International Journal on Artificial Intelligence Tools
Vol. 17, No. 5 (2008) 1003–1022

World Scientific
www.worldscientific.com

STRUCTURAL CLASSIFICATION OF XML DOCUMENTS USING MULTISETS

SWAMI IYER

Department of Computer Science, Universisty of Massachusetts at Boston, Boston, Massachusetts 02125, USA
swamir@cs.umb.edu

DAN A. SIMOVICI

Department of Computer Science, Universisty of Massachusetts at Boston, Boston, Massachusetts 02125, USA
dsim@cs.umb.edu

In this paper, we investigate the problem of clustering XML documents based on their structure. We represent the paths in an XML document as a multiset and use the symmetric difference operation on multisets to define certain metrics. These metrics are then used to obtain a measure of similarity between any two documents in a collection. Our technique was successfully applied to real and synthesized XML documents yielding high-quality clusterings.

Keywords: XML; clustering.

1. Introduction

In the recent years, the Extended Markup Language (XML), due to its simple and flexible text format, has been playing an increasingly vital role in the exchange of a wide variety of data on the web and elsewhere. However, with this proliferation of disparate XML sources, there has also been a growing need for the organization of the documents produced by these sources according to their structural traits — a process referred to as *clustering* in the data mining literature. Clustering methods use the distances that estimate the similarity between document structures in terms of the hierarchical relationships of their nodes. Most of the XML documents found on the web, especially when they have been created from legacy HTML, do not have an associated Document Type Descriptor (DTD). Hence the XML document classifier has to rely on the structure of the *instance* document alone.

Clustering XML documents is useful for several reasons. Once a given set of XML documents has been classified into groups containing structurally related documents, a DTD inference engine can assign a DTD to each group individually rather than assigning one to the entire set of documents. Formulation and optimization of queries on homogeneous XML data repositories is much easier and efficient than

on repositories with structurally unrelated documents. Clustering XML documents also helps solving the problem of recognizing different data sources that provide the same kind of information.

Various techniques have been proposed for clustering XML documents based on their structure. Long, Schwartz, and Soecklin[1] view XML documents as trees, and recursively compute the overall distance between two XML trees from the root nodes to leaf nodes. They model the problem of computing the minimum distance between two sets of elements as the worker-to-task-assignment problem, and use the Munkres' (aka Hungarian) algorithm to compute the minimum cost.

Liu, Wang, Hsu, and Herbert[3] use principal component analysis (PCA) to cluster documents with the same DTD. They extract features from documents, modeled by ordered labeled trees, and transform the documents to vectors in a high-dimensional Euclidean space based on the occurrences of the features in the documents. They then reduce the dimensionality of the vectors by principal component analysis (PCA) and cluster the vectors in the reduced dimensional space.

Flesca, Manco, Masciari, Pontieri, and Pugliese[4] represent the structure of an XML document a time series. By analyzing the coefficients of the corresponding Fourier transform it is possible to evaluate the degree of similarity between documents.

Another approach for evaluating structural dissimilarities between two trees introduced by Chawathe, Rajaraman, Garcia-Molina, and Widom[5] consists of finding a "minimum-cost edit script" that transforms one data tree into another. A variant of this approach is considered by Nierman and Jagadish,[6] which introduces a metric based on an "XML-aware" edit distance between ordered labeled trees. Costa, Manco, Ortale, and Tagarelli[7] propose algorithms that accomplish clustering by comparing cluster representatives, which are XML documents subsuming the most typical structural specifics of a set of XML documents.

Algorithms that calculate the tree edit distances between XML documents by considering the structural summaries of the documents instead of the actual documents thus minimizing the processing requirements, are discussed by Dalamagas, Cheng, Winkel, and Sellis.[2]

Lee, Yang, Hsu, and Yang[8] introduce XClust, an integration strategy that involves the clustering of DTDs. A matching algorithm based on the semantics, immediate descendents and leaf-context similarity of DTD elements is developed.

Jianwu and Xiaoou[9] present a structured link vector (SLVM) to take advantage of the structure and link information in a semi-structured XML document for better mining. They represent a document as a vector and the vectors' elements are determined by terms, document structure, and neighboring documents. Text mining based on SLVM is described using K-means.

Yoon, Raghavan, and Chakilam[10] describe a new bitmap indexing based technique to cluster XML documents. They define the similarity and popularity operations available in bitmap indexes and propose a method for partitioning a XML document set.

Bertino, Guerinni, Mesiti, Rivara, and Tavella[11] propose a metric for quantifying the structural similarity between an XML document and a DTD. This metric is then employed for document classification and clustering. In the first case, the proposed metric is used for selecting a DTD from the ones in the source, whose structure is the most similar to that of the document. In the second case, the aim is to group the documents according to their structural similarities.

Mesiti, Rosso, and Merlo[12] propose a Bayesian approach to Word Sense Disambiguation (WSD) for the retrieval of XML documents.

Tagarelli and Greco[13] address the problem of clustering XML documents according to structure as well as content features. They propose a framework for clustering semantically cohesive XML structures based on a transactional representation model.

In this paper, we propose a novel and efficient approach to the problem of clustering XML documents. We model an XML document as a labeled rooted tree and represent the rooted labeled paths — a sequence of nodes of the tree starting with the root of the tree and ending with a leaf node — of the tree as a multiset, which is a function mapping each path to its multiplicity, i.e., the number of occurrences of the path within the tree. We extend the notion of symmetric difference of sets to that of multisets, and we define metrics on multisets based on their symmetric difference. Thus, given a set of XML documents, we can compute their pairwise distance measures by first building a multiset representation for each of the document, and by computing the distance measures between the multisets using the distance metrics we have introduced. Once we have the distance matrix for the set of documents, we can use one of the standard hierarchical clustering algorithms to cluster the documents. Our approach is efficient; the time taken to build the multiset representation for a document is $O(k|V|)$, where k is the maximum level of nesting in the document, and $|V|$ is the number of elements in the document; the time taken to compute the distance measure between two XML documents with multiset representations M and P is $O(\mathsf{size}(M) + \mathsf{size}(P))$, where $\mathsf{size}(Q)$ is the number of unique element names in the document with multiset representation Q. Our solution works not only with XML documents that belong to strictly different — differing at the root level onwards — classes, but also with documents that differ only at levels that are farther away from the root.

The rest of this paper is organized as follows. Section 2 introduces the notion of a multiset, defines various set-theoretic operations on multisets, and based on these operations, defines a set of metrics on multisets. Section 3 describes how the paths of a labeled rooted tree can be represented as a multiset. Section 4 defines measures of dissimilarity between labeled rooted trees given their multiset representations. Section 5 provides the algorithms for building a multiset for a labeled rooted tree, and for computing the distance measures between any two such trees. Section 6 presents experimental results from running one of the popular hierarchical clustering algorithms on real and synthesized data using the distance measures we have introduced, and, finally, Section 7 concludes our work.

2. Multisets

Multisets[15] are generalizations of sets that capture the multiplicity of elements.

