9. Randomly select one object v_i
10. If v_i is not chosen, Assign v_i to other clusters by computing:

$$\triangle f = \sum_{i=1}^{k} \sum_{v_i \in S_i^*} | v_i - c_i^{'} |^2 - \sum_{i=1}^{k} \sum_{v_i \in S_i} | v_i - c_i |^2$$

11. t=t+1
12. If $\Delta f < \varepsilon$
13. $f_{local-opt} = f_{local-opt} + \Delta f$
14. If $f_{local-opt} < f_{opti}$
15. $f_{opti} = f_{local-opt}$; $P_k^{'} = P_k$; $t = 0$
16. } until (t<n)

At first, ICX algorithm randomly finds a local centroid vector by standard c-means clustering method. From the line 9 to line15 indicates the local search process. Firstly, we select a vector v_i of cluster S_i randomly and reassigned to cluster S_j, the centroid vectors will be updated according to equations $c_i^{'} = \frac{n_i \times c_i - v_i}{n_i - 1}$ and $c_j^{'} = \frac{n_j \times c_j + v_i}{n_j + 1}$, where n_i and n_j is the number of XML documents. Then, we measure the influence of this reassignment using the increment $\triangle f = \sum_{i=1}^{k} \sum_{v_i \in S_i^*} | v_i - c_i^{'} |^2 - \sum_{i=1}^{k} \sum_{v_i \in S_i} | v_i - c_i |^2$. If $\triangle f < \varepsilon$, the algorithm regards current partition as local minimum and starts the local search; otherwise, the algorithm assigns the vector v_i to other clusters. During the process, if v_i is assigned for k times, we have to try to choose another $v_j (v_i \neq v_j)$. The disturbance can help for skipping out of the local resolutions to improve the quality of cluster solutions.

4 Experiments and Analysis

Our experiments were conducted on a workstation of 1.5GHz Intel Pentium 4 machine with 512 MB main memory.

4.1 Dataset

We choose a variety of XML datasets including two widely used real datasets and one synthetic dataset, Xmark. One real dataset is obtained from DBLP [16], the bibliographical data of scientific conferences and journals; the other is Swiss Prot, a real-life data set with annotations on proteins; Xmark, a synthetic dataset that models

transactions on an on-line auction site. Compared with DBLP, the data in Xmark is relatively tilted and sparse, with more complex structures.

The test subset of DBLP we used consists of 10 different ACM Journals. Each journal with 100 documents is grouped, denoted by $G_i, 1 \le i \le 10$. We mix these documents together and cluster them for our test. In the context of clustering, we can also produce 10 categories, denoted by $C_i, 1 \le i \le 10$. Similarly, the subset of Protein set contains 1324 document that have been classified into 54 categories.

For the synthetic dataset, Xmark, our experiment is based on the hypothesis that the documents with the same DTD will be clustered in the same class. When we generate files using Xmark, the scale parameter of Xmark is 0.2. That is, each generated document is 20M or so. We get 5 DTD (Data Type Definition) documents [18] and for each DTD generate 20, 40, 60, 80, 100 XML documents, respectively. The five generated datasets are denoted as Xmark1, Xmark2, Xmark3, Xmark4 and Xmark5, respectively.

4.2 Measurement

In order to measure the clustering accuracy, we take the DBLP as an example. As mentioned above, the groups we specify beforehand are denoted by G_i, and the final clustered groups in the experiments are denoted by C_i. The δ function is given by

$$\delta(d_1, d_2, C_i) = \begin{cases} 0, \ if \ \exists j, d_1, d_2 \in G_j \\ 1, \ if \ \neg\exists j, d_1, d_2 \in G_j \end{cases} . \tag{5}$$

where d_1 d_2 are documents from C_i category. To quantify the clustering accuracy of ICX technique, we define Classified Error Rate (CER) as follows.

$$CER = \frac{\sum_i \sum_{m,n \in C_i \wedge m \neq n} \delta(m, n, C_i)}{\sum_i [i \times (i-1)/2]} \tag{6}$$

If there is no pair of documents occurring in both C and G classes, the error rate will reach the maximum value, e.g., combination $C_i^2 = i \times (i-1)/2$. CER is a relative error rate value, $0 \le CER \le 1$.

4.3 Results Analysis

In order to compare to naïve method, which uses the standard C-means method, we also implement the naïve clustering method. Besides, the documents were parsed into labeled trees via the parser developed by Zhang et al [15] in pre-process.

In the stage of standard C-means procedure, the choice of k is often ad hoc, larger than the number of classes in general. In our case, we choose the class number. Since C-means is sensitive to the input order of vectors, we did each experiment several times and obtained the mean of CER. Fig.2 shows the results of the two methods.

