Fundamenta Informaticae 97 (2009) 215–234
DOI 10.3233/FI-2009-198
IOS Press

Tensor Framework and Combined Symmetry for Hypertext Mining

Suman Saha[*]**, C.A. Murthy and Sankar K. Pal,**
Center for Soft Computing Research Indian Statistical Institute, India
{ssaha_r, murthy, sankar}@isical.ac.in

Abstract. We have made a case here for utilizing tensor framework for hypertext mining. Tensor is a generalization of vector and tensor framework discussed here is a generalization of vector space model which is widely used in the information retrieval and web mining literature. Most hypertext documents have an inherent internal tag structure and external link structure that render the desirable use of multidimensional representations such as those offered by tensor objects. We have focused on the advantages of Tensor Space Model, in which documents are represented using sixth-order tensors. We have exploited the local-structure and neighborhood recommendation encapsulated by the proposed representation. We have defined a similarity measure for tensor objects corresponding to hypertext documents, and evaluated the proposed measure for mining tasks. The superior performance of the proposed methodology for clustering and classification tasks of hypertext documents have been demonstrated here. The experiment using different types of similarity measure in the different components of hypertext documents provides the main advantage of the proposed model. It has been shown theoretically that, the computational complexity of an algorithm performing on tensor framework using tensor similarity measure as distance is at most the computational complexity of the same algorithm performing on vector space model using vector similarity measure as distance.

Keywords: tensor space, hypertext, internal structure, similarity measure.

1. Introduction

Most of the hypertext categorization systems use simple models of documents and document collections. For instance, it is common to model documents as ‘bags of words’, and to model a collection as a set of documents drawn from some fixed distribution [25]. An interesting question is how to exploit more detailed information about the structure of individual documents, or the structure of a collection of documents. For web page categorization, a frequently used approach is to use hyperlink information,

[*] Address for correspondence: Center for Soft Computing Research, Indian Statistical Institute, India

which improves categorization accuracy [30]. Often hyperlink structure is used to support the predictions of a learned classifier, so that documents that are pointed to by the same page will be more likely to have the same classification. There exists several number of publications using different kinds of features (URL, anchortext, meta-tags, neighborhood, etc.....), but finally they are represented in a single vector, thereby loosing the information about the structural component of hypertext where the word appeared. As an example, the fact that a word appearing in the title or URL is more important than the same word appearing in the text content, is ignored. Details of hypertext features have been given in section 2.

The information regarding the structure of hypertext document is not frequently used in web page categorization algorithms. Vector Space Model (VSM), the footstone of many web mining and information retrieval techniques [28], is used to represent the text documents and define the similarity among them. Bag of Word (BOW) [18] is the earliest approach used to represent document as a bag of words under the VSM. In the BOW representation, a document is encoded as a feature vector, with each element in the vector indicating the presence or absence of a word in the document by TFIDF (Term Frequency Inverse Document Frequency) indexing. A document vector has no memory about the structure of the hypertext. Information about the HTML markup structure and hyperlink connectivity is ignored in VSM representation.

In this article we have proposed a novel tensor framework for hypertext representation. Our model relies on different types of features, which are extracted from a hypertext document and its neighbors. The proposed model consists of a sixth order tensor for each hypertext document. In this representation the features extracted from URL or Title or any other part are assigned to different vector spaces and tensor is defined on the product space. This representation model does not ignore the informations about internal markup structure and link structure of the hypertext documents. Details of the proposed, tensor framework for hypertext representation are given in section 3.

Mathematical formulation of the proposed model and some definitions regarding tensor similarity have been also given in section 3. The tensor similarity measure computes component wise similarity between two hypertext and add up the similarities of all components to obtain the similarity between two tensors corresponding to hypertexts. Similarity measure defined in this article compares two hypertext documents based on their corresponding parts, which is not possible in a vector space model. Sum of component's similarity differ from similarity of all feature vectors (4.2). A tensor similarity measure using different types of similarity for different component has been discussed in section 4.4. In section 4.3, it has been shown theoretically that, the computational complexity of an algorithm performing on tensor framework using tensor similarity measure as distance is at most the computational complexity of the same algorithm performing on vector space model using vector similarity measure as distance.

The experimental results regarding the performance of learning algorithms on this representation model are stated in section 5. Experimental results regarding comparison of VSM using all features and TSM are given in subsections 5.3 and 5.4 Experiments regarding semantic similarity measure for some selected components are given in subsection 5.7.

2. Hypertext features

A hypertext document consists of different types of features which are found to be useful for representing a web page [11]. Written in HTML, web pages contain additional information other than text content, such as URLs, hyperlinks, content of neighborhood pages and anchor text (Fig 1). These features can be divided into two broad classes: on-page features, which are directly located on the page to be represented, and features of neighbors, which are found on the pages related in some way with the page to be represented.