Definition 2.1. A *multiset* on a set X is a function $M : X \longrightarrow \mathbb{N}$. The number $M(x)$ is the multiplicity of x in M. If $M(x) > 0$ we say that x is an element of M.

The set of multisets on a set X is denoted by $\mathcal{M}(X)$. The *empty multiset* on X is the multiset $\emptyset_X$ defined by $\emptyset_X(x) = 0$ for every $x \in X$.

The *support set* of the multiset M is the set

$$\mathsf{sset}(M) = \{x \in X | M(x) > 0\}. \tag{1}$$

If the set $\mathsf{sset}(M)$ is finite, then we say that *the multiset M is finite*. The cardinality of the set $\mathsf{sset}(M)$ is the *size* of the multiset M, denoted by $\mathsf{size}(M)$.

We will denote a finite multiset M on X as a formal sum

$$M = m_1 x_1 + \cdots + m_k x_k, \tag{2}$$

where $x_1, \ldots, x_k$ are the distinct members of the set $\mathsf{sset}(M)$ and $M(x_i) = m_i$.

The union, intersection and symmetric difference of two multisets are defined such that they generalize the usual set-theoretic operations. Let M, P be two multisets on a set X. The *union of M and P* is the multiset $M \cup P$ such that $(M \cup P)(x) = \max\{M(x), P(x)\}$; the *intersection of M and P* is the multiset $M \cap P$ given by $(M \cap P)(x) = \min\{M(x), P(x)\}$ for $x \in X$.

We define two extensions of the symmetric difference.

Definition 2.2. The *weak symmetric difference* of the multisets M and P on the set X is the multiset $M \oplus P$ on X defined by $(M \oplus P)(x) = |M(x) - P(x)|$ for every $x \in X$.

Note that unlike the usual symmetric difference of sets, this is not an associative operation because, in general $||a - b| - c| \neq |a - |b - c||$ (e.g., $||7 - 5| - 3| = 1$, while $|7 - |5 - 3|| = 5$). We have

$$\mathsf{sset}(M) \cup \mathsf{sset}(P) \subseteq \mathsf{sset}(M \oplus P) \tag{3}$$

for every multiset M, P. This inclusion may be strict if M and P have at least one common element with distinct multiplicities.

The *strong symmetric difference* $M \boxplus P$ of multisets that we define next preserves more properties of set difference. Let $\phi : \mathbb{N}^2 \longrightarrow \mathbb{N}$ be the function defined by $\phi(m, p) = 0$ if $m = p = 0$ or $m > 0$ and $p > 0, \phi(m, p) = \max\{m, p\}$ if exactly one of m, p is positive, for $m, p \in \mathbb{N}$.

Definition 2.3. The *strong symmetric difference* of the multisets M and P on the set X is the multiset $M \boxplus P$ on X defined by $(M \boxplus P)(x) = \phi(M(x), P(x))$ for every $x \in X$.

Observe that $\phi(\phi(m,p),q) > 0$ if and only if $\phi(m,\phi(p,q)) > 0$, as it can be easily verified by considering all possible cases of nullity of m, p and q. Thus, $\mathsf{sset}((M \boxplus P) \boxplus Q) = \mathsf{sset}(M \boxplus (P \boxplus Q))$ for every multiset M, P, Q, which extends the associative property of set difference.

The distributivity of set intersection with respect to symmetric difference of sets is preserved by the strong symmetric difference of multisets, as shown next.

Theorem 2.1. Let $M, P, Q \in \mathcal{M}(X)$ be three multisets on a set X. We have

$$M \cap (P \boxplus Q) = (M \cap P) \boxplus (M \cap Q) \tag{4}$$

Proof. Let $m = M(x)$, $p = P(x)$, and $q = Q(x)$. The statement follows by analyzing the eight cases, that occur depending whether each of these numbers is 0 or greater than 0. □

In general, we have $M \boxplus P \leq M \oplus P$ for every multisets M, P.

Next, we use the weak and strong symmetric difference of two multisets $M, P \in \mathcal{M}(X)$ to define a metric on $\mathcal{M}(X)$, where X is a finite set.

Theorem 2.2. Let X be a finite set. The mapping $\delta_{\oplus} : \mathcal{M}(X)^2 \longrightarrow \mathbb{R}_{\geq 0}$ given by

$$\delta_{\oplus}(M,P) = \sum_{x \in X} \frac{(M \oplus P)(x)}{(M \cup P)(x)} \tag{5}$$

where $M, P \in \mathcal{M}(X)$ are multisets on X, is a metric on $\mathcal{M}(X)$.

Proof. Let M, P be two finite multisets on a finite set X. If $|X| = n$, we can define a metric on $\mathcal{M}(X)$ using the Minkowski metric on $\mathbb{R}^n$ as

$$\delta_k(M,P) = \left(\sum_{x \in X} |M(x) - P(x)|^k \right)^{\frac{1}{k}} \tag{6}$$

where $M, P \in \mathcal{M}(X)$ are multisets on X and $k \geq 1$. In particular, for $k = 1$ we have the metric

$$\delta_1(M,P) = \sum_{x \in X} |M(x) - P(x)| = \sum_{x \in X} (M \oplus P)(x). \tag{7}$$

It is easy to see that for any choice of a weighting function $w : X \longrightarrow \mathbb{R}_{\geq 0}$ the following is a metric on the the set of multisets:

$$\begin{aligned} \delta_{\oplus}(M,P) &= \sum_{x \in X} w(x)|M(x) - P(x)| \\ &= \sum_{x \in X} w(x)(M \oplus P)(x). \end{aligned} \tag{8}$$

Thus,

$$\delta_{\oplus}(M,P) = \sum_{x \in X} \frac{(M \oplus P)(x)}{(M \cup P)(x)}, \tag{9}$$

where $w(x) = \frac{1}{(M \cup P)(x)}$, is a metric. □

Lemma 2.1. *Let X be a set and let M, P be two multisets on X. Define $\Psi_{MP}(x)$ as*

$$\Psi_{MP}(x) = \begin{cases} 0 & \text{if } M(x) = P(x) = 0 \\ \frac{(M \uplus P)(x)}{(M \cup P)(x)} & \text{otherwise,} \end{cases} \tag{10}$$

for $x \in X$. We have $\Psi_{MP}(x) \in \{0,1\}$ for every $x \in X$ and

$$\Psi_{MP}(x) \leq \Psi_{MQ}(x) + \Psi_{QP}(x) \tag{11}$$

for every $M, Q, P \in \mathcal{M}(X)$ and $x \in X$.

Proof. The fact that $\Psi_{MP}(x) \in \{0,1\}$ for every $x \in X$ is immediate.

Note that if $M(x) = P(x) = 0$ the inequality is clearly satisfied. This is also the case if we have both $M(x) > 0$ and $P(x) > 0$ because in this case $\phi(M(x), P(x)) = 0$.

