Using Proportional Transportation Similarity with Learned Element Semantics for XML Document Clustering

Xiaojun Wan, Jianwu Yang
Institute of Computer Science and Technology, Peking University, Beijing 100871, China
{wanxiaojun, yangjianwu}@icst.pku.edu.cn

ABSTRACT

This paper proposes a novel approach to measuring XML document similarity by taking into account the semantics between XML elements. The motivation of the proposed approach is to overcome the problems of "under-contribution" and "over-contribution" existing in previous work. The element semantics are learned in an unsupervised way and the Proportional Transportation Similarity is proposed to evaluate XML document similarity by modeling the similarity calculation as a transportation problem. Experiments of clustering are performed on three ACM SIGMOD data sets and results show the favorable performance of the proposed approach.

Categories and Subject Descriptors:

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *clustering*

General Terms: Theory, Experimentation

Keywords: XML document clustering, Proportional Transportation Similarity

1. INTRODUCTION

XML document clustering aims to group XML documents into different clusters with a standard of document similarity. Much previous work has explored clustering methods for grouping structurally similar XML documents so as to find common structures or DTDs for various purposes. While in this study we focus on grouping topically similar XML documents, i.e., the text contents of the XML documents within a cluster express a similar topic or subject. Based on the traditional Vector Space Model (VSM), XML documents are processed as ordinary unstructured documents by removing all element tags and thus the structural information is totally lost. Another intuitive method takes into account the structural information by computing the similarity of texts within each element respectively and then combining these similarities linearly, which is called C-VSM. Much work [1, 4] explores this problem by extending VSM in order to incorporate the element tag information. For SLVM proposed in [4], each document, doc_x, is represented as a matrix $d_x \in R^{n \times m}$, given as $d_x=<d_{x(1)},d_{x(2)},\ldots,d_{x(n)}>^T$, $d_{x(i)}=<d_{x(i,1)},d_{x(i,2)},\ldots,d_{x(i,m)}>$, where m is the number of XML elements, and n is the number of terms. $d_{x(i)} \in R^m$ is a feature vector related to the term t_i for all the elements, $d_{x(i,j)}$ is a feature related to the term t_i and specific to the element e_j, given as $d_{x(i,j)}= TF(t_i,doc_x.e_j)*IDF(t_i)$ and $TF(t_i,doc_x.e_j)$ is the frequency of the term t_i in the element e_j of the documents doc_x. And each $d_{x(i,j)}$ is normalized by $\sum_i d_{x(i,j)}$. The similarity between two documents doc_x and doc_y is then defined with an element semantic matrix M_e introduced, given as

$$sim_{\mathrm{SLVM}}(doc_x, doc_y) = \sum_{i=1}^{n} d_{x(i)}{}^T \bullet M_e \bullet d_{y(i)} \qquad (1)$$

where M_e is an $m*m$ element semantic matrix which captures both the similarity between a pair of XML elements as well as the contribution of the pair to the overall document similarity.

The popular way to determine the value of M_e is based on the edit distance [5]. In order to acquire a more appropriate element semantic matrix, with the notion that term similarity should be affecting document similarity and vice versa, we propose an iterative algorithm for learning the element semantic matrix M_e, given as

$$S_d = (\sum_{i=1}^{n} \lambda_i \cdot B_{(i)}{}^T \bullet M_e \bullet B_{(i)}) / n \qquad (2)$$

$$M_e = (\sum_{i=1}^{n} \lambda_i \cdot B_{(i)} \bullet S_d \bullet B_{(i)}{}^T) / n \qquad (3)$$

$\vec{B} =< B_{(1)}, B_{(2)}, \cdots\cdots, B_{(n)} >^T$ represents a set of XML documents, where $B_{(i)} \in R^{m \times p}$ is a matrix with its k^{th} column corresponding to $d_{k(i)}$ of the *k-th* document and p is the number of documents. $\lambda_i = 0.9 / \max(|B_i|_1, |B_i|_\infty)$. All the entries' values of matrix S_d are normalized between zero and one. Two totally different documents have a similarity value of zero and two identical documents have a similarity value of one. An additional constraint for getting a non-trivial solution of having both matrices with all zero elements is required to force the diagonal elements of S_d (i.e., the similarity of identical documents) to take the value of one.

VSM does not consider structural information and C-VSM only allows the text within an element of one document to correspond to the text within the same element of the other document. The two cases ignore the semantic relationship between different elements and have "under-contribution" problem. SLVM overcomes the above "under-contribution" problem by allowing the text within an element of one document to correspond to the text within any element of the other document. However, the corresponding relation between elements is loose, i.e. the text with a weight within an element of one document can always use its total weight to correspond to the text within any element of the other document, which is believed to have the so-called "over-contribution" problem.

In order to address the above problems of "under-contribution" and "over-contribution", illuminated by the Proportional Transportation Distances [2], we propose the Proportional Transportation Similarity (PTS) to measure the document similarity. PTS allows the text within an element of one document to correspond to the text within any element of the other document

WWW 2006, May 23–26, 2006, Edinburgh, Scotland.
ACM 1-59593-323-9/06/0005.

under a few strict constraints by modeling a transportation problem in the linear programming field. The document similarity over one single term is computed as follows:

Given two feature vectors $d_{x(i)}=<d_{x(i,1)},d_{x(i,2)},\ldots,d_{x(i,m)}>$ and $d_{y(i)}=<d_{y(i,1)},d_{y(i,2)},\ldots,d_{y(i,m)}>$, related to the particular term t_i for all the elements in documents doc_x and doc_y respectively, where m is the number of elements, a weighted graph G is constructed as follows:

Let $d_{x(i)}=<d_{x(i,1)},d_{x(i,2)},\ldots,d_{x(i,m)}>$ as the weighted point set for the term t_i in document doc_x, $d_{x(i,j)}$ is a feature related to the term t_i and specific to the element e_j given as the $TF(t_i,doc_x.e_j)*IDF(t_i)$ value.