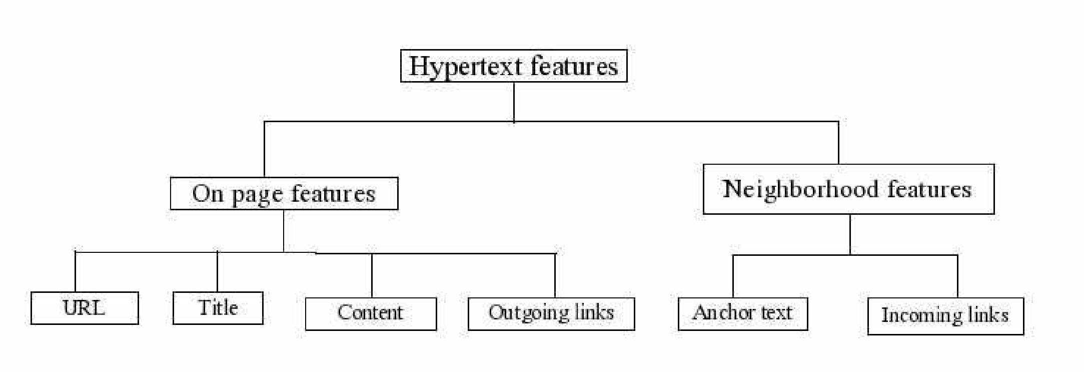


Figure 1. Different type of features of hypertext document

Most commonly used on-page features are URL of the web page, outgoing links of web page, HTML tags, title-headers and text body content of the web page.

1) Features of URL: Uniform resource locaters (URLs), which mark the address of a resource on the World Wide Web, provide valuable information about the document and can be used to predict the category of the resource. A URL is first divided to yield a baseline segmentation into its components as given by the URI protocol (e.g., scheme :// host / path elements / document . extension), and further segmented wherever one or more non-alphanumeric characters appear (e.g., faculty-info → faculty info).

2) Anchor text: Anchor text usually provides relevant descriptive or contextual information about the content of the link's destination. Thus it can be used to predict the category of the target page. Anchor text can provide a good source of information about a target page because it represents how people linking to the page actually describe it. Several studies have tried to use either the anchor text or the text near it to predict a target page's content.

3) Link structure: Link structure of the Web offers some important information for analyzing the relevance and quality of Web pages. Intuitively, the author of a Web page A, who places a link to Web page B, believes that B is relevant to A. The term in-links refers to the hyperlinks pointing to a page. Usually, the larger the number of in-links, the higher a page will be rated. The rationale is similar to citation analysis, in which an often-cited article is considered better than the one never cited. The assumption is made that if two pages are linked to each other, they are likely to be on the same topic. One study actually found that the likelihood of linked pages having similar textual content was high, if one considered random pairs of pages on the Web [8]. Researchers have developed several link-analysis algorithms over the past few years. The most popular link-based Web analysis algorithms include PageRank [4] and HITS [33].

4) Neighborhood category: Category of the already classified neighboring pages can be used to determine the categories of unvisited web pages. In general, features of neighbors provide an alternative view of a web page, which supplement the view from on-page features. Therefore, collectively

considering both can help in reducing the categorization error. Underlying mechanism of collective inference has been investigated by the researchers and has been argued that the benefit does not only come from a larger feature space, but also from modeling dependencies among neighbors and utilizing known class labels [7]. Such explanations may also apply to why web page classification benefits from utilizing features of neighbors.

5) Title and headers: Title and headers can be most significant features found in a hypertext document, because they generally summarize the content of the page. Researchers have shown that incorporating features of title and headers improve the categorization results.

6) Text content: The text on a page is the most relevent component for categorization. However, due to a variety of uncontrolled noises in web pages, a bag-of-words representation for all terms may not result in top performance. Researchers have tried various methods to make better use of the textual features. Popular methods are feature selection and n-gram representation. Feature vector for n-gram representation includes not only single terms, but also up to 5 consecutive words [18]. The advantage of using n-gram representation is that it is able to capture the concepts expressed by a sequence of terms (phrases), which are unlikely to be characterized using single terms. However, an n-gram approach has a significant drawback; it usually generates a space with much higher dimensionality than the bag-of-words representation does. Therefore, it is usually performed in combination with feature selection [18].

