함수 변환과 FFT에 기반한 조정자가 없는 XML 문서 클러스터링 기법

이 호 석†

요 약

본 논문은 함수 변환(Function Transform)과 FFT(Fast Fourier Transform)를 사용하는 새로운 XML 문서 클러스터링 기법에 대하여 논한다. 본 문서 클러스터링 기법은 조정자 없이 점진적으로 수행된다. XML 문서는 엘리먼트의 계층적인 구조에 기반하여 이산 함수로 변환된다. 이산 함수는 FFT를 사용하여 벡터로 변환된다. 문서를 나타내는 벡터는 가중치 유클리디안 거리 메트릭을 사용하여 비교된다. 비교 결과가 미리 정의된 값보다 작을 때에는, 비교되는 두 개의 문서는 구조적으로 비슷한 것으로 간주되어 동일한 그룹으로 분류된다. XML 문서 클러스터링은 XML 문서의 저장과 검색에 유용하게 사용될 수 있다. 800개의 합성 문서와 520개의 실제 문서를 사용하여 실험하였다. 실험 결과는 함수 변환과 FFT는 XML 문서를 엘리먼트의 구조를 기반으로 하여 점진적으로 조정자 없이 효과적으로 분류하는 것을 보여주었다.

키워드 : 조정자 없는 클러스터링, 엘리먼트 구조, 함수 변환, FFT, 가중치 유클리디안 거리

An Unsupervised Clustering Technique of XML Documents based on Function Transform and FFT

Lee, Ho Suk†

ABSTRACT

This paper discusses a new unsupervised XML document clustering technique based on the function transform and FFT(Fast Fourier Transform). An XML document is transformed into a discrete function based on the hierarchical nesting structure of the elements. The discrete function is, then, transformed into vectors using FFT. The vectors of two documents are compared using a weighted Euclidean distance metric. If the comparison is lower than the pre specified threshold, the two documents are considered similar in the structure and are grouped into the same cluster. XML clustering can be useful for the storage and searching of XML documents. The experiments were conducted with 800 synthetic documents and also with 520 real documents. The experiments showed that the function transform and FFT are effective for the incremental and unsupervised clustering of XML documents similar in structure.

Key Words : Unsupervised Clustering, Structure of Elements, Function Transform, FFT, Weighted Euclidean Distance

1. Introduction

XML(Extensible Markup Language) is used as a standard means of information representation and exchange on the internet and on computer science. Applications such as credit card information processing, internet network monitoring, sensor data collection, and e-business services constantly collect and process a large amount of data to foresee trends and to detect anomalies in the applications and services. The amount and fashion of data collected and transferred by these applications necessarily require an incremental and unsupervised clustering of XML documents for document storage, searching, and querying.

Data clustering has been studied during the last decades and many techniques or algorithms have been developed [1]. The data clustering can be classified into hierarchical clustering, partitional clustering, probabilistic model-based clustering, and incremental clustering. The hierarchical clustering attempts to build up meaningful clusters from each document by grouping the related documents into a cluster in a bottom-up fashion. The partitional clustering starts from the whole documents and attempts to partition the whole documents into meaningful clusters in a top-down fashion. The probabilistic clustering uses a probabilistic model to cluster the data. The data are assumed to come from a finite mixture model of probability distributions.

※ This research was supported by the academic research fund of Hoseo University in 2006 (Grant : 20070023)
† 정 회 원 : 호서대학교 공과대학 뉴미디어학과 교수
논문접수 : 2006년 7월 19일, 심사완료 : 2006년 11월 7일

The most commonly used probabilistic model is a Gaussian function. The incremental clustering begins with a single document in a cluster and attempts to cluster the next incoming document into the cluster or builds another cluster to group it. The Euclidean distance is commonly used for the computation of dissimilarity between two points. The weighted Euclidean distance is obtained by multiplying a weight to the Euclidean distance. K-means clustering algorithm is the most frequently used algorithm for clustering. The simple implementation of K-means algorithm computes the average of the cluster and uses it as the centroid for clustering. The algorithm is known to work well for compact and isotropic data. Clustering can also be understood in terms of supervised or unsupervised clustering. Supervised clustering assumes the existence of an estimation function for the appropriateness of the clustering. Unsupervised clustering does not assume an estimation function. The incremental clustering is a kind of unsupervised clustering[2, 3, 4].

The general purpose of data clustering is to derive some relevant information from the data. The clustered data may show some tendency or regularity in the data and may even show some relevant knowledge worth noting. The clustering of XML documents can be used for XML data querying for information retrieval, for the efficient storage of XML documents, and even for system protection purpose because unusual document can be discovered easily. The clustering of documents according to their structure can also facilitate searching because similar documents can be searched and processed within a specific category.

This paper discusses a new technique of XML document clustering based only on the structure of elements of the document and also shows the effectiveness of weighted Euclidean distance metric over the simple K-means clustering algorithm. This paper also compares the proposed approach with the approaches of [5] and [7] and shows the advantages of the proposed approach. The structure of elements of a document is represented by a pair of n-dimensional vectors after the function transform and FFT.

2. Related works

There have been considerable amount of works on document clustering. Tag encoding method was developed for the detection of structural similarity of XML document [5]. They developed encoding functions named as direct encoding, pairwise encoding, and nested encoding. The encoding functions transformed the XML document into time series. They defined the DFT(Discrete Fourier Transform) distance of documents as the approximation of the difference of the magnitude of the DFT of the two encoded documents. The definition of DFT distance had the property of metric satisfying the triangular inequality and used to compute the structural dissimilarity of the documents.

Algorithms were proposed to extract features from XML documents, where a feature may refer to a path or a node pair in a XML tree[6]. They transformed a XML tree into a vector of features and built a high-dimensional matrix to represent the XML trees. They applied principal component analysis to reduce the dimension of the matrix and used the K-means algorithm to cluster the vectors, which meant the documents.

A structure graph (s-graph) was discussed to represent the structure of the XML document[7]. The s graph is a directed graph with nodes and edges, where the nodes are elements or the attributes and the edges represent the parent child relationship or the element attribute relationship. They defined the distance metric using the number of edges in the structure graph. The clustering was implemented in two steps. In step one, they scanned the nodes of the document stored in memory, computed the structure graph, and encoded them in a data structure. In step two, they applied a clustering algorithm to generate the clusters. The structure graph was represented in bit strings, so the actual clustering was performed on a set of bit strings.