Let $d_{y(i)}=<d_{y(i,1)},d_{y(i,2)},\ldots,d_{y(i,m)}>$ as the weighted point set for the term t_i in document doc_y, $d_{y(i,j)}$ is a feature related to the term t_i and specific to the element e_j given as the $TF(t_i,doc_y.e_j)*IDF(t_i)$ value..

Let $G=\{d_{x(i)}, d_{y(i)}, M_e\}$ as a weighted graph constructed by $d_{x(i)}$, $d_{y(i)}$, and M_e. $V=d_{x(i)} \cup d_{y(i)}$ is the vertex set while M_e is the edge matrix, either learned or obtained based on edit distance. $W_{d_{x(i)}}$, $W_{d_{y(i)}}$ are the total weights of $d_{x(i)}$, $d_{y(i)}$, i.e. the sums of all weights of points within $d_{x(i)}$, $d_{y(i)}$, respectively.

Based on the weighted graph G , the possible flows $\zeta = [f_{uv}]$, with f_{uv} the flow from $d_{x(i,u)}$ to $d_{y(i,v)}$, are defined by the following constraints:

$$f_{uv} \geq 0 \qquad 1 \leq u \leq m \quad 1 \leq v \leq m \tag{4}$$

$$\sum_{v=1}^{m} f_{uv} = d_{x(i,u)} \qquad 1 \leq u \leq m \tag{5}$$

$$\sum_{u=1}^{m} f_{uv} = \frac{d_{y(i,v)} W_{d_{x(i)}}}{W_{d_{y(i)}}} \qquad 1 \leq v \leq m \tag{6}$$

$$\sum_{u=1}^{m}\sum_{v=1}^{m} f_{uv} = W_{d_{x(i)}} \tag{7}$$

Constraint (4) allows moving weights from $d_{x(i)}$ to $d_{y(i)}$ and not vice versa. Constraint (5) and (7) force all of $d_{x(i)}$'s weight to move to the positions of points in $d_{y(i)}$. Constraint (6) ensures that this is done in a way that preserves the old percentages of weight in $d_{y(i)}$.

And PTS is given by

$$PTS\,(d_{x(i)}, d_{y(i)}) = \max_{F \in \zeta} \sum_{u=1}^{m}\sum_{v=1}^{m} f_{uv}\,(M_e)_{uv} \tag{8}$$

The overall document similarity based on PTS is obtained as

$$sim_{PTS}(doc_x, doc_y) = \sum_{i=1}^{n} PTS(d_{x(i)}, d_{y(i)}) \tag{9}$$

The above transportation problem can be efficiently solved by interior-point algorithms [3] which have polynomial time complexity.

2. EXPERIMENTAL RESULTS

In the experiments, we use the agglomerative hierarchical clustering (AHC) algorithm as the cluster engine. Three benchmarking datasets with different sizes extracted from ACM SIGMOD Record 1999[1], which is composed of hundreds of documents from past issues of SIGMOD Record, are used for evaluation. The class each record belonging to is given by the "category" tag[2]. The weighted F-measure is used as the evaluation metric. Table 1 gives the results. For SLVM and PTS, either the edit distance approach or the learning method can be employed to acquire element semantics, the corresponding results are given in different columns respectively.

Table 1. F-measure (%) comparison results

Data Set	VSM	C-VSM	SLVM		PTS	
			Edit Distance	Learned Semantics	Edit Distance	Learned Semantics
ACM-8	40.0	36.9	46.0	53.0	43.3	**58.9**
ACM-12	21.2	22.4	24.2	43.0	22.9	**46.6**
ACM-16	42.4	47.7	38.8	58.8	35.1	**69.9**

Seen from the above table, the proposed PTS with learned element semantics performs best over all three data sets. The approaches with the learned element semantics, i.e. SLVM and PTS, both significantly outperform the traditional approaches not considering the element semantics, i.e. VSM and C-VSM, which shows the importance of incorporating the element semantics for XML document clustering. For SLVM and PTS, we find that the performance based on the learning method for acquiring the element semantics is significantly better than that based on the edit distance approach, which demonstrates that the learned element semantics can reflect the true underlying semantics between XML elements.

The experimental results demonstrate that with the appropriate element semantics, PTS can improve the performance by circumventing both the "over-contribution" problem and the "under-contribution" problem.

3. REFERENCES

[1] A. Doucet, H. A. Myka. Naive Clustering of a Large XML Document Collection. In Proceedings of the 1st INEX, Germany, 2002.

[2] P. Giannopoulos and R. C. Veltkamp. A Pseudo-Metric for Weighted Point Sets. In *Proceedings of the 7th European Conference on Computer Vision (ECCV)* , 715–730, 2002.

[3] N. Karmarkar. A new polynomial-time algorithm for linear programming. In Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing , 302-311, 1984.

[4] J.W. Yang and X.O. Chen. A semi-structured document model for text mining. *Journal of Computer Science and Technology*, 17(5): 603-610, 2002.

[5] K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput., 18(6):1245–1262, 1989.

[1] http://www.acm.org/sigs/sigmod/record/xml/XMLSigmodRecordMar1999.zip

[2] All the "category" tags as well as the inner texts and the corresponding descriptions in the record are removed to make the answer class blind to the clustering algorithm.