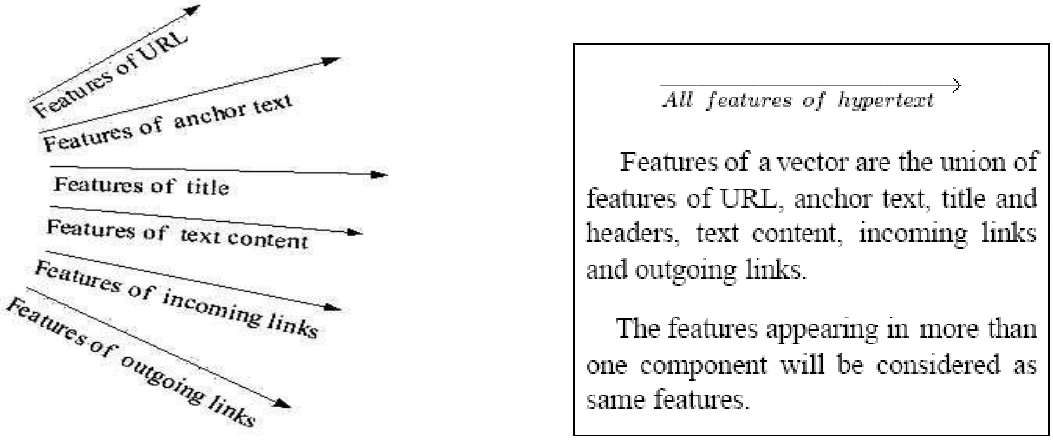
3. Tensor space model

Tensors provide a natural and concise mathematical framework for formulating and solving problems in high dimensional space analysis [3]. Tensor algebra and multilinear analysis have been applied successfully in many domains such as; face recognition, machine vision, document analysis, feature decomposition, text mining etc. [21, 26, 16, 17, 6, 5, 20, 22]. An n-order tensor in m-dimensional space is a function that has n indices and m^n value fields. i.e. $\mathcal{T} : V \times V \times \cdots \times_n V \longrightarrow R$, where $\mathcal{T}$ is a tensor, V is a m dimensonal vector space and R is the real line. Tensor is also defined on product of different vector spaces. i.e. $\mathcal{T} : V_1 \times V_2 \times \cdots \times_n V_n \longrightarrow R$, where $\mathcal{T}$ is a tensor, V_k is a m_k dimensonal vector space and R is the real line.

Tensors are generalizations of scalars (0-order), which have no indices, vectors (1-order), which have a single index, and matrices (2-order), which have two indices and the domain is the product of same vector space.

Document indexing and representation has been a fundamental problem in information retrieval for many years. Most of the previous works are based on the Vector Space Model (VSM). The documents are represented as vectors, and each word corresponds to a dimension. In this section, we introduce a new Tensor Space Model (TSM) for document representation. In Tensor Space Model, a document is represented as a tensor (Fig 2), where domain of the tensor is the product of different vector spaces. Each vector space is associated with a particular type of features of the hypertext documents. The vector spaces considered here are corresponding to $1)$ features of URL, $2)$ features of anchor text, $3)$ features of title and headers, $4)$ features of text content, $5)$ features of outgoing links and $6)$ features of incoming links, the features are word in our case.

In this paper, we propose a novel Tensor Space Model (TSM) for hypertext representation. The proposed TSM is based on different types of features extracted from the HTML document and their neighbors. It offers a potent mathematical framework for analyzing the internal markup structure and link structure of HTML documents along with text content. The proposed TSM for hypertext consists of a 6^{th} order tensor, for each order the dimension is the number of terms of the corresponding types extracted from the hypertexts.



Features of a vector are the union of features of URL, anchor text, title and headers, text content, incoming links and outgoing links.

The features appearing in more than one component will be considered as same features.

(a) TSM (b) VSM

Figure 2. Hypertext representation using (a) tensor framework and (b) vector space model.

Hypertext documents are tokenized with syntactic rules and canonical forms. First we select a set of relevant features from an HTML document. For each type of feature a vector space is constructed. Tensor is defined on the product of this vector spaces. The proposed tensor framework captures the structural representation of hypertext document.

4. Hypertext mining using TSM

4.1. Tensor similarity

Let $\mathcal{T}$ be the tensor space corresponding to hypertext documents. Each member T of $\mathcal{T}$ is of the form $T = t_{ijkxyz}$ where i ranges over the number of features obtained from URL, i.e. i ranges over the dimension of the vector space V_u corresponding to the features of URL. Similarly j ranges over the dimension of the vector space V_a corresponding to the features of anchor text, k ranges over the dimension of the vector space V_t corresponding to the features of title and headers,x ranges over the dimension of the vector space V_c corresponding to the features of text content,y ranges over the dimension of the vector space V_{out} corresponding to the features of outgoing links and z ranges over the dimension of the vector space V_{in} corresponding to the features of incoming links.

The tensor similarity between two tensor T_1 and T_2 of $\mathcal{T}$ is defined as

$$sim(T_1, T_2) = \sum_r sim_r(P_r(T_1), P_r(T_2)),$$

where $sim_r(P_r(T_1), P_r(T_2))$ is the sim_r similarity between $P_r(T_1)$ and $P_r(T_2)$, the function P_r defined as $P_r : \mathcal{T} \longrightarrow V_r,$. Now, for each r, the $P_r = T|_{V_r}$. $sim_r(P_r(T_1), P_r(T_2))$ is basically the similarity between two vectors in V_r. Note that, here sim_r is chosen depending on the characteristic of the r^{th} vector space.

4.2. Similarity measures on VSM

4.2.1. Cosine similarity

Cosine similarity is a measure of similarity between two vectors of same dimension by finding the angle between them. It is often used to compare documents in text mining. Given two vectors of attributes, A and B, the cosine similarity, $Sim(A, B) = (A.B)/(|A|.|B|)$ where the word vectors $AandB$ are vectors found after removing stop words and stemming. For text matching, the attribute vectors A and B are usually the tf-idf vectors of the documents. The resulting similarity will yield the value of 0 meaning, the vectors are independent, and 0 meaning, the vectors are same, with in-between values indicating intermediate similarities or dissimilarities.