An edit script algorithm was discussed to detect changes between documents, especially between the different versions[8]. They defined five edit operations such as insert, delete, update, move, and copy. The algorithm worked in two stages. In the first stage, matching relationships between the old and new versions of a document were produced by applying the path matching algorithm in a bottom-up manner. Then, edit scripts were computed by doing a top-down breadth first search on the two versions. The method using edit distance between tree structures of documents have a time complexity of $O(|n|.|m|)$, where $|n|$ and $|m|$ are the respective sizes of the documents. In reference[9], another algorithm using edit distance was discussed.

There are various approaches for document clustering [10~20]. In reference[21], they discussed feature sets for document clustering. The feature sets were "text-features", "tag-features", and "text plus tag features". They used a publicly available clustering tool to evaluate their feature sets for clustering effectiveness. In references [22, 23, 24], incremental techniques for conceptual clustering were explained. In reference[25], an incremental clustering technique, in which an interactive clustering technique using a mixture of supervised and unsupervised learning methods was discussed. They designed an interactive learning algorithm combining the advantages of both learning methods and applied the algorithm for the learning of spoken language for mobile robot.

3. Problem Statement and Overview of the Proposal

In this paper, a new and efficient clustering algorithm for XML documents based only on the structure of elements is proposed. The basic requirements of the clustering algorithm were considered to be a fast execution and an effective clustering.

Generally, the goal of clustering is to group documents of similar topics together without necessarily considering the structure of document. The goal of this approach, however, is to group documents of similar structures together because it is observed that documents of similar structures contain similar contents and could be clustered together under the condition that the source of documents is already known in advance[26].

The document is read into memory using a DOM parser and the nodes are traversed using a recursive function and the structure of the document is transformed into a discrete function. The discrete function, then, is transformed into frequency domain by the FFT[28]. The result of FFT is a pair of complex numbers consisting of x and y values and considered to be a pair of n-dimensional vectors. The pair of n dimensional vectors is considered to represent the hierarchical nesting structure of the elements in the document. The storage for cluster is represented by a tree structure consisting of cluster heads with information such as cluster name and cluster average and the document nodes with the information of document such as the document file name, the vectors, the averages, and the standard deviations.

The clustering algorithm works in an incremental and unsupervised fashion. Pairs of n-dimensional vectors are compared using a weighted Euclidean distance metric. The first vector is put into the first cluster. The next vector is compared with the first vector in the previous cluster using the weighted Euclidean distance metric. If the comparison is below the pre-specified threshold of the weighted Euclidean distance metric, the current vector is considered similar with the first vector and also with other vectors already in the cluster and grouped into the same cluster, otherwise it is compared with another vector in the next cluster. This process continues until all documents are clustered. The number of clusters is not determined a priori. The structural differences between the documents only determine the number of clusters. In the simple implementation of K-means clustering, the average of the vector is compared with the cluster average. If the comparison is below the pre-specified threshold of the K-means clustering algorithm, the document is grouped into the cluster, otherwise the average of the vector is compared with the average of the next cluster. This process continues until all documents are clustered.

The XML documents used for clustering experiments consists of three groups. One group contains synthetic XML documents generated using ToXgene document generation tool developed by University of Toronto and IBM[27]. The other group contains real XML documents downloaded from a news release web site called www.prweb.com[26]. The third group includes synthetic XML documents from ToXgene documents for the explicit demonstration of clustering of documents with different element structures.

To summarize, this approach shows a new XML documents clustering technique using function transform, FFT, and distance computation by weighted Euclidean distance metric. The complexity of clustering mainly depends on the FFT. This approach is implemented using Java 2 v1.5.0_06. The literal values of elements and attributes are not considered for clustering in this approach.

4. Function Transform

The reference[5] used tag encoding to transform the XML document into time series or signal data as used in most approaches of data mining applications. But a XML document can better be transformed into a discrete function based on the nesting structure of the elements because the nesting structure can be considered the main essential structure representing the semantic structure of the document. The encoded time series or signal data do not seem to represent the semantic structure of the document.

The XML document is parsed and read into memory and the nodes of the document are traversed recursively and a discrete function is generated representing the nesting structure of the elements of the document. The discrete function itself is represented with x coordinate values and the corresponding y coordinate values and stored in arrays.

The following shows the function transform algorithm in a simple procedure.

```
Procedure function_transform(node)
{
  for ( nodes) {
      if (node = document_node) {
         x coordinate computation;
         y coordinate computation;
         call function_tranform(node);
        }
        if (node = element_node) {
           x coordinate computation;
           y coordinate computation;
           get the list of child nodes;
           for(length of list)
             call function_transform(node list);
         }
    }
}
```

The followings show the sample synthetic document "review0.xml" and the discrete function representing the sample document. The "review0.xml" is generated by the ToXgene tool.

The length of x coordinate values is equal to the length of the document. The y values represent the nesting structure of elements. For example, "review" element is at nesting level 1, "book" element at nesting level 2, "title" element at nesting level 3, and finally "end" element at level 1 in the discrete function of "review0.xml" below. The "end" element is not included in the original XML document but is included only in the discrete function to end the function. It can be seen that the y coordinate values correctly represent the nesting structure of elements in the document.

The followings show the document "review0.xml" and the discrete function of the document.