Suppose, therefore that exactly one of the numbers $M(x)$ or $P(x)$, say $M(x)$, is nonnegative, so $\Psi_{MP} = 1$. We have two cases to consider:

Case 1: $Q(x) > 0$. In this case, $\Psi_{MQ}(x) = 0$ and $\Psi_{QP}(x) = 1$, which means that the inequality of the lemma is satisfied.

Case 2: $Q(x) = 0$. In this case, $\Psi_{MQ}(x) = 1$ and $\Psi_{QP}(x) = 0$, which means again that the same inequality is satisfied. □

Theorem 2.3. The mapping $\delta_{\uplus} : \mathcal{M}(X)^2 \longrightarrow \mathbb{R}_{\geq 0}$ given by

$$\delta_{\uplus}(M,P) = \sum_{x \in X} \Psi_{MP}(x) \tag{12}$$

for $M, P \in \mathcal{M}(X)$ is a semi-metric on $\mathcal{M}(X)$.

Proof. This is an immediate consequence of Lemma 2.1 □

3. The Multiset of Paths of a Labeled Rooted Tree

A *tree* is a connected acyclic graph $\mathcal{T} = (V, E)$; a *rooted tree* is a pair $(\mathcal{T}, v_0)$, where v_0 is a vertex called the *root*.

A *labeled rooted tree* is a 4-tuple $(\mathcal{T}, v_0, l, L)$, where $(\mathcal{T}, v_0)$ is a rooted tree, $l : V \longrightarrow L$ is a function, and L is a set whose elements are referred to as *labels*; $l(v)$ is the *label of the vertex v*.

The set of finite sequences of elements of a set E is denoted by $\mathbf{seq}(E)$. A *rooted labeled path* in a labeled rooted tree $(\mathcal{T}, v_0, l, L)$ is a sequence of labels $\mathbf{l} =$

$(a_0, a_1, \ldots, a_n) \in \mathbf{seq}(L)$ such that there exists a path $(v_0, v_1, \ldots, v_n)$ in $\mathcal{T}$ and $l(v_i) = a_i$ for $0 \leq i \leq n$. Clearly, for each vertex v of the rooted tree $(\mathcal{T}, v_0)$ there exists a unique path that starts with v_0 and ends with v and for any such vertex there is a rooted labeled path that ends with $l(v)$.

Unlike the usual practice in graph theory, we define the length of the rooted path $\mathbf{l} = (a_0, a_1, \ldots, a_n)$ simply as the length $n + 1$ of the sequence and denote it as $\ell(\mathbf{l})$.

For a multiset $M = m_1 p_1 + \cdots + m_k p_k$ of sequences of elements of L and a sequence r we define the multiset rM as

$$rM = m_1 r p_1 + \cdots + m_k r p_k, \tag{13}$$

where rp_i is the sequence obtained by concatenating r and p_i.

The multiset of rooted labeled paths of a labeled rooted tree $(\mathcal{T}, v_0, l, L)$, denoted by $\mathsf{RLP}(\mathcal{T}, v_0, l, L)$, is a multiset of sequences of labels. This set can be defined recursively as follows:

(1) If $\mathcal{T} = (\{v_0\}, \emptyset)$ and $l(v_0) = a$, then $\mathsf{RLP}(\mathcal{T}, v_0, l, L) = 1(a)$.
(2) Suppose that the immediate descendants of v_0 in $\mathcal{T}$ are $v_1, \ldots, v_m$, and the subtrees of $\mathcal{T}$ having the roots in $v_1, \ldots, v_m$ are $\mathcal{T}_1, \ldots, \mathcal{T}_m$, respectively. Then,

$$\mathsf{RLP}(\mathcal{T}, v_0, l, L) = 1(a) + \sum_{i=1}^{m} a\mathsf{RLP}(\mathcal{T}_i, v_i, l, L). \tag{14}$$

Example 3.1. Consider the labeled rooted trees shown in Figure 1. Their respective multisets of rooted labeled paths are given by

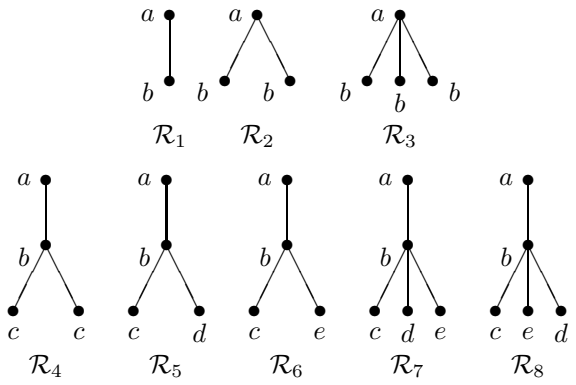

Fig. 1. Examples of labeled rooted trees.

A characterization of multisets of paths of labeled rooted trees is given next. Recall that a sequence $\mathbf{u}$ is a prefix of a sequence $\mathbf{v}$ if $\mathbf{v}$ can be written as $\mathbf{v} = \mathbf{uw}$ for some sequence $\mathbf{w}$. Further, $\mathbf{u}$ is a *proper prefix* of $\mathbf{v}$ if $\mathbf{u}$ is a prefix of $\mathbf{v}$ and $\mathbf{u} \neq \mathbf{v}$.

Table 1. Multisets of the rooted labeled paths for the trees in Fig. 1.

Tree	Multiset of Rooted Labeled Paths
$\mathcal{R}_1$	$1(a) + 1(a, b)$
$\mathcal{R}_2$	$1(a) + 2(a, b)$
$\mathcal{R}_3$	$1(a) + 3(a, b)$
$\mathcal{R}_4$	$1(a) + 1(a, b) + 2(a, b, c)$
$\mathcal{R}_5$	$1(a) + 1(a, b) + 1(a, b, c) + 1(a, b, d)$
$\mathcal{R}_6$	$1(a) + 1(a, b) + 1(a, b, c) + 1(a, b, e)$
$\mathcal{R}_7$	$1(a) + 1(a, b) + 1(a, b, c) + 1(a, b, d) + 1(a, b, e)$
$\mathcal{R}_8$	$1(a) + 1(a, b) + 1(a, b, c) + 1(a, b, e) + 1(a, b, d)$

Theorem 3.1. Let L be a set. A finite multiset of sequences M over $\mathbf{seq}(L)$ is the multiset of paths of a labeled rooted tree if and only if the following conditions are satisfied:

(1) there exists a sequence (a) with $M((a)) = 1$ that is a proper prefix of every sequence $\mathbf{p}$ such that $M(\mathbf{p}) > 0$;
(2) for every prefix $\mathbf{r}$ of a sequence $\mathbf{p}$ such that $M(\mathbf{p}) > 0$ we have $M(\mathbf{r}) > 0$.

Proof. Suppose that the conditions of the theorem are satisfied. We must show the existence of a labeled rooted tree $\mathcal{R} = (\mathcal{T}, v_0, l, L)$ such that $\mathsf{RLP}(\mathcal{R}) = M$.

The vertices of $\mathcal{T}$ will be indexed by sequences $\mathbf{p} \in \mathbf{seq}(L)$ such that $M(\mathbf{p}) > 0$ and the root of the tree will be the sequence $v_{(a)}$. Note that the first condition implies that the sequence (a) is the unique sequence with this property.