4.2.2. Semantic similarity

A semantic network is a directed graph consisting of vertices, representing concepts, and edges, representing semantic relations between the concepts. A semantic network is often used as a form of knowledge representation. WordNet (http://wordnet.princeton.edu/), an online lexical database of English, is an example of a semantic network. The semantic vectors are created using the extended vector using semantic network for a given word vector. The extended vector includes the terms of the word and the related concepts in the semantic network. Semantic similarity between two word vectors is determined by calculating the Cosine measure between the semantic vectors associated with the word vectors [13].

4.3. Computational complexity on TSM

Let n be the total number of features of hypertext documents under consideration. Let $n_1, n_2, \ldots, n_r$ be the number of features associated with the $1^{st}, 2^{nd}, \ldots, r^{th}$ components of the tensor respectively. From the definition of TSM we obtain $\sum_{i=1}^{r} n_i = n$. Let m be the number of documents. The complexity of an algorithm $\mathcal{A}$, constructed on VSM can be expressed as $f(m, n, \alpha)$, where α is corresponding to specific parameters of $\mathcal{A}$. The expression of complexity $f(m, n, \alpha)$ is written as: $O(m^i n^j \alpha^k)$. The complexity of the same algorithm $\mathcal{A}$, constructed on TSM can be written as: $O(m^i n_t^j \alpha^k)$, where $n_t = max_{s=1}^{r}\{n_1, n_2, \ldots, n_r\}$. Since, $n_t < n$, we can write $(n_t)^j \leq n^j$. Hence, $O(m^i n_t^j \alpha^k) \leq O(m^i n^j \alpha^k)$. Thus, the following theorem can be stated.

Theorem: Computational complexity of an algorithm performing on tensor framework using tensor similarity measure as distance is at most the computational complexity of the same algorithm performing on vector space model using vector similarity measure as distance.

4.4. Similarity computation on different vector spaces

It has been stated in section 2 that different types of features are present in an HTML document. The computation of similarity between two observations, corresponding to each of the features are described below.

Content similarity: $c(A, B) = (A.B)/(|A|.|B|)$ where A, B are two word vectors, represented after removing stop words and stemming. This is actually the 'cosine similarity' function, traditionally used in information retrieval.

URL similarity: The URL similarity is measured by the common substrings that the URLs of two web pages have. A URL is first divided to yield a baseline segmentation into its components as given by the URI protocol (e.g., scheme :// host / path elements / document . extension), and further segmented wherever one or more non-alphanumeric characters appear (e.g., faculty-info → faculty info). These segmented substrings are treated as words. All these words found in a URL will be represented in a vector and cosine similarity measure will be applied to these.

Anchor text similarity: The anchor text similarity of two pages measures the similarity of the anchor text in those two pages. It is computed the same way as content similarity, except substituting each document by a virtual document. The virtual document is created considering only the anchor texts found inside that document. Still, the similarity score is computed as the cosine similarity of the two vectors, each representing a virtual document. IDF is estimated on the collection of these virtual documents.

Title-headers similarity: The title-headers similarity between two documents measures the similarity of the title and headers in those two pages. It is computed in the same way, as content similarity, except substituting each document by a virtual document. The virtual document is created considering only the title and headers found inside that document. The similarity score is computed as the cosine similarity of the two vectors, each representing a virtual document. IDF is estimated on the collection of these virtual documents.

In-link similarity: All in-links are first divided to yield a baseline segmentation into its components as given by the URI protocol, and further segmented wherever one or more non-alphanumeric characters appear. The tokens obtained by segmentations of the in-links are stored in a vector. The cosine similarity is computed between two vectors.

Out-link similarity: Out-link similarity is computed in the same way as in-link similarity.

5. Experimental results

We performed a large number of experiments to evaluate the proposed tensor framework for hypertext representation. The experiments are conducted on four hypertext datasets using tensor framework for representation and tensor similarity measure as distance between two hypertext documents. The purpose of the experiments are to find the clustering and classification of the data sets for tensor framework. The results of clustering and classification have been compared with the results obtained using the same algorithms for clustering and classification respectively considering vector space model for hypertext representation and vector similarity measure as distance between two hypertext documents. We obtained better results on all datasets for both clustering and classification when proposed model is considered. These are described below.

5.1. Data collection

We used four data sets, Looksmart, Dmoz, webkb and Yahoo for our experiments. We crawled the Looksmart and Dmoz web directories. These directories are well known for maintaining a categorized hypertext documents. The web directories are multi-level tree-structured hierarchy. The top level of the tree, which is the first level below the root of the tree, contains 13 categories in Looksmart (Table 2) and 16 categories for Dmoz (Table 1). Each of these categories contains sub-categories that are placed in the second level below the root. We use the top-level categories to label the web pages in our experiments.