"review0.xml"

```
<?xml version="1.0" encoding="US-ASCII" ?>
 -<!--generated by ToXgene on Thu June 1 18:34:56 KST 2006-->
 -<review>
   -<book isbn="9491701375">
     <title>sly, permanent pearls before the permanent, ruthless</title>
     <author>Usha Degimbe</author>
     <author>Bud Diz</author>
     <author>Hovav Salmon</author>
     <author>Jessie English</author>
     <author>Kojiro Bogle</author>
     <author>Wuu Vefsnmo</author>
     <author>Xiaoquing Vershinin</author>
    </book>
    <user>Balakrishnan.McKinney@verity.com</user>
   -<review rating="3">
       <p>attainments do boost doggedly slyly permanent
          orbits;daring tithes at the ir</p>
       <p>even gifts hinder never;careful notornis will have
          to haggle carefully thin players--courts throughout
          the sly pinto was blithely from the sentiments!daring
          theodolites according to the even, thin forges
          can</p>
    </review>
  </review>
```

Discrete function of "review0.xml"

```
(ELEMENT = review)(X=1,Y=1)
(ELEMENT = book)(X=2,Y=2)
(ELEMENT = title)(X=3,Y=3)
(ELEMENT = author)(X=4,Y=3)
(ELEMENT = author)(X=5,Y=3)
(ELEMENT = author)(X=6,Y=3)
(ELEMENT = author)(X=7,Y=3)
(ELEMENT = author)(X=8,Y=3)
(ELEMENT = author)(X=9,Y=3)
(ELEMENT = author)(X=10,Y=3)
(ELEMENT = user)(X=11,Y=2)
(ELEMENT = review)(X=12,Y=2)
(ELEMENT = p)(X=13,Y=3)
(ELEMENT = p)(X=14,Y=3)
(ELEMENT = p)(X=15,Y=3)
(ELEMENT = p)(X=16,Y=3)
(ELEMENT = end) (X=17,Y=1)
```

The following shows the discrete function of a real document "new0.xml" downloaded from internet. The document contains 216 elements but only 1 through 30 elements are shown without losing the general structure of the discrete function of the document.

Discrete function of "new0.xml"

```
(ELEMENT = rss)(X=1,Y=1)
(ELEMENT = channel)(X=2,Y=2)
(ELEMENT = title)(X=3,Y=3)
(ELEMENT = link)(X=4,Y=3)
(ELEMENT = description)(X=5,Y=3)
(ELEMENT = language)(X=6,Y=3)
(ELEMENT = image)(X=7,Y=3)
(ELEMENT = title)(X=8,Y=4)
(ELEMENT = url)(X=9,Y=4)
(ELEMENT = link)(X=10,Y=4)
(ELEMENT = width)(X=11,Y=4)
(ELEMENT = height)(X=12,Y=4)
(ELEMENT = managingEditor)(X=13,Y=3)
(ELEMENT = webMaster)(X=14,Y=3)
(ELEMENT = ttl)(X=15,Y=3)
(ELEMENT = item)(X=16,Y=3)
(ELEMENT = title)(X=17,Y=4)
(ELEMENT = link)(X=18,Y=4)
(ELEMENT = description)(X=19,Y=4)
(ELEMENT = guid)(X=20,Y=4)
(ELEMENT = item)(X=21,Y=3)
(ELEMENT = title)(X=22,Y=4)
(ELEMENT = link)(X=23,Y=4)
(ELEMENT = description)(X=24,Y=4)
(ELEMENT = guid)(X=25,Y=4)
(ELEMENT = item)(X=26,Y=3)
(ELEMENT = title)(X=27,Y=4)
(ELEMENT = link)(X=28,Y=4)
(ELEMENT = description)(X=29,Y=4)
(ELEMENT = guid)(X=30,Y=4)
 ... ...

(ELEMENT = description)(X=214,Y=4)
(ELEMENT = guid)(X=215,Y=4)
(ELEMENT = end) (X=216,Y=1)
```

5. Fast Fourier Transform

The discrete function is transformed into a frequency domain by FFT[28] for comparison. FFT works efficiently on input of length of powers of two. The length of y values is checked whether it is powers of two. If the

length is not powers of two, zero values are added to make the length of y values powers of two. Then, the FFT performs and produces arrays of complex numbers as the results. The values are produced in the form of (x, y) and are interpreted as a pair of n dimensional vectors representing the structure of the document. The averages and the standard deviations of x and y values are also computed and stored. The complexity of FFT is O(nlogn), when the length of data is powers of two, where n is the length of input data to FFT. The length of output vectors from FFT is the same with the length of the discrete function.

The followings show the document "review882.xml", the result of function transform, and a pair of output vectors computed by FFT, which is considered to represent the XML document with the structures of the document encoded in it.

review882.xml

```
<?xml version="1.0" encoding="US-ASCII" ?>
-<!--generated by ToXgene ··· on Thu May 18 08:59:24
  KST 2006-->
-<review>
-<book isbn="9178217221">
    <title>even forges closely furious warthogs</title>
    <author>Dzung Pierce</author>
    <author>Marisa Vassallo</author>
    <author>Moss Frisberg</author>
    <author>Shigeyuki Tahar</author>
    <author>Amandio Lamma</author>
    <author>Iraj Noakes</author>
    <author>Dannz Lipper</author>
    <author>Baziley Arimoto</author>
    <author>Donko Ullah</author>
    <author>Esen Ghidini</author>
  </book>
  <user>Ferran.Crosby@cas.cz</user>
 -<review rating="1" date="2000 05 10">
    <p>fluffy, into:sly, quick decoys doubt bli</p>
  </review>
</review>
```

Result of function transform of "review882.xml"

```
(ELEMENT = review)(X=1,Y=1)
(ELEMENT = book)(X=2,Y=2)
(ELEMENT = title)(X=3,Y=3)
(ELEMENT = author)(X=4,Y=3)
(ELEMENT = author)(X=5,Y=3)
(ELEMENT = author)(X=6,Y=3)
(ELEMENT = author)(X=7,Y=3)
(ELEMENT = author)(X=8,Y=3)
(ELEMENT = author)(X=9,Y=3)
(ELEMENT = author)(X=10,Y=3)
(ELEMENT = author)(X=11,Y=3)
(ELEMENT = author)(X=12,Y=3)
(ELEMENT = author)(X=13,Y=3)
(ELEMENT = user)(X=14,Y=2)
(ELEMENT = review)(X=15,Y=2)
(ELEMENT = p)(X=16,Y=3)
(ELEMENT = end) (X=17,Y=1)
```

Output vectors from FFT of "review882.xml"

```
23, 0
-2, -13
 0, -1
-3, -1
 2, -1
-2, 0
 1, 0
 0, 0
 1, 0
 0, 0
 1, 0
-2, 0
 2, 1
-3, 1
 0, 1
-2, 13
```

6. Experiments of Clustering

The clustering algorithm works in an incremental and unsupervised fashion. The first vector is put into a cluster and the next vector is compared with the vector in the previous cluster using the weighted Euclidean distance metric. The average of next vector is compared with the cluster average in the case of the K-means clustering algorithm. The unsupervised clustering means that the structural difference of the documents solely determines the clustering process and the number of clusters.