Suppose that $\mathbf{p}, \mathbf{q}$ are two distinct sequences in $\mathbf{seq}(L)$ such that $\mathbf{p}$ is a prefix of $\mathbf{q}$ and $M(\mathbf{q}) > 0$. If $\mathbf{p}$ is a sequence of maximal length having these properties, then $\mathbf{q} = \mathbf{p}a$ for some $a \in L$. Indeed, if this is not the case, then there exists a sequence $\mathbf{r}$ such that $\mathbf{r}$ is a prefix of $\mathbf{q}$ and $\mathbf{p}$ is a prefix of $\mathbf{r}$ and we have both $\mathbf{r} \neq \mathbf{q}$ and $\mathbf{r} \neq \mathbf{p}$. By the second condition of the theorem we have $M(\mathbf{r}) > 0$; since $|\mathbf{r}| > |\mathbf{p}|$ this contradicts the maximality of the length of $\mathbf{p}$. We will consider a pair $(v_{\mathbf{p}}, v_{\mathbf{q}})$ as an edge in $\mathcal{T}$ and all edges of this graph will have this form. Note that this argument implies that for every vertex $v_{\mathbf{q}}$ there is a unique path that begins with $v_{(a)}$ and ends with $v_{\mathbf{q}}$. Thus, $\mathcal{T}$ is indeed a tree. The function l is given by $l(v_{\mathbf{q}}) = a$, where a is the last symbol of the sequence $\mathbf{q}$. This completes the definition of $\mathcal{R}$.

We need to verify now that $\mathsf{RLP}(\mathcal{R}) = M$. The argument is by induction on the number n of vertices of the underlying tree of $\mathcal{R}$.

The basis step, $n = 1$, is immediate. Suppose that the equality holds for trees with fewer than n vertices and let $\mathcal{T}$ be the underlying tree of $\mathcal{R}$. Let $\mathcal{T}_1, \ldots, \mathcal{T}_m$ be the immediate subtrees of $\mathcal{R}$ and let $\mathcal{R}_1, \ldots, \mathcal{R}_m$ be the corresponding labeled rooted trees, $\mathcal{R}_i = (\mathcal{T}_i, v_{(a_i)}, l_i, L)$. Let K_i be the multiset of labeled rooted paths of $\mathcal{R}_i$. If we construct the rooted labeled tree for K_i as we did above for M, then $\mathcal{R}_{K_i}$

coincides with $\mathcal{R}_i$. Thus, by inductive hypothesis, $\mathsf{RLP}(\mathcal{R}_i) = \mathsf{RLP}(\mathcal{R}_{K_i}) = K_i$. Since $M = (a) + \sum_{i=1}^{m}(a)K_i = (a) + \sum_{i=1}^{m}(a)\mathsf{RLP}(\mathcal{R}_i)$, we have $M = \mathsf{RLP}(\mathcal{R})$.

Necessity of the conditions can be easily shown and we omit the argument. □

4. Metric Space of Labeled Rooted Trees

We introduce a dissimilarity measure between labeled rooted trees having a set of labels L using the multisets of rooted labeled paths and a metric defined on the class of these multisets that uses a weight function.

Definition 4.1. Let $\mathcal{R} = \{\mathcal{T}, v_0, l, L\}$ and $\mathcal{R}' = \{\mathcal{T}', v_0', l', L\}$ be two rooted labeled trees.

The weak dissimilarity between $\mathcal{R}$ and $\mathcal{R}'$ is the number

$$d_{\oplus}(\mathcal{R}, \mathcal{R}') = \sum_{k\in\mathbb{N}} 2^{-(k+1)}|L|^{-k} \cdot \sum \frac{(\mathsf{RLP}(\mathcal{R}) \oplus \mathsf{RLP}(\mathcal{R}'))(p)}{(\mathsf{RLP}(\mathcal{R}) \cup \mathsf{RLP}(\mathcal{R}'))(p)}, \tag{15}$$

and the strong dissimilarity between $\mathcal{R}$ and $\mathcal{R}'$ is defined by

$$d_{\boxplus}(\mathcal{R}, \mathcal{R}') = \sum_{k\in\mathbb{N}} 2^{-(k+1)}|L|^{-k} \cdot \sum \Psi_{\mathsf{RLP}(\mathcal{R}),\mathsf{RLP}(\mathcal{R}')}(p) \tag{16}$$

where $p \in \mathbf{seq}(L)$ and $\ell(p) = k$.

It is clear that the weak and strong dissimilarity measures are semi-metrics on the class of labeled rooted trees. In other words, $d_\star(\mathcal{R}, \mathcal{R}) = 0$, $d_\star(\mathcal{R}, \mathcal{R}') = d_\star(\mathcal{R}', \mathcal{R})$, and $d_\star(\mathcal{R}, \mathcal{R}'') \leq d_\star(\mathcal{R}, \mathcal{R}') + d_\star(\mathcal{R}', \mathcal{R}'')$ for every $\mathcal{R}, \mathcal{R}', \mathcal{R}''$, where $\star$ is the $\oplus$ or $\boxplus$ operation. However, if $d_\star(\mathcal{R}, \mathcal{R}') = 0$ then $\mathcal{R}, \mathcal{R}'$ can differ relative to the order of descendants of a vertex. Note that $d_\star(\mathcal{R}, \mathcal{R}') \in [0, 1]$.

Example 4.1. The weak and strong dissimilarity measures between the labeled rooted trees $\mathcal{R}_1, \ldots, \mathcal{R}_8$ displayed in Fig. 1 are shown below.

A well-formed XML document, disregarding the IDREFS, can be represented as a labeled rooted tree, with the document element forming the root of the tree, and its sub-elements forming the other vertices. Attributes of an element can be treated as being the element's children, and hence also as vertices of the tree. Consider the following XML document:

```
<books>
  <book year="1910">
    <title>Principia Mathematica</title>
    <author>Alfred North Whitehead</author>
    <author>Bertrand Russell</author>
    <publisher>Cambridge University Press</publisher>
  </book>
</books>
```

The above XML document can be modeled as the labeled rooted tree $(\mathcal{T}, v_0, l, L)$, where $(\mathcal{T}, v_0)$ is a rooted tree, $\mathcal{T} = (V, E)$ is a connected acyclic graph with the set of vertices $V = \{v_0, v_1, v_2, v_3, v_4, v_5, v_6\}$ and the set of edges $E = \{(v_0, v_1), (v_1, v_2), (v_1, v_3), (v_1, v_4), (v_1, v_5), (v_1, v_6)\}$, and $l : V \longrightarrow L$ is a function such that $l(v_0) = books, l(v_1) = book, l(v_2) = year, l(v_3) = title, l(v_4) = l(v_5) = author$ and $l(v_6) = publisher$. Note how we pay no attention to the content of an XML document; we are concerned here only with the document's structure.

Table 2. Weak and strong dissimilarity measures.