Table 1. Class distribution and features of the dmoz data in links and pages.

(a)

Class	#Pages	%Pages	#Links	%Links
Arts	1855	6.27	4292	8.25
Business	1672	5.65	3665	7.04
Computers	2017	6.82	3946	7.58
Games	1500	5.07	2124	4.08
Health	1343	4.54	3210	6.17
Home	1786	6.04	2895	5.56
Sports	2537	8.58	3374	6.48
Kids and Teens	2290	7.74	2978	5.72
News	2626	8.88	3702	7.11
Recreation	2631	8.89	2996	5.76
Reference	1032	3.49	3389	6.51
Regional	1492	5.04	5441	10.46
Science	2387	8.07	2977	5.72
Shopping	1596	5.39	2020	3.88
World	1529	5.17	2093	4.02
Society	1271	4.29	2896	5.56
Total	29564	100	51998	100

(b)

Components	# features
URL	27935
Anchor	25111
Title	36965
Text	104126
In-link	23903
Out-link	21878
Total	239918
Union	188519
Intersection	51399

The webkb data set was collected from the WebKB project. The pages in the WebKB dataset are classified into one of the categories Student, Course, Department, Faculty, Project, Staff and Other (Table 3). Here there are 8077 documents in 7 categories. The largest category (Other) consists of 3025 pages; while the smallest category (Staff) consists of only 135 pages.

Another data set consisting of 40000 Web pages is obtained from the Yahoo (http://dir.yahoo.com) topic directory. This is a big hypertext corpora, manually classified by the human experts. The extracted subset includes 33253 pages, which are distributed among 14 top level categories. The largest category (Science) consists of 4627 pages; while the smallest category (Regional) consists of only 782 pages. Detailed information about number of pages and number of links in the each categoryof the Yahoo data set is given in the Table 4.

Table 2. Class distribution and features of the looksmart data.

(a)

Class	#Pages	%Pages	#Links	%Links
Auto	677	5.38	1859	7.12
Education	1211	9.64	3463	13.26
Health	1087	8.65	2655	10.17
Money	631	5.02	1193	4.57
Recreation	131	1.04	654	2.50
Style	976	7.76	1353	5.18
Travel	595	4.73	1622	6.21
Cities	1245	9.91	2396	9.17
Food	1203	9.57	2371	9.08
HomeLiving	1676	13.34	2796	10.71
Music	1236	9.83	2971	11.38
Sports	742	5.90	1483	5.68
Tech Games	1152	9.17	1285	4.92
Total	12562	100	26101	100

(b)

Components	# features
URL	17469
Anchor	17766
Title	11463
Text	41153
In-link	16599
Out-link	13272
Total	117722
Union	86822
Intersection	30900

Table 3. Class distribution and features of the webkb data.

(a)

Class	#Pages	%Pages	#Links	%Links
Student	1639	20.29	2544	19.07
Faculty	1121	13.87	2147	16.09
Course	926	11.46	1229	9.21
Project	701	8.67	1083	8.11
Department	530	6.56	1194	8.95
Other	3025	37.45	4730	35.45
Staff	135	1.67	413	3.09
Total	8077	100	13340	100

(b)

Components	# features
URL	12898
Anchor	10515
Title	16193
Text	23582
In-link	14529
Out-link	14094
Total	91811
Union	72071
Intersection	19740

Table 4. Class distribution and features of the yahoo data.

(a)

Class	#Pages	%Pages	#Links	%Links
Arts	2731	8.21	4269	7.70
Business	4627	13.91	6092	11.00
Computers	3205	9.63	6444	11.63
Education	2976	8.94	5357	9.67
Entertainment	1592	4.78	2184	3.94
Government	782	2.35	1703	3.07
Health	2542	7.64	3999	7.22
News	3716	11.17	6580	11.88
Recreation	1482	4.45	2965	5.35
Reference	1183	3.55	3165	5.71
Regional	1020	3.06	2219	4.00
Science	3350	10.07	4486	8.10
Social Sc.	2859	8.59	3493	6.30
Society	1188	3.57	2424	4.37
Total	33253	100	55380	100

(b)

Components	# features
URL	34045
Anchor	31863
Title	43428
Text	127459
In-link	44720
Out-link	40163
Total	321678
Union	256118
Intersection	65560

We processed the data sets to remove image and scripts followed by stop-words removal and stemming. Link graph has been constructed for each of the datasets for extracting neighborhood features. URLs have been segmented for extracting URL features. Finally features extracted from all the components of hypertext have been represented using tensor framework and vector space model for our experiments.

5.2. Evaluation measure

We employ the standard measures to evaluate the performance of Web classification, i.e. precision, recall and F_1-measure. Precision (P) is the proportion of actual positive class members returned by the system among all predicted positive class members returned by the system. Recall (R) is the proportion of predicted positive members among all actual positive class members in the data. F_1 is the harmonic average of precision and recall as shown below: $F1 = \frac{2PR}{P+R}$. To evaluate the average performance across multiple categories, there are two conventional methods: micro-average-F_1 and macro-average-F_1. Micro-average-F_1 is the global calculation of F_1 measure regardless of categories. Macro-average-F_1 is the average on F_1 scores of all categories. Micro-average gives equal weight to every document, while macro-average gives equal weight to every category, regardless of its frequency. In our experiments, precision, recall micro-average-F_1 and macro-average-F_1 will be used to evaluate the performance of classification.