The weighted Euclidean distance constitutes the continuous space and is considered to satisfy the positivity, symmetry, and triangular inequality properties and is used as a distance metric[2]. The following shows the equation.

$$d=(\sum w(xi\ yi)2)1/2 \qquad (1)$$

where w stands for the weight. The average of standard deviation of vector x and vector y was used as the weight of weighted Euclidean distance metric. The weight was used to reduce the influence of spread of the computed distance. The pre-specified threshold for the weighted Euclidean distance comparison can be selected between 0.1~2.5. When the threshold nears 0.1, more clusters are generated because the comparison is minute. The pre-specified threshold means that the documents grouped into a specific cluster are more similar than the documents grouped into a different cluster in terms of the value of pre-specified threshold. The pre-specified threshold of K-means comparison is $0.1x10^{-17}$, because the cluster average is around 1.0 such as 1.0000000000000007 or 0.9999999999999989. When

the threshold is above this value, only one cluster is generated.

The experiments were conducted with three types of XML documents. One type of XML documents consists of the synthetic documents generated by the ToXgene XML document generation tool developed by the University of Toronto and IBM[27]. The characteristics of the documents are as follows, the elements of the documents are all identical, the values of the elements are meaningless because the values are generated by a random distribution, and the structures of the documents are very similar varying only in the frequencies of elements appearing in the document. A total number of 800 documents were used for the clustering experiments. Table 3 shows the experiment results. The other type of XML documents are real ones downloaded from the news release website, www.prweb.com. The documents are about the daily news in every aspect of our lives such as economics, politics, business, sports, computer industry, and entertainment. The characteristics of the documents are as follows, the elements of documents are limited and come from a known set of elements, the structure of documents belonging to a specific news category are similar with repetitions of elements. A total number of 520 documents were used for the clustering experiments. The third type includes four synthetic XML documents from ToXgene documents for the explicit demonstration of clustering with different element structures. The test computer is Pentium IV 1.80GHz with 256MB RAM under the MS Windows XP professional version.

The experiments were conducted in two modes. In mode one, the clustering algorithm was experimented using weighted Euclidean distance metric. The following shows the results of clustering of synthetic documents using the weighted Euclidean distance metric.

```
Weighted Euclidean distance clustering of 200 synthetic xml files
56 clusters created.
Elapsed Time for clustering: 3781(ms).

CLUSTER NAME: cluster(1)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review0.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review47.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review54.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review192.xml
CLUSTER NAME: cluster(2)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review1.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review18.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review70.xml
   ... ...
```

The following shows the results of clustering of synthetic documents using K-means clustering algorithm.

```
K-means clustering of 200 synthetic xml files.
7 clusters created
Threshold=0.00000000000000001
Elapsed Time for clustering: 3297(ms)

CLUSTER NAME: cluster(1)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review0.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review2.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review7.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review8.xml
   ... ...
```

The following shows one sample of real documents named "new0.xml" and shows the results of clustering using the weighted Euclidean distance metric. The file was renamed for the convenience of processing.

"new0.xml"

```
<?xml version="1.0" encoding="ISO 8859 1" ?>
<!DOCTYPE rss (View Source for full doctype...)>
 -<rss version="0.91">
   -<channel>
      <title>PRWeb: Art and Entertainment Celebrities</title>
      <link>http://www.prweb.com</link>
      <description>Latest news releases from PRWEB.COM
                  for Art and Entertainment Celebrities
      </description>
      <language>en</language>
   -<image>
       <title>PRWeb Press Releasse Newswire</title>
       <url>http://www.prweb.com/prwebrss.gif</url>
       <link>http://www.prweb.com</link>
       <width>130</width>
       <height>49</height>
     </image>
      <managingEditor>xml@emediawire.com
      </managingEditor>
      <webMaster>xml@emediawire.com</webMaster>
   -<item>
      <title>Talking Ronald Reagan Doll is Proof that the
            Bible is Real, Says Author</title>
      <link>http://www.prweb.com/releases/20069910/5/
            prweb378649.htm</link>
    -<description>
      -<![CDATA[ Author Gregory Gordon claims that a
        talking doll of Ronald Reagan is the talking image
        of the Beast spoken of in Revelation chapter 13
        verse 15. (PRWEB May 31, 2006)
        Trackback URL: http://www.prweb.com/chachingpr.php
        /TG92ZS1TaW5nLU1hZ24tU3VtbS1IYWxmLVplcm8=
        ]]>
      </description>
    </item>
   -<item> ... ... </item>
      ... ...
 </channel>
</rss>
```

```
Weighted Euclidean clustering 200 real xml files
19 clusters created
Elapsed Time for clustering: 23453(ms)
```

```
CLUSTER NAME: cluster(1)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new0.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new2.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new4.xml
   ... ...
CLUSTER NAME: cluster(2)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new1.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new3.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new5.xml
   ... ...
CLUSTER NAME: cluster(3)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new31.xml
CLUSTER NAME: cluster(4)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new37.xml
CLUSTER NAME: cluster(5)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new45.xml
```

The following shows the results of clustering of real documents using K-means clustering algorithm.

```
K-means clustering of 200 real xml files
threshold=0.00000000000000001
17 clusters created

Elapsed Time for clustering: 23844(ms)

CLUSTER NAME: cluster(1)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new0.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new2.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new4.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new6.xml
   ... ...
CLUSTER NAME: cluster(2)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new1.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new3.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new5.xml
   ... ...
CLUSTER NAME: cluster(3)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new31.xml
CLUSTER NAME: cluster(4)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new37.xml
CLUSTER NAME: cluster(5)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new45.xml
   ... ...
```

We will explain the results of the experiment in detail in the following section 7.

7. Performances

First, we show the clustering differences between using the weighted Euclidean distance clustering and the K-means clustering. They are different because the computation of distances for the comparison of documents is different. When the averages of documents are nearly centered on a single value, i.e., when the structure of the documents are very much similar, the threshold for K-means clustering algorithm must be selected carefully in order to produce reasonable clustering of documents. When the weighted Euclidean distance metric is used, the computation of distances usually takes more time but the clustering is more stable in the experiments, because the distances span more widely.