$d_{\oplus}$	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$	$\mathcal{R}_5$	$\mathcal{R}_6$	$\mathcal{R}_7$	$\mathcal{R}_8$
$\mathcal{R}_1$	0	0.12	0.17	0.12	0.12	0.12	0.12	0.12
$\mathcal{R}_2$	0.12	0	0.08	0.25	0.25	0.25	0.25	0.25
$\mathcal{R}_3$	0.17	0.08	0	0.29	0.29	0.29	0.29	0.29
$\mathcal{R}_4$	0.12	0.25	0.29	0	0.09	0.09	0.10	0.10
$\mathcal{R}_5$	0.12	0.25	0.29	0.09	0	0.08	0.04	0.04
$\mathcal{R}_6$	0.12	0.25	0.29	0.09	0.08	0	0.04	0.04
$\mathcal{R}_7$	0.12	0.25	0.29	0.10	0.04	0.04	0	0
$\mathcal{R}_8$	0.12	0.25	0.29	0.10	0.04	0.04	0	0
$d_{\boxplus}$	$\mathcal{R}_1$	$\mathcal{R}_2$	$\mathcal{R}_3$	$\mathcal{R}_4$	$\mathcal{R}_5$	$\mathcal{R}_6$	$\mathcal{R}_7$	$\mathcal{R}_8$
$\mathcal{R}_1$	0	0	0	0.12	0.12	0.12	0.12	0.12
$\mathcal{R}_2$	0	0	0	0.12	0.12	0.12	0.12	0.12
$\mathcal{R}_3$	0	0	0	0.12	0.12	0.12	0.12	0.12
$\mathcal{R}_4$	0.12	0.12	0.12	0	0.06	0.06	0.08	0.08
$\mathcal{R}_5$	0.12	0.12	0.12	0.06	0	0.08	0.04	0.04
$\mathcal{R}_6$	0.12	0.12	0.12	0.06	0.08	0	0.04	0.04
$\mathcal{R}_7$	0.12	0.12	0.12	0.08	0.04	0.04	0	0
$\mathcal{R}_8$	0.12	0.12	0.12	0.08	0.04	0.04	0	0

The multiset of rooted labeled paths for the labeled rooted tree representing an XML document can be constructed using the `buildMultiset` procedure discussed in Section 5. The multiset for the above mentioned document can be expressed as the formal sum $1(books) + 1(books, book) + 1(books, book, year) + 1(books, book, title) + 2(books, book, author) + 1(books, book, publisher)$. Once we have constructed the multisets for any two XML documents, we can use the `computeWeakDistance` and `computeStrongDistance` procedures, also discussed in Section 5, to compute a measure of dissimilarity between the documents.

5. Algorithms

We first present the algorithm for building the multiset for a labeled rooted tree. The algorithm does a depth-first traversal of the tree in order to build the multiset.

Algorithm 1: buildMultiset

```
input  : Labeled rooted tree R = {T, v0, l, L}
output: The multiset for R
Multiset multiset
Stack nodeStack, sequenceStack
Sequence s
s.add(l(v0))
sequenceStack.push(s)
multiset.addSequence(s)
nodeStack.push(v0)
while nodeStack is not empty do
    Node topNode ⇐ nodeStack.peek()
    Node unvisitedChild ⇐ unvisited child of topNode
    if unvisitedChild is not null then
        Sequence topSequence ⇐ sequenceStack.peek()
        Sequence newTopSequence ⇐ topSequence.copy()
        newTopSequence.add(l(unvisitedChild))
        sequenceStack.push(newTopSequence)
        multiset.addSequence(newTopSequence)
        nodeStack.push(unvisitedChild)
    else
        sequenceStack.pop()
        nodeStack.pop()
return multiset
```

The above algorithm runs in $O(k|V|)$ time, where k is the length of the longest rooted labeled path in $\mathcal{R}$, and $|V|$ is the number of vertices in $\mathcal{T}$.

The algorithms for computing the weak and strong dissimilarity measures between two labeled rooted trees given their multiset representations are discussed next.

Algorithm 2: computeWeakDistance

input : Multisets M, P
output: The weak distance $d_{\oplus}(M, P)$
$distance \Leftarrow 0.0$
$Q \Leftarrow \mathsf{sset}(M) \cup \mathsf{sset}(P)$
Map $elementCount$
foreach $q \in Q$ **do**
  $count \Leftarrow elementCount.get(\ell(q))$
  if *count is null* **then**
    $elementCount.put(\ell(q), 1)$
  else
    $elementCount.put(\ell(q), count + 1)$
foreach $q \in Q$ **do**
  $m \Leftarrow M.multiplicity(q)$
  $p \Leftarrow P.multiplicity(q)$
  $d \Leftarrow \frac{|m-p|}{\max\{m,p\}}$
  $distance \Leftarrow distance + \frac{d}{elementCount.get(\ell(q)) \times 2^{(\ell(q)+1)}}$
return *distance*

Algorithm 3: computeStrongDistance

input : Multisets M, P
output: The strong distance $d_{\boxplus}(M, P)$
$distance \Leftarrow 0.0$
$Q \Leftarrow \mathsf{sset}(M) \cup \mathsf{sset}(P)$
Map $elementCount$
foreach $q \in Q$ **do**
  $count \Leftarrow elementCount.get(\ell(q))$
  if *count is null* **then**
    $elementCount.put(\ell(q), 1)$
  else
    $elementCount.put(\ell(q), count + 1)$
foreach $q \in Q$ **do**
  $m \Leftarrow M.multiplicity(q)$
  $p \Leftarrow P.multiplicity(q)$
  $\psi \Leftarrow 1$
  if $m = 0$ *and* $p = 0$ *or* $m > 0$ *and* $p > 0$ **then**
    $\psi \Leftarrow 0$
  $distance \Leftarrow distance + \frac{\psi}{elementCount.get(\ell(q)) \times 2^{(\ell(q)+1)}}$
return *distance*

Both `computeWeakDistance` and `computeStrongDistance` algorithms run in $O(\mathsf{size}(M) + \mathsf{size}(P))$ time, where M and P are multisets.

6. Experimental Results

We wrote a program called MUDXML,[16] an acronym of *Multiset Distance for XML*, that implements the algorithms discussed in Section 5. MUDXML processes the XML documents contained in the directory specified as input, computes their pairwise (weak and strong) distance measures, and prints the distance matrix to standard output. We used MUDXML to cluster three data sets. The first, namely "Niagara",[17] comprised of randomly picked XML documents belonging to three distinct classes: department, astronomy, and club. The second, namely "Sigmod",[19] comprised of randomly picked XML documents belonging to three distinct classes: index terms, proceedings, ordinary issue. The third, namely "Synthesized", comprised of XML documents generated from three DTDs [a] by ToXGene.[20] The table below summarizes the contents of the data sets used:

Table 3. Summary of data sets.