5.3. Clustering results

We can define the goal in hard flat clustering as follows. Given (i) a set of documents $D = \{d_1, \ldots, d_N\}$, (ii) a desired number of clusters K, and (iii) an objective function that evaluates the quality of a clustering, we want to compute an assignment $D \rightarrow \{1, ..., K\}$ that minimizes the objective function. Note that none of the K clusters is empty. The objective function is often defined in terms of similarity or distance between documents. In our experiment the objective in K-means clustering is to minimize the average distance between documents and the corresponding centroids. For comparing the proposed tensor framework with vector space model, we have used tensor similarity measure and vector similarity measure respectively in the objective functions. The performance of these two clustering methods have been observed on four different datasets, WebKB, Looksmart, Yahoo and Dmoz. The comparisons are shown in tables 5, 5 5 and 5. It can be observed from the tables that clustering results are better when tensor framework for hypertext representation is considered compared to clustering results when vector space model for representation is considered. The results are shown in terms of precision, recall, micro-average F_1 and macro-average-F_1.

Table 5. Results of k-means clustering on VSM and TSM in terms of (a) precision, (b) recall, (c) micro-average-F_1 and (d) macro-average-F_1

(a)

Data set	VSM	TSM	Better?
Dmoz	54.72	61.86	√
Looksmart	68.30	76.41	√
WebKB	50.26	56.10	√
Yahoo	64.33	69.79	√

(b)

Data set	VSM	TSM	Better?
Dmoz	50.01	54.68	√
Looksmart	65.82	68.08	√
WebKB	48.44	51.81	√
Yahoo	61.23	64.43	√

(c)

Data set	VSM	TSM	Better?
Dmoz	52.25	58.04	√
Looksmart	67.03	72.00	√
WebKB	49.33	53.86	√
Yahoo	62.74	67.00	√

(d)

Data set	VSM	TSM	Better?
Dmoz	50.17	56.33	√
Looksmart	66.12	71.31	√
WebKB	47.56	50.22	√
Yahoo	61.48	63.84	√

5.4. Classification results

Decisions of many vector space classifiers are based on a notion of distance, e.g., when computing the nearest neighbors in k-NN classification. For evaluation of the proposed tensor framework for hypertext representation, we have constructed two k-NN classifiers. In the first k-NN classifier vector space representation for hypertext document is considered and vector similarity measure is used to compute nearest neighbor. In the second k-NN classifier proposed tensor space model for hypertext representation is considered and proposed tensor similarity measure is used to compute nearest neighbor. distance as the underlying distance. The performance of these two classifiers have been observed on four different datasets, webKB, Looksmart, Yahoo and Dmoz. The classification results of comparisons are shown in tables 6(a), 6, 6 and 6. It can be observed from the tables that classification results are better when tensor framework for hypertext representation is considered compared to classification results when vector space model for representation is considered. The results have been shown in terms of precision, recall, micro-average-F_1 and macro-average-F_1.

Table 6. Results of k-NN classification on VSM and TSM in terms of (a) precision, (b) recall, (c) micro-average-F_1 and (d) macro-average-F_1

(a)

Data set	VSM	TSM	Better?
Dmoz	92.94	96.44	√
Looksmart	93.74	96.97	√
WebKB	91.85	95.79	√
Yahoo	88.24	91.97	√

(b)

Data set	VSM	TSM	Better?
Dmoz	83.27	91.36	√
Looksmart	85.54	92.20	√
WebKB	85.80	90.70	√
Yahoo	82.36	88.18	√

(c)

Data set	VSM	TSM	Better?
Dmoz	87.84	93.83	√
Looksmart	89.45	94.52	√
WebKB	88.72	93.17	√
Yahoo	85.20	90.03	√

(d)

Data set	VSM	TSM	Better?
Dmoz	85.69	89.61	√
Looksmart	87.18	90.57	√
WebKB	85.76	87.86	√
Yahoo	84.50	86.02	√

5.5. Comparisons with some hypertext clustering techniques

We have also compared the performance of tensor framework with existing clustering techniques. A brief review of existing hypertext clustering techniques are given below and these methods are considered for comparisions.

A_0) In the article, ”Web Document Clustering: A Feasibility Demonstration”[32], an incremental, linear time (in the document collection size) algorithm called Suffix Tree Clustering (STC) was introduced. This method creates clusters based on phrases shared between documents. It was shown that STC performed faster than standard clustering methods in this domain, and the authors argued that Web document clustering via STC is both feasible and potentially beneficial.

B_0)In the article, ”Web document clustering using hyperlink structures”[12], document clustering using normalized-cut method was proposed. This method considers textual informations, hyperlink structures and co-citation relations for document clustering.