The following shows the clustering of synthetic documents using the weighted Euclidean distance metric and using the K-means clustering algorithm for the comparison of clustering differences. The document "review0.xml" is shown on page 3.

```
Weighted Euclidean distance clustering of 200 synthetic xml files

CLUSTER NAME: cluster(1)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review0.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review47.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review54.xml

K-means clustering of 200 synthetic xml files.

CLUSTER NAME: cluster(1)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review0.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review2.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/result/output1/review7.xml
```

"review47.xml"

```
<?xml version="1.0" encoding="US ASCII" ?>
 -<!--generated by ToXgene on Wed May 17 20:10:17 KST
      2006-->
 -<review>
   -<book isbn="6301948479">
      <title>Tiresias shall have to maintain;dug</title>
      <author>Jinhua Note</author>
      <author>Manuk Constantineau</author>
     </book>
     <user>Surapant.Peak@ernet.in</user>
    <review rating="3">
     <p>bold, … … ruthle</p>
     <p>sauternes … … quie</p>
    </review>
</review>
```

"review54"

```
<?xml version="1.0" encoding="US ASCII" ?>
 -<!--generated by ToXgene on Wed May 17 20:10:17 KST
      2006-->
 -<review>
   -<book isbn="1998853966">
      <title>bold, … … fluffy</title>
      <author>Haifeng White</author>
      <author>Milton Nisnevich</author>
      <author>Reimund Herber</author>
      <author>Goli Carchiolo</author>
      <author>Hacene Dyckhoff</author>
      <author>Hiroyoshi Hogen</author>
      <author>Merritt Baar</author>
    </book>
    <user>Saurab.McAlister@bellatlantic.net</user>
```

```
-<review rating="2">
    <p>brave, … … after</p>
    <p>permanent, … … darin</p>
    <p>multipliers … … doggedly</p>
    <p>sheaves … … permanent</p>
  </review>
</review>
```

"review2.xml"

```
<?xml version="1.0" encoding="US ASCII" ?>
-<!--generated by ToXgene on Wed May 17 20:10:16 KST
    2006-->
-<review>
  -<book isbn="1310559268">
    <title>quick warthogs</title>
    <author>Dhiraj Jurka</author>
    <author>Klaus Burkhart</author>
   </book>
   <user>Sukemitsu.Lanfear@emc.com</user>
  -<review rating="0">
    <p>stealthly … … regularl</p>
    <p>somas … … across</p>
    <p>slow … … attainm</p>
    <p>ironic … … sentiment</p>
  </review>
</review>
```

"review7.xml"

```
<?xml version="1.0" encoding="US ASCII" ?>
-<!--generated by ToXgene on Wed May 17 20:10:16 KST
    2006-->
-<review>
  -<book isbn="1488943965">
    <title>ironic sen</title>
    <author>Liuba Weinstein</author>
    <author>Michi Jacquin</author>
    <author>Epaminondas Urpani</author>
    <author>Earlin Getta</author>
    <author>Haifeng White</author>
    <author>Gerassimos Altepeter</author>
    <author>Evangelos Kroll</author>
    <author>Jose Drabenstott</author>
    <author>Francisco Molenaar</author>
  </book>
  <user>Augustus.Hedger@forwiss.de</user>
-<review rating="2">
    <p>platelets … … could hav</p>
    <p>permanent … … pl</p>
    <p>dogged, … … brave</p>
    <p>quick, … … beyond</p>
  </review>
</review>
```

If the three documents "review0", "review47", and "review 54" in cluster(1) clustered using the weighted Euclidean distance metric are compared with the three documents "review0", "review2", and "review7" clustered using the K-means clustering algorithm, it can be seen that the documents clustered using the weighted Euclidean distance metric are more similar than the documents clustered using the K-means clustering algorithm.

〈Table 1〉 Count of elements of documents in cluster (1) using the weighted Euclidean distance clustering

Element	review0.xml	review47.xml	review54.xml
book	8	3	8
review	2	2	4

〈Table 2〉 Count of elements of documents in cluster (1) using the K means clustering

Element	review0.xml	review2.xml	review7.xml
book	8	3	10
review	2	4	4

The following Table 1 shows the count of elements in the outer book structure and the review structure in "review0.xml", "review47.xml", and "review54.xml" documents. We can see that the "review0.xml" and "review47.xml" have the same number of two elements in the outer review structure. And we can see that the "review0.xml" and "review54.xml" have the same number of eight elements in the outer book structure.

The following Table 2 shows the count of elements in the outer book structure and the review structure in "review0.xml", "review2.xml", and "review7.xml" documents. We can see that the "review0.xml" and "review2.xml" have nothing in common in the number of elements in the outer book and review structure. We can only see that "review2.xml" and "review7.xml" have the same number of four elements in the outer review structure.

These results show that the weighted Euclidean distance clustering is more effective than the K means clustering in the clustering of similar synthetic documents.

We also experimented with the real documents. The clustering algorithm generated 19 clusters using weighted Euclidean distance metric and 17 clusters using K means clustering algorithm. The followings show a portion of the clustering result of each method.

```
Weighted Euclidean clustering 200 real xml files

CLUSTER NAME: cluster(1)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new0.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new2.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new4.xml

K-means clustering of 200 real xml files

CLUSTER NAME: cluster(1)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new0.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new2.xml
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/news/new4.xml
```

If we compare the clustering results of each method, we can see that the clustering results are very similar except for some clusters. The following explains the similarities and the differences. In both cases, cluster(1) contains nearly even numbered documents and cluster(2) contains nearly odd numbered documents. Cluster(3) through cluster(7) contain same documents. Cluster(8) contains "new51.xml" and "new113.xml" using weighted Euclidean distance metric but contains "new51.xml", "new113.xml", and "new171.xml" using K means clustering algorithm. Cluster(9) through cluster(17) contain the same documents. But cluster(19) contains "new189.xml" using weighted Euclidean distance metric but "new189.xml" was contained in cluster(1) using K-means clustering algorithm. Using the clustering of weighted Euclidean distance metric, cluster(3) contains "new31.xml", cluster(4) contains "new37.xml", cluster(5) contains "new45.xml", cluster(6) contains "new47.xml", cluster(7) contains "new49.xml" and "new69.xml", cluster(9) contains "new61.xml", cluster(10) contains "new103.xml", cluster(11) contains "new105.xml", cluster(12) contains "new115.xml", cluster(13) contains "new145.xml", cluster(14) contains "new151.xml", cluster(15) contains "new170.xml", cluster(16) contains "new171.xml", cluster(17) contains "new172.xml", and cluster(18) contains "new173.xml". After examining the clusters, it can be observed that these clusters were generated based on the information of the nesting structure of the elements and the number of elements in the documents as expected.