Dataset	Class (# of Documents)
Niagara	Department (17) Astronomy (16) Club (12)
Sigmod	Index Terms (16) Proceedings (16) Ordinary Issue (16)
Synthesized	Book Catalog 1 (15) Book Catalog 2 (15) Book Catalog 3 (15)

For each of the above data sets, we computed the distance matrix using our program. In order to cluster the documents based these distance matrices, we used the `hclust` function from the *cluster* package of the statistical computing software "R".[18] We employed the "average" hierarchical clustering algorithm. Figures 2, 3, 4, and 5 show the dendrogram plots for the "Niagara" and "Synthesized" data sets respectively, with the rectangular regions highlighting the clusters. Both weak and strong distance measures perform very well in classifying the "Niagara" data set, since the documents belonging to the different classes in this data set have rooted labeled paths that approximately have the same multiplicities. The weak distance measure does somewhat poorly on the "Synthesized" data set, since the rooted labeled paths in the documents belonging to this data set have a wide range of multiplicities. The strong distance measure does extremely well on this data set, since each rooted labeled path in the documents belonging to the data set has a non-zero multiplicity.

[a]The DTDs, representing catalogs of books, were very similar, differing only at levels farther away from the root

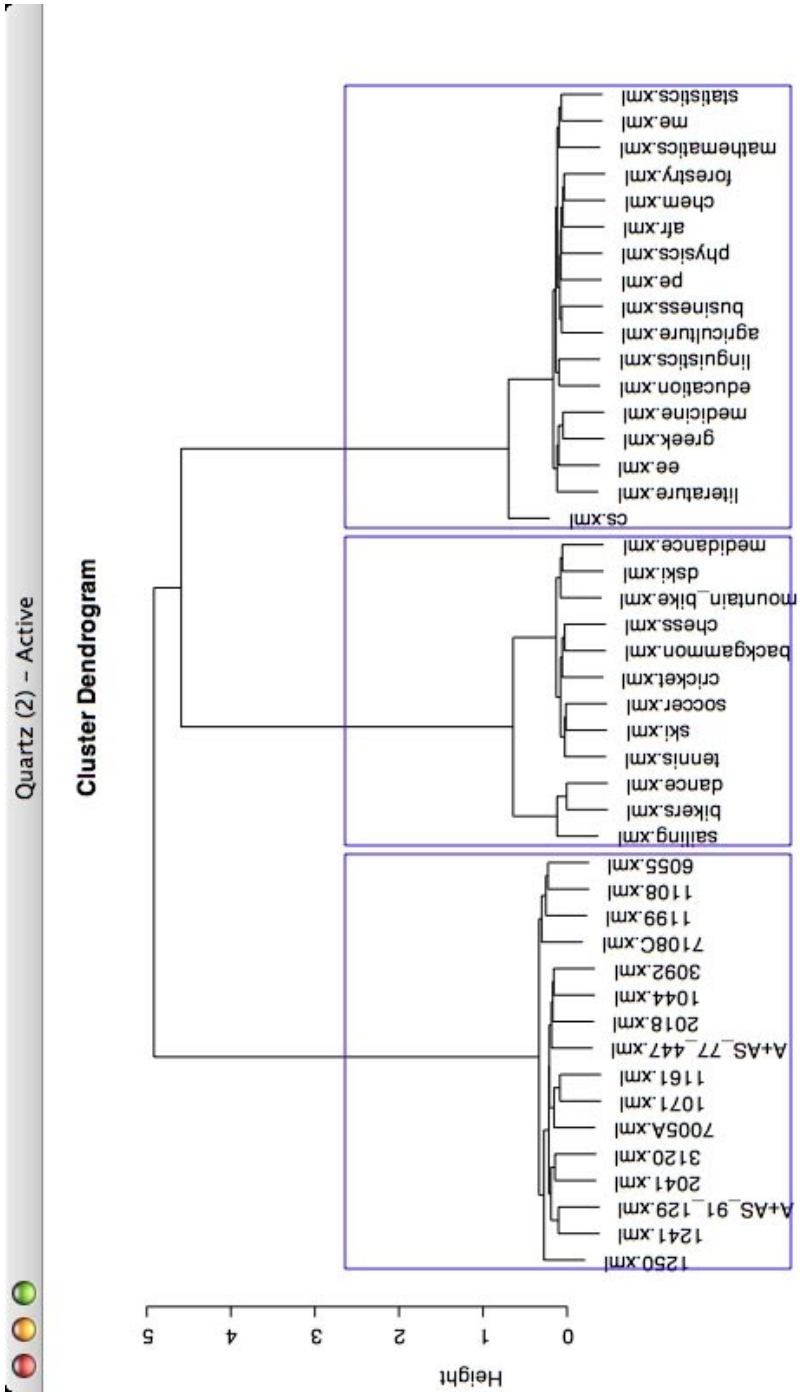

Fig. 2. Niagara data set clusters (weak distance measure).