C_0)The article ”Clustering Relational Data Using Attribute and Link Information” ([19], describes the work synthesizing data clustering and graph partitioning techniques into improved clustering algorithms for relational data.

D_0) In the article, ”ReCoM: Reinforcement Clustering of Multi-Type Interrelated Data Objects” [27], a novel clustering approach for clustering multi-type interrelated data objects has been proposed. In this approach, relationships among data objects are used to improve the cluster quality of interrelated data objects through an iterative reinforcement clustering process. At the same time, the link structure derived from relationships of the interrelated data objects is used to differentiate the importance of objects and the learned importance is also used in the clustering process to further improve the clustering results.

E_0) The article, ”Explaining Text Clustering Results using Semantic Structures”[13], discusses a way of integrating a large thesaurus and the computation of lattices of resulting clusters into common text clustering in order to achieve an explanation using an appropriate level of granularity at the concept level.

F_0)In the article, ”Utilizing Hyperlink Transitivity to Improve Web Page Clustering”[14], an approach to measure web page similarity has been proposed. This approach takes hyperlink transitivity and page importance into consideration to compute similarity and uses this to cluster web pages.

G_0) The article, ”Multi view clustering”[2], describes the development and study of hierarchical multi-view clustering algorithms for text data. It has been found empirically that the multiview versions of k-Means and EM (expectation maximization) greatly improve on their single-view counterparts.

H_0) The article, ”Web Documents Clustering with Interest Links”[10], the web documents in WWW Cache is modeled as an undirected web graph. Then the clustering algorithm based on the web graph model is given. Finally, Experimental results show that the algorithm is efficient and feasible.

We have compared our method with other hypertext clustering algorithms mentioned above. Results, in terms of precision, recall, micro-F_1 and macro-F_1 of A_0, B_0, C_0, D_0, E_0, F_0, G_0, H_0 and TSM has been reported in tables 7, 8, 9 and 10 respectively. Same set of features has been considered for all the algorithms. It can be observed that performance of TSM is marginally better than others in terms of the measures.

Table 7. Comparison of TSM with other hypertext clustering methods in terms of precision.

DATA SET	A_0	B_0	C_0	D_0	E_0	F_0	G_0	H_0	TSM
DMOZ	51.80	52.49	43.37	47.96	58.49	53.29	52.75	56.94	61.86
LOOKSMART	67.41	71.41	62.16	72.01	71.20	69.11	75.10	73.74	76.41
WEBKB	47.00	49.81	39.70	55.31	50.98	48.31	54.55	51.85	56.10
YAHOO	60.25	66.82	66.85	63.47	59.81	62.39	67.74	68.24	69.79

Table 8. Comparison of TSM with other hypertext clustering methods in terms of recall.

DATA SET	A_0	B_0	C_0	D_0	E_0	F_0	G_0	H_0	TSM
DMOZ	49.91	54.37	45.86	49.31	48.84	48.44	52.63	52.45	54.68
LOOKSMART	56.46	63.11	61.09	59.79	66.62	60.02	63.01	65.54	68.08
WEBKB	50.00	47.67	48.51	45.15	49.99	45.53	51.75	50.81	51.81
YAHOO	57.20	58.40	62.02	61.65	62.96	59.05	58.79	62.37	64.43

Table 9. Comparison of TSM with other hypertext clustering methods in terms of micro average F_1.

DATA SET	A_0	B_0	C_0	D_0	E_0	F_0	G_0	H_0	TSM
DMOZ	50.83	53.41	44.58	48.62	53.23	50.74	52.68	54.60	58.04
LOOKSMART	61.45	67.00	61.62	65.33	68.83	64.24	68.52	69.39	72.00
WEBKB	48.45	48.71	43.66	49.71	50.48	46.87	53.11	51.32	53.86
YAHOO	58.68	62.32	64.34	62.54	61.34	60.67	62.94	65.17	67.00

Table 10. Comparison of TSM with other hypertext clustering methods in terms of macro average F_1.

DATA SET	A_0	B_0	C_0	D_0	E_0	F_0	G_0	H_0	TSM
DMOZ	48.11	50.42	43.69	46.13	52.95	48.83	51.45	52.92	56.55
LOOKSMART	60.76	65.20	59.05	63.74	65.87	63.81	67.30	67.11	71.51
WEBKB	48.00	47.32	41.96	48.95	49.03	44.41	52.47	49.56	50.03
YAHOO	57.50	60.13	63.45	60.26	60.31	58.09	60.46	62.79	63.57

5.6. Comparisons with some recent classification techniques

We have also compared the performance of tensor framework with existing classification techniques. A brief review of existing hypertext classification techniques are given below and these methods are considered for comparisions.

$A_1)$ The article ”Enhanced hypertext categorization using hyperlinks”[7], is the first hypertext classification system that combines textual and linkage features into a general statistical model to infer the of interlinked documants. Relaxation labeling technique is used for better classification by exploiting link information in a small neighborhood around documents.