Based on these observations, it can be concluded that the weighted Euclidean distance metric is more stable and effective in XML document clustering, because the computation of difference between documents include all the values of vectors not just the averages of the clusters. The clustering results validate the appropriateness of the function transform of XML document and the use of FFT for the generation of vectors for the comparison of difference between the documents.

Second, we conduct another experiment to explicitly demonstrate the clustering based only on the hierarchical nesting structure of the document using the weighted Euclidean distance metric. The following shows four synthetic documents "test0.xml", "test1.xml", "test2.xml", and "test4.xml" produced by editing the "review0.xml" by emphasizing the nesting structure of elements from 1 through 4 levels.

"test0.xml"

```
<?xml version="1.0" encoding="US ASCII"?>
<!--generated by ToXgene on Thu June 1 18:35:01 KST 2006-->
<review>
  <book isbn="1953229935">
    <title>thin courts</title>
    <author>Enno Poesio</author>
    <author>Gunvant Borrowman</author>
  </book>
<user>Katsuji.Camacho@conclusivestrategies.com</user>
  <review rating="2">
    <p>ironic, slow </p>
    <p>regular</p>
    <p>bold, daring</p>
    <p>sometimes</p>
  </review>
</review>
```

"test1.xml"

```
<?xml version="1.0" encoding="US ASCII"?>
<!--generated by ToXgene on Thu June 1 18:35:01 KST 2006-->
<review>
  <book isbn="1953229935">
    <title>thin courts on the sometimes</title>
    <author>Enno Poesio</author>
      <book isbn="1953229935">
        <title>thin courts on the sometimes</title>
        <author>Enno Poesio</author>
      </book>
    <author>Gunvant Borrowman</author>
  </book>
  <user>Katsuji.Camacho@conclusivestrategies.com</user>
  <review rating="2">
    <p>ironic, slow dependencies ought to be yo</p>
    <p>regular dinos to the regular, silent she</p>
    <p>bold, daring epitaphs within the closely</p>
    <p>sometimes idle dinos could have to integ</p>
  </review>
</review>
```

"test2.xml"

```
<?xml version="1.0" encoding="US ASCII"?>
<!--generated by ToXgene on Thu June 1 18:35:01 KST 2006-->
<review>
  <book isbn="1953229935">
    <title>thin courts on the sometimes</title>
    <author>Enno Poesio</author>
      <book isbn="1953229935">
        <title>thin courts on the sometimes</title>
            <book isbn="1953229935">
              <title>thin courts on the sometimes</title>
            </book>
        <author>Enno Poesio</author>
      </book>
    <author>Gunvant Borrowman</author>
  </book>
  <user>Katsuji.Camacho@conclusivestrategies.com</user>
  <review rating="2">
    <p>ironic, slow dependencies ought to be yo</p>
    <p>regular dinos to the regular, silent she</p>
    <p>bold, daring epitaphs within the closely</p>
    <p>sometimes idle dinos could have to integ</p>
  </review>
</review>
```

"test3.xml"

```
<?xml version="1.0" encoding="US ASCII"?>
<!--generated by ToXgene on Thu June 1 18:35:01 KST 2006-->
<review>
  <book isbn="1953229935">
    <title>thin courts on the sometimes</title>
    <author>Enno Poesio</author>
  </book>
  <book isbn="1953229935">
    <title>thin courts on the sometimes</title>
    <author>Enno Poesio</author>
      <book isbn="1953229935">
        <title>thin courts on the sometimes</title>
        <author>Enno Poesio</author>
      </book>
    <author>Gunvant Borrowman</author>
  </book>
  <book>
    <author>Gunvant Borrowman</author>
  </book>
  <user>Katsuji.Camacho@conclusivestrategies.com</user>
  <review rating="2">
    <p>ironic, slow dependencies ought to be yo</p>
    <p>regular dinos to the regular, silent she</p>
    <p>bold, daring epitaphs within the closely</p>
    <p>sometimes idle dinos could have to integ</p>
  </review>
</review>
```

The following shows the expected clustering results.

```
Elapsed Time for clustering: 750(ms)

CLUSTER NAME: cluster(1)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/test/test0.xml
CLUSTER NAME: cluster(2)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/test/test1.xml
CLUSTER NAME: cluster(3)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/test/test2.xml
CLUSTER NAME: cluster(4)
   DOCUMENT NAME: C:/Eclipse/java2/xstream1/test/test3.xml
```

Observing these experimental results, we can see that the four distinct documents are correctly clustered into four separate clusters using the weighted Euclidean distance metric.

Third, we compare the proposed approach with other approaches. In reference[5], the authors used a nearest neighbor method for the evaluation of clustering and defined three measures for the evaluation. First, they used similarity matrix CS to represent the average similarity between classes. Second, they used error rate of a kNN(k Nearest Neighbor) classifier, e, defined as the average of the document membership in a specific cluster. Low values of the error rate correspond to good result. Third, they defined the average percentage of documents, q, in the k neighborhood of a generic document belonging to the same class of that document. This provides a measure of the stability of a Nearest-Neighbor. Fourth, they used a measure of sensibility of the similarity measure, $e(S)$, between documents. Low value of sensibility probability measure denotes a good performance. They used each of the value 0.0892, 0.1880, and 0.7417 as the value of these measures for the nested encoding of the synthesized documents and showed the average similarity value of the nested encoding from 0.058 to 0.840 in Table 5. They used the values 0.1124, 0.0493, and 0.9243 as the values of these measures for the nested encoding of the real documents.

In reference[7], the authors used a structure graph method with four parameters to measure the clustering accuracy. The four parameters are CS, IS, SD, and R. CS means the closeness between the clusters, IS means the average similarity over all pairs of clusters, SD is the standard deviation of the number of documents in the clusters, and finally R means the ratio of outlier documents. A good cluster means a large value of CS (close to 1) and a small IS and SD (close to 0). The values of CS range from 0.4 to 0.97, IS from 0.019 to 0.33, SD from 5 to 13011, and finally R from 0.011 to 0.077 as a function of database size as shown in Table 5. The values of CS range from 0.64 to 0.914, IS from 0.114 to 0.158, SD from 62 to 420, and finally R from 0.009 to 0.22 by varying the number of clusters.