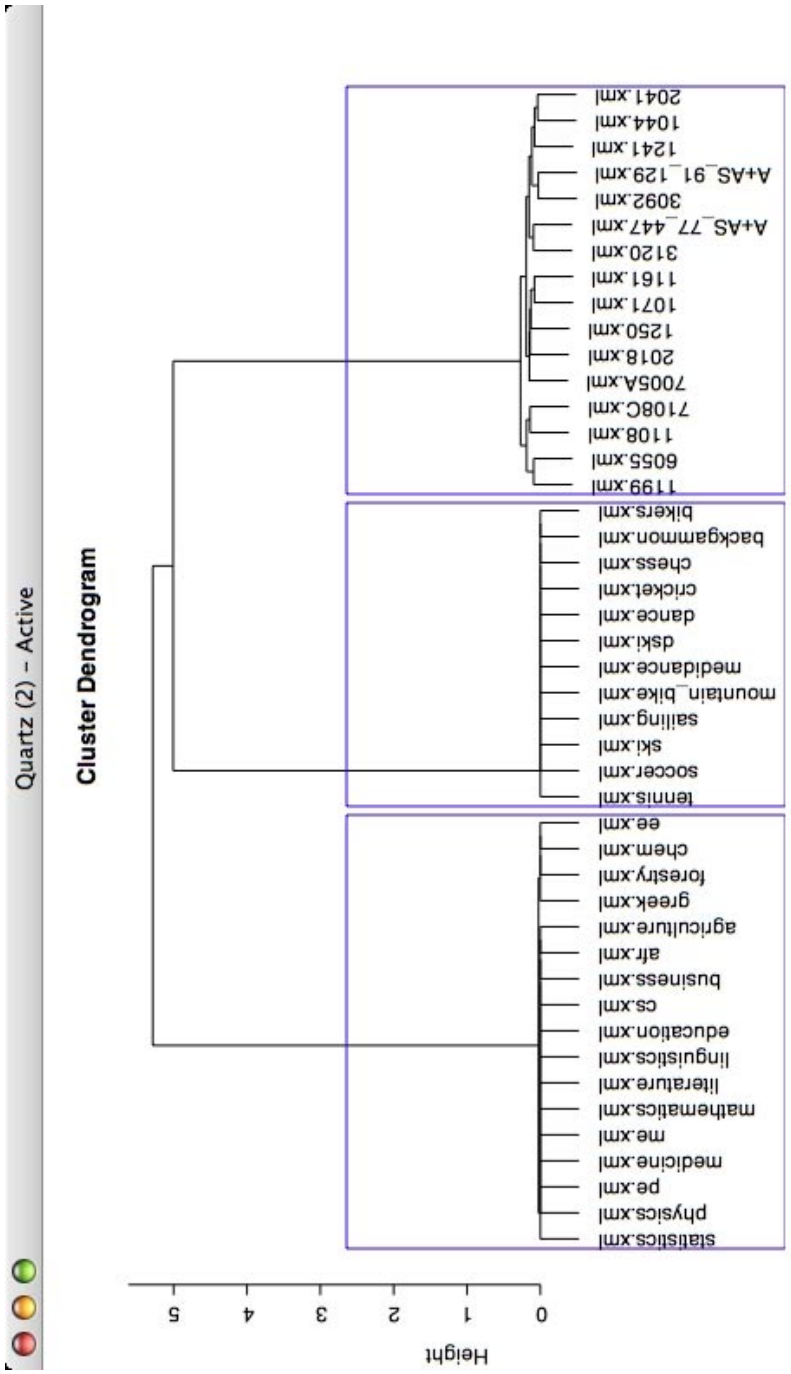

Fig. 3. Niagara data set clusters (strong distance measure).

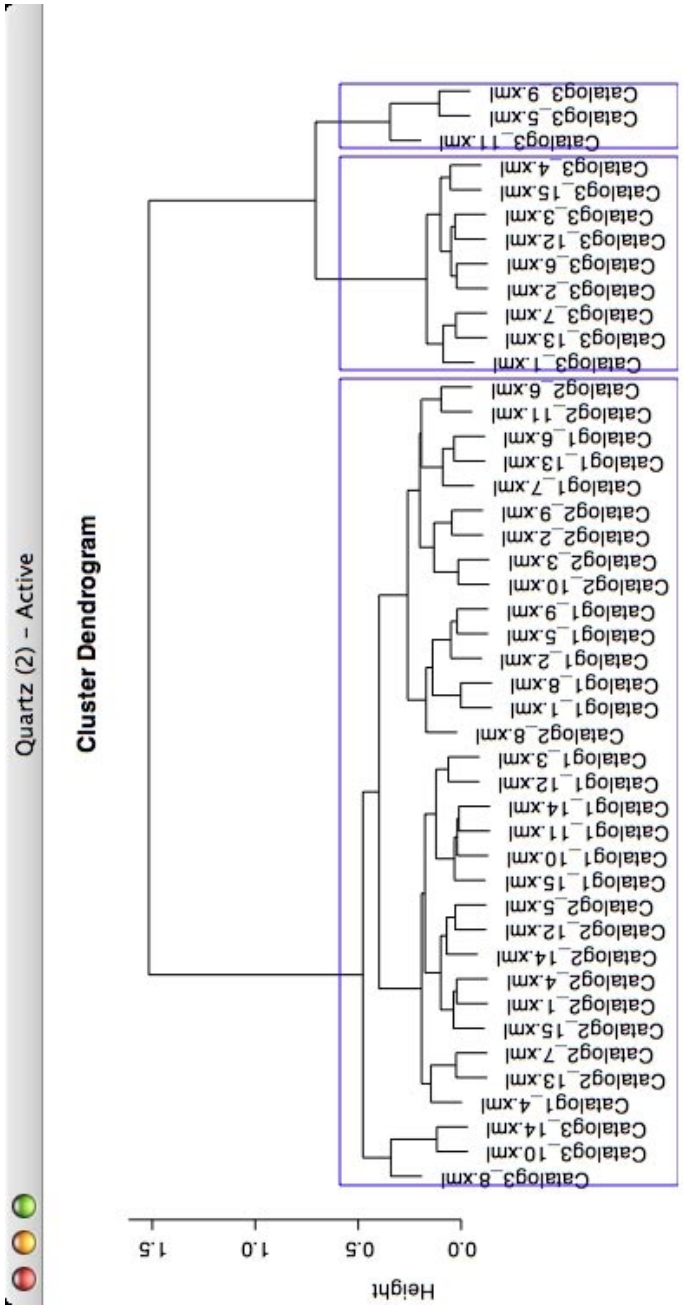

Fig. 4. Synthesized data set clusters (weak distance measure).

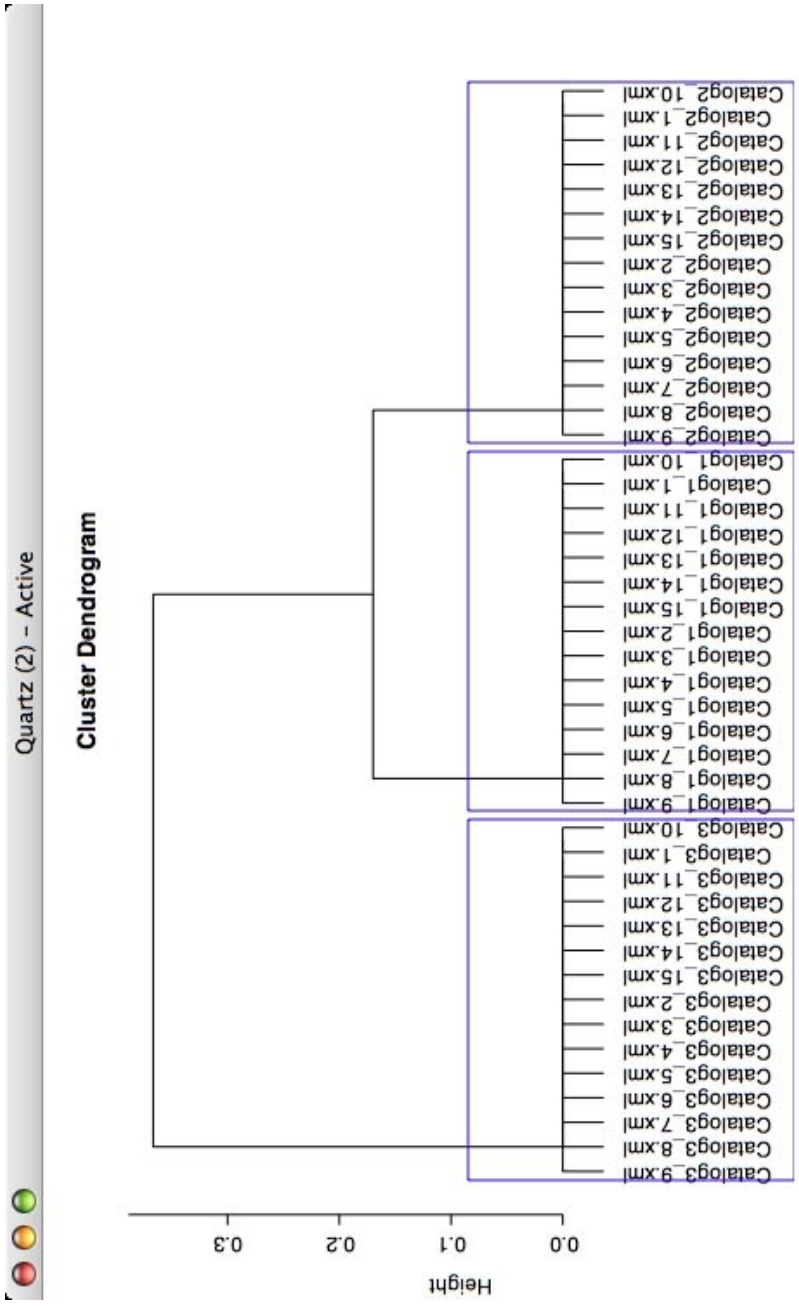

Fig. 5. Synthesized Data set clusters (strong distance measure).