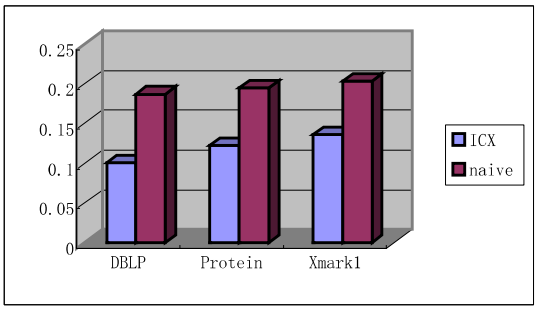

Fig. 2. Classified Error Rate of two methods: ICX and naïve method

The first case is to test the accuracy of ICX method. From figure2, for all the datasets, it is obvious that CER value of ICX outperformed that of naïve clustering. That's to say, the local search in ICX method significantly improves the clustering quality.

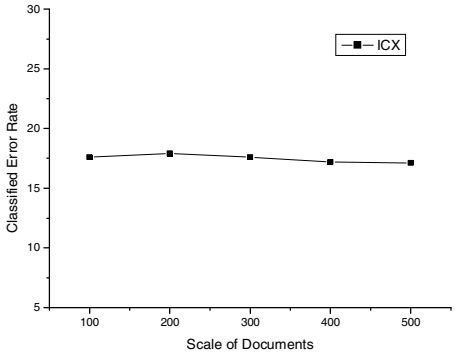

Fig. 3. The scalability of ICX method. The test dataset is Xmark datasets: Xmark1 (n=100), Xmark2(n=200), Xmark3(n=300), Xmark4(n=400) and Xmark5(n=500), respectively.

Then, we test the scalability of ICAXC. In this experiment, Xmark1, Xmark2, Xmark3, Xmark4 and Xmark5 are used as the test dataset, one by one. Figure 3 shows that the Classified Error Rate of these dataset varies very small when the number of the documents increases. It demonstrates that ICX algorithm is a robust and stable algorithm when the scale of dataset is large. Thus, the proposed method can be used for clustering a high-volume XML documents collection over the web.

5 Conclusion

In order to cluster high-volume XML documents efficiently, we have proposed a indirect distance metric based on quantitative path information. To improve the clustering quality, an improved C-means clustering is applied in our case. Experimental results show the proposed method is efficient for large scale of XML documents.

References

1. Faloutsos C. Oard D. A survey of information retrieval and filtering methods. Department of Computer Science. University of M aryland, Technical Report, CS-TR-3514, (1995)
2. O. Zamir, O. Etzioni, O. Madani, and R.M. Karp, "Fast and Intuitive Clustering of Web Documents," Proc. Second Int'l Conf.Knowledge Discovery and Data Mining,(1997) 287-290
3. A. Nierman and H.V. Jagadish, "Evaluating Structural Similarity in XML Documents," Proc. Fifth Int'l Workshop Web and Databases (2002)
4. Gianni Costa, Giuseppe Manco, Riccardo Ortale, Andrea Tagarelli: A Tree-Based Approach to Clustering XML Documents by Structure. PKDD 2004. (2004) 137-148
5. Theodore Dalamagas, Tao Cheng, Klaas-Jan Winkel, Timos K. Sellis: A Methodology for Clustering XML documents using Tree Summaries and Structural Distance Metrics. HDMS (2004)
6. Al-Sultan, K. S. and Khan, M. M..Computational experience on four algorithms forthe hard clustering problem. Pattern Recogn. Lett.17, 3, (1996) 295–308.
7. George A. Miller, Richard Beckwith, Introduction to WordNet: An On-line Lexical Database International journal of Lexicography, 3(4), (1990)235-312.
8. Mong-Li Lee, Liang Huai Yang, Wynne Hsu, Xia Yang: XClust: clustering XML schemas for effective integration. CIKM 2002, 292-299
9. Aoying Zhou, Weining Qian, Hailei Qian: Clustering DTDs: An Interactive Two-Level Ap-proach. J. Comput. Sci. Technol. 17(6) (2002) 807-819
10. H. V. Jagadish, Nick Koudas, Divesh Srivastava: On Effective Multi-Dimensional Indexing for Strings. Proceedings of the ACM SIGMOD Conference on Management of Data. (2000) 403-414
11. Antoine Doucet, Helena Ahonen-Myka: Naive Clustering of a large XML Document Collection. INEX Workshop 2002.(2002) 81-87
12. X. Cui, T. E. Potok, and P. Palathingal, Document Clustering using Particle Swarm Optimi-zation, In Proceedings of the 2005 IEEE Swarm Intelligence Symposium, June, 2005, Pasa-dena, California, USA,(2005)
13. Abiteboul, S., Buneman, P., Suciu, D.: Data On The Web: From relations to Semistructured Data and XML. Morgan Kaufmann Publishers, San Francisco, California,(2000)
14. Selim, S. Z. And Ismail, M. A.. K-means type algorithms: A generalized convergence theorem and characterization of local optimality. IEEE Trans. Pattern Anal. Mach. Intell. 6, (1984) 81–87
15. S. Zhang, J. T. L.Wang, and K. G. Herbert. Xml query by example. International Journal of Computational Intelligence and Applications, Vol. 2, No.3 (2002) 329–337
16. DBLP Computer Science Bibliography. 2004. http:// www.informatik.uni-trier.de/~ley/db/