We used the weighted Euclidean distance shown in (1) as a measure of similarity between the documents in the clusters and used the value 0.1~2.5 as the threshold. The distance between the documents clustered into a specific cluster is lower than the threshold. For example, if we set the threshold at the value of 0.5, the distance between "review0.xml" and "review2.xml" is 0.0483 and the distance between "review4.xml" is 0.0867. The threshold value serves the purpose of setting the limit of the similarity between the documents in a cluster and of the dissimilarity between the clusters themselves. From these observations, it can be seen that the weighted Euclidean distance computed based on the function transform and FFT is an effective measure for the computation of similarity between the XML documents for clustering compared to other measures used in the references [5] and [7].

The following Table 3 shows the number of clusters generated using the synthetic documents by varying the threshold values from 0.1 to 2.5. The numbers 56, 66, … 10, 4 mean the number of clusters generated. The execution time of clustering for 200 documents is about 2,620 ms, and for 400 documents about 4,600ms, and for 600 documents about 6,200 ms, and for 800 documents about 8,200ms.

The number of clusters decreases rapidly at the threshold value of 1.0. The users can select the threshold value

〈Table 3〉 Number of clusters generated using the synthetic documents by varying the threshold values

Thr	0.1	0.3	0.5	0.7	1.0	1.5	2.0	2.5
200	56	56	56	48	31	16	10	4
400	66	66	66	57	34	17	10	4
600	71	71	71	59	35	17	10	4
800	74	74	74	61	35	17	10	4

* Thr : Threshold, * 0.1~2.5 : Threshold values
* 200, 400, 600, 800 : No. of documents, * 74, 66, ···, 4 : No. of clusters

〈Table 4〉 Number of clusters generated using the real documents by varying the threshold values

Thr	0.1	0.3	0.5	0.7	1.0	1.5	2.0	2.5
200	19	19	19	19	19	19	19	19
400	29	29	29	29	29	29	29	29
520	35	35	35	35	35	35	35	35

* Thr : Threshold, * 0.1~2.5 : Threshold values
* 200, 400, 520 : No. of documents, * 19, 29, 35 : No. of clusters

〈Table 5〉 Comparison of approaches

[5]		[7]		Proposed approach	
CS	0.058~0.840	CS	0.4~0.97	Threshold(s)	0.1~2.5
$E_k(S)$	0.1880	IS	0.019~0.33	Average(s)	1.0
				SD(s)	7.775
				Clusters(s)	4~74
$Q_k(S)$	0.7417	SD	5~13011	Threshold(r)	0.1~2.5
e(S)	0.0892	R	0.011~0.077	Average(r)	1.00002
				SD(r)	69.74
				Clusters(r)	19~35

* CS : similarity matrix, * Ek(S) : error rate, * Qk(S) : stability,
* e(S) : sensitivity of similarity,
* CS : closeness, * IS : mean similarity, * SD : standard deviation,
* R : outlier ratio
* Threshold(s) : Threshold(synthetic documents)
* Average(s) : average of first cluster(threshold=1.0, documents=200)
* SD(s) : standard deviation of first cluster(threshold=1.0, docs=200)
* Clusters(s) : No. of clusters(synthetic documents)
* Threshold(r) : Threshold(real documents)
* Average(r) : average of first cluster (threshold=1.0, documents=200)
* SD(r) : standard deviation of first cluster(threshold=1.0, docs=200)
* Clusters(r) : No. of clusters(real documents)

from 0.5 to 2.0 depending on the applications.

The following Table 4 shows the number of clusters generated using the real documents by varying the threshold values from 0.1 to 2.5. The numbers 19, 29, 35 mean the number of clusters generated. The execution time of clustering for 200 documents is about 20,700 ms, and for 400 documents about 37,500ms, and for 520 documents about 48,500ms.

The number of clusters generated does not change by varying the threshold values. This is because the structures of the documents belonging to a specific news category are similar but the structures of documents belonging to other news category are quite different and thus cannot be clustered into the same cluster. This can mean that the documents are already clustered according to the news category in this case.

The following Table 5 shows the data of [5, 7], and the proposed approach. We can see that the parameters and the data are different. In[5], CS means the average similarity between classes. $E_k(S)$ means the error rate. $Q_k(S)$ means the average percentage of documents belonging to the same class. e(S) means the sensitivity of the similarity measure. In[7], CS is a measure on the closeness between the clusters. IS is the average similarity over all pairs of clusters. SD is the standard deviation of the number of documents in the clusters. R is the ratio of outlier documents.

But as Table 3 and Table 5 show the number of clusters, the parameters, and the data, the proposed approach is comparable to the approaches of [5] and [7] and is better in terms of the threshold selection for the number of clusters to generate to meet the changing needs of the applications.

And as Table 1 shows the number of elements in the structure and the clustering result, we know that the proposed approach has the advantage in terms of the accuracy of clustering based solely on the structure of elements represented by the function transform and the distance computed by the weighted Euclidean metric.

The incremental and unsupervised clustering technique of this approach is unique in using discrete function transform representing the hierarchical structure of elements in the document for clustering. The clustering, thus, is done based only on the nesting structure of the elements. The pre-specified threshold solely determines the incremental and unsupervised clustering process. It controls the cohesiveness and separation of the clusters. That is, it controls the similarity limit of the documents contained within the clusters and the dissimilarity boundary of the clusters themselves. The current threshold range was selected by the experimentation using the current synthetic and real documents.

8. Conclusion

This paper shows an incremental and unsupervised clustering technique of synthetic and real XML documents using function transform and FFT. This approach clusters the XML documents based on the hierarchical nesting structure of the elements. The difference between the documents was computed using the weighted Euclidean distance metric. If a user wants a fast solution considering only the nesting structure of the XML document, this approach can be a feasible solution. The experiments show that the weighted Euclidean distance metric is more effective for the clustering of XML documents for the synthetic and real XML documents. The proposed approach has the advantage in terms of the accuracy of clustering based solely on the structure of elements. The proposed approach is comparable to the approaches of [5] and [7] and is better in terms of the number of clusters to generate to meet the changing needs of the applications by adjusting the thresholds.