We used the *silhouette*[14] method for evaluating the quality of clusters produced. We again used "R" for this purpose. The following table shows the average silhouette coefficients for the three data sets, along with the value of k[b]:

Table 4. Average silhouette measures for data sets used.

Dataset	Distance	Average sihouette	k
Niagara	Weak	0.94	3
Niagara	Strong	0.98	3
Sigmod	Weak	0.97	3
Sigmod	Strong	0.99	3
Synthesized	Weak	0.51	3
Synthesized	Strong	1.00	3

The documents belonging to the different classes in the "Niagara" and "Sigmod" data sets are have rooted labeled paths that are almost equally represented, i.e., they have approximately equal multiplicities. Hence the weak and strong distance measures, as indicated by the average silhouette coefficients, are very successful in classifying these documents. On the other hand, the documents belonging to the "Synthesized" data set have rooted labeled paths with widely varying multiplicities. Hence the weak distance measure classifies them poorly. Since each rooted labeled path has a non-zero multiplicity, the strong distance measure does perfectly, i.e., has silhouette coefficient equal to 1.

From the dendrogram plots and from the silhouette coefficients for the data sets, we can infer that for XML documents belonging to strictly different classes ("Niagara" and "Sigmod" in our experiment), both the weak and strong distance measures yield "pure" clusters, i.e., the average silhouette coefficient ≈ 1. When the XML documents differ only at levels farther away from the root, if all the rooted labeled paths are equally represented in the documents, i.e., they have approximately equal multiplicities, the weak distance measure would yield "pure" clusters. With such documents, the strong distance measure would yield extremely "pure" clusters as long as the rooted labeled paths have non-zero multiplicities.

7. Conclusions and Future Work

In this work, we presented a novel approach to the problem of clustering XML documents based on their structure. We modeled an XML document as a labeled rooted tree and represented it as a multiset mapping the rooted labeled paths in the tree to the corresponding multiplicities. We defined distance measures on labeled rooted trees based on the symmetric difference of their multisets. We presented

[b]One of the parameters of the `pam` (Partitioning Around Medoids) function in "R" that specifies the number of clusters to look for. The value specified is the one that resulted the maximum silhouette coefficient.

algorithms to build a multiset given a labeled rooted tree, and to compute the weak and strong distance measures given two multisets. We applied these algorithms to both real and synthetic data sets. Clusterings that are formed based on the distances introduced in this paper separate well the documents that are structurally different at various depths.

Our future research in the domain of XML document classification will span two areas, both involving the semantics of the documents. Firstly, although XML documents have proper structures, differently annotated XML documents, owing to subjective definitions of markup tags, may encode related semantics. Classification of such documents requires finding the relatedness of tags. Secondly, XML documents with the same structure could be semantically unrelated. For example, two RSS news feeds that have the same structure could be completely different in terms of their content; one might be on politics while the other might be on medicine. Classifying such documents invariably involves looking at the content of the document.

References

1. J. Long, D. Schwartz, and S. Soecklin. An XML Distance Measure. In *Proceedings of the International Conference on Data Mining (DMIN)*, 2005.
2. T. Dalamagas, T. Cheng, K. Winkel, and T. Sellis. Clustering XML Documents by Structure. In *Proceedings of the Hellenic Conference on Artificial Intelligence (SETN)*, pages 112–121, 2004.
3. J. Liu, J.T.L. Wang, W.Hsu, K.G. Herbert. XML Clustering by Principal Component Analysis. In *Proceedings of International Conference on Tools in Artificial Intelligence*, pages 658–662, 2004.
4. S. Flesca, G. Manco, E. Masciari, L. Pontieri, and A. Pugliese. Detecting Structural Similarities between XML Documents. In *Proceedings of the International Workshop on the Web and Databases (WebDB)*, 2002.
5. S. Chawathe, A. Rajaraman, H. Garcia-Molina, and J. Widom. Change detection in hierarchically structured information. In *In Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 493–504, 1996.
6. A. Nierman and H. Jagadish. Evaluating Structural Similarity in XML Documents. In *Proceedings of the International Workshop on the Web and Databases (WebDB)*, 2002.
7. G. Costa, G. Manco, R. Ortale, and A. Tagarelli. A Tree-Based Approach to Clustering XML Documents by Structure. In *Proceedings of the Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, 2004.
8. M.L. Lee, L.H. Yang, W. Hsu, X. Yang. XClust: clustering XML schemas for effective integration. In *Proceedings of the eleventh international conference on Information and knowledge management (CIKM)*, pages 292–299, 2002.
9. Y. Jianwu and C. Xiaoou. A Semi-Structured Document Model for Text Mining. *Journal of Computer Science and Technology*, pages 603–610, 2002.
10. J. Yoon, V. Raghavan and Venu Chakilam. BitCube: A Three Dimensional Bitmap Indexing for XML Documents. In *Proceedings of the Thirteenth International Conference on Scientific and Statistical Database Management*, pages 241-254, 2001.
11. E. Bertino, G. Guerrini, M. Mesiti, I. Rivara, C. Tavella. Measuring the Structural Similarity among XML Documents and DTDs, 2001.

12. M. Mesiti, P. Rosso, and M. Merlo. A Bayesian Approach to WSD for the Retrieval of XML Documents. In *In Proceedings of JOTRI*, pages 11–18, 2002.
13. A. Tagarelli and Sergio Greco. Toward Semantic XML Clustering. In *Proceedings of the Sixth International Conference on Data Mining*, pages 188-199, 2006.
14. L. Kaufman and P. Rousseeuw. *Finding Groups in Data - An Introduction to Cluster Analysis*. J. Wiley, New York, 1990.
15. A. Syropoulos. Mathematics of multisets. In *Multiset Processing: Mathematical, Computer Science, and Molecular Computing points of view*, Lecture Notes in Computer Science 2235, pages 347–358. Springer-Verlag.
16. http://www.cs.umb.edu/~swamir/programs/mudxml.zip.
17. http://www.cs.wisc.edu/niagara/data.html.
18. Available at: http://www.r-project.org/index.html.
19. http://www.sigmod.org/record/xml/.
20. http://www.cs.toronto.edu/tox/toxgene/.