In future research directions, the function transform including the attributes as well as the relationship of the element and attribute will be addressed.

References

[1] A.K. Jain, M.N. Murty, P.J. Flynn, "Data Clustering: A Review," ACM Computing Surveys, Vol.31, No.3, pp.264-323, September 1999.

[2] David Hand, Heikki Mannila, Padhraic Smyth, Principles of Data Mining, The MIT Press, 2001.

[3] Mehmed Kantardzic, Data Mining Concepts, Models, Methods, and Algorithms, IEEE Press, 2003.

[4] Pang Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, Addison Wesley, 2006.

[5] Sergio Flesca, Giuseppe Manco, Elio Masciari, Luigi Pontieri, Andrea Pugliese, "Fast Detection of XML Structural Similarity," IEEE Trans. on Knowledge and Data Engineering, Vol.17, No.2, pp.160-175, February 2005.

[6] Jianghui Liu, Jason T. L. Wang, Wynne Hsu, Katherine G.. Herbert, "XML Clustering by Principal Component Analysis," Proc. of the 16th IEEE Int'l Conf. on Tools with Artificial Intelligence(ICTAI 2004), 2004.

[7] Wang Lian, David Wai lok Cheung, Nikos Mamoulis, Siu Ming Yiu, "An Efficient and Scalable Algorithm for Clustering XML Documents by Structure," IEEE Trans. on Knowledge and Data Engineering, Vol.16, No.1, pp.82-96, January 2004.

[8] Kyong Ho Lee, Yoon Chul Choy, Sung Bae Cho, "An Efficient Algorithm to Compute Differences between Structured Documents," IEEE Trans. on Knowledge and Data Engineering, Vol.16, No.8, pp.965-979, August 2004.

[9] Andrew Nierman, H. V. Jagadish, "Evaluating Structural Similarity in XML Documents," Proc. of the 5th Int'l Workshop on Web and Databases, 2002.

[10] Dongkyu Kim, Sang goo Lee, Jonghoon Chun, Juhnyoung Lee, "A Semantic Classification Model for e Catalog," Proc. of the IEEE Int'l Conf. on E Commerce Technology, 2004.

[11] Mu Chun Su, Chien Hsing Chou, "A Modified Version of the K Means Algorithm with a Distance based on Cluster Symmetry," IEEE Trans. on PAMI, Vol.23, No.6, pp.674-680, June 2001.

[12] Jong Soo Kim, Myoung Ho Kim, "On Effective Data Clustering in Bitemporal Databases," Proc. of the 4th Int'l Workshop on Temporal Representation and Reasoning, pp.54-61, Florida, USA, May 1997.

[13] Sudipto Guha, Rajeev Rastogi, Kyuseok Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attributes," Proc. of 15th Int'l Conf. on Data Engineering," pp.512-521, 1999.

[14] C.C. Aggarwal, J. Han, J. Wang, Philip Yu, "CluStream: A Framework for Clustering Evolving Data Streams," Proc. of Int'l Conf. on Very Large DataBases, pp.81-92, September 2003.

[15] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, Philip S. Yu, "A Framework for On Demand Classification of Evolving Data Streams," IEEE Trans. on Knowledge and Data Engineering, Vol.18, No.5, pp.577-589, May 2006.

[16] David Gondek, Thomas Hofmann, "Non Redundant Data Clustering," Proc. of the 4th IEEE Int'l Conf. on Data Mining, 2004.

[17] M. L. Zaki, C. Aggarwal, "Xrules: An Effective Structural Classifier for XML Data," Machine Learning Journal, Vol 62, No.1-2, pp.137-170, February 2006.

[18] Yuan Wang, David J. DeWitt, Jin Yi Cai, "X-Diff: An Effective Change Detection Algorithm for XML Documents," Proc. of the 19th Int'l Conf. on Data Engineering, pp.519-530, Bangalore India, March 2003.

[19] James W. Cooper, Anni R. Coden, Eric W. Brown, "A Novel Method for Detecting Similar Documents," Proc. of the 35th Annual Hawaii Int'l Conference on System Sciences, 2002.

[20] Pavel Berkhin, "Survey of Clustering Data Mining Techniques," Technical report, Accrue Software, 2002.

[21] Antoine Doucet, Helena Ahonen Myka, "Naïve clustering of a large XML document collection," Proc. of the 1st Annual Workshop of the Initiative for the Evaluation of XML Retrieval(INEX'02), pp.81-88, Germany, December 2002.

[22] Dwi H. Widyantoro, Thomas R. Ioerger, John Yen, "An Incremental Approach to Building a Cluster Hierarchy, Proc. of the 2002 IEEE Int'l Conf. on Data Mining, pp.705-708, 2002.

[23] Pyo Jae Kim, Jin Young Choi, "Incremental Conceptual Clustering Using a Modified Category Utility," Int'l Technical Conference on Circuits/Systems, Computers and Communications, Vol.1, No.1, pp.23-24, July 2005.

[24] Matthaios Theodorakis, Andreas Vlachos, Theodore Z. Kalamboukis, "Using Hierarchical Clustering to Enhance Classification Accuracy," Proc. of the 3rd Hellenic Conf. in Artificial Intelligence, Samos, May 2004.

[25] Qiong Liu, Stephen Levinson, Ying Wu, Thomas Huang, "Interactive and Incremental Learning via a Mixture of Supervised and Unsupervised Learning Strategies," Proc. of the 5th Joint Conf. on Information Science, Vol.1, pp.555-558, Atlantic City, USA, 2002.

[26] PRWeb Press Release Service, http://www.prweb.com.

[27] Denilson Barbosa, "ToXgene Template Specification Language," Dept. of Computer Science, University of Toronto, version 2.1, March 2003.

[28] Alan V. Oppenheim, Ronald W. Schafer, John R. Buck, Discrete Time Signal Processing (2nd ed.), Prentice Hall, 1999.

이 호 석

e-mail: hslee@office.hoseo.ac.kr

1983년 서울대학교 전자계산기공학과 (공학사)

1985년 서울대학교 컴퓨터공학과 대학원 (공학석사)

1993년 서울대학교 컴퓨터공학부 대학원 (공학박사)

1994년 ~ 현재 호서대학교 뉴미디어학과 교수

관심분야 : 컴퓨터공학