$B_1)$ In the article, ”A Study of Approaches to Hypertext Categorization”[31], it has been shown that adding the words in the linked neighborhood to the page having those links are helpful for the classification. It has been also observed that extracting meta data from related web sites is extremly useful for improving classification accuracy.

$C_1)$ The article ”Improving A Page Classifier with Anchor Extraction and Link Analysis”[9], describes a technique that improves a simple web page classifier’s performance on pages from a new, unseen web site, by exploiting link structure within a site as well as page structure within hub pages. On real-world test cases, this technique significantly and substantially improves the accuracy of a bag-of-words classifier, reducing error rate by about half, on average.

$D_1)$ The article ”Fast webpage classification using URL features”[15], uses URLs for web page categorization via a two-phase pipeline of word segmentation and classification. This method is compared against document-based methods, which require the retrieval of the source document.

$E_1)$ The article, ”Link-Local Features for Hypertext Classification”[24], demonstrates the need to focus on relevant parts of predecessor pages, namely on the region in the neighborhood of the origin of an incoming link. Authors have investigated different ways for extracting such features, and compared several different techniques for using them in a text classifier.

$F_1)$ The article, ”A Comparison of Implicit and Explicit Links for Web Page Classification”[23], provides an approach for automatically building the implicit links between Web pages using Web query logs, together with a thorough comparison between the uses of implicit and explicit links in Web page classification. Experimental results demonstrated on a large dataset confirm that the use of the implicit links is better than using explicit links in classification performance, with an increase of more than 10.5

$G_1)$ ”Graph based Text Classification: Learn from Your Neighbors”[1], this paper presents a new method for graph-based classification, with particular emphasis on hyperlinked text documents but broader applicability. Its approach is based on iterative relaxation labeling and can be combined with either Bayesian or SVM classifiers on the feature spaces of the given data items. The graph neighborhood is taken into consideration to exploit locality patterns while at the same time avoiding overfitting.

$H_1)$ In the article, ”Web Page Classification with Heterogeneous Data Fusion”[29], the contextual and structural information, of web pages has been represented into a common format of kernel matrix, via a kernel function. A generalized similarity measure between a pair of web pages is proposed. The experimental results on a collection of the ODP database validate the advantages of the proposed method over traditional methods based on any single data source and the uniformly weighted combination of them.

We have compared our method with other hypertext classification algorithms mentioned above. Results, in terms of precision, recall, micro-F_1 and macro-F_1 of A_1, B_1, C_1, D_1, E_1, F_1, G_1, H_1 and TSM has been reported in tables 11, 12, 13 and 14 respectively. It can be observed that performance of TSM is better than others in terms of the measures, except for H_1 on Dmoz and WebKB data sets, for the measure macro-F_1. Out of the set of 128 labeled results for other methods (8 methods, 4 data sets and 4 measures; $8 * 4 * 4 = 128$), 126 results indicate that the proposed method worked well.

Table 11. Comparison of TSM with other hypertext classification methods in terms of precision.

DATA SET	A_1	B_1	C_1	D_1	E_1	F_1	G_1	H_1	TSM
DMOZ	86.40	87.79	95.11	87.62	93.17	95.98	89.70	95.32	96.44
LOOKSMART	91.82	86.15	89.81	87.27	87.78	88.90	93.55	92.02	96.97
WEBKB	87.00	89.90	86.43	93.50	93.18	87.39	94.67	90.72	95.79
YAHOO	82.34	85.78	83.51	84.68	86.12	89.72	88.12	88.24	91.97

Table 12. Comparison of TSM with other hypertext classification methods in terms of recall.

DATA SET	A_1	B_1	C_1	D_1	E_1	F_1	G_1	H_1	TSM
DMOZ	82.62	86.78	90.18	84.21	83.95	84.71	83.89	90.15	91.36
LOOKSMART	83.44	88.56	84.62	82.11	91.26	91.29	83.26	88.67	92.20
WEBKB	80.00	81.38	83.61	84.33	83.26	83.46	84.91	85.81	90.70
YAHOO	80.50	85.54	83.61	86.11	81.56	87.13	81.42	83.14	88.18

Table 13. Comparison of TSM with other hypertext classification methods in terms of micro average F_1.

DATA SET	A_1	B_1	C_1	D_1	E_1	F_1	G_1	H_1	TSM
DMOZ	84.46	87.28	92.57	85.88	88.32	89.99	86.69	92.66	93.83
LOOKSMART	87.42	87.33	87.13	84.61	89.48	90.07	88.10	90.31	94.52
WEBKB	83.35	85.42	84.99	88.67	87.94	85.37	89.52	88.19	93.17
YAHOO	81.40	85.65	83.55	85.38	83.77	88.40	84.63	85.61	90.03

Table 14. Comparison of TSM with other hypertext classification methods in terms of macro average F_1.

DATA SET	A_1	B_1	C_1	D_1	E_1	F_1	G_1	H_1	TSM
DMOZ	81.26	85.78	87.82	82.43	83.28	87.99	85.32	90.04	89.41
LOOKSMART	86.09	85.45	83.25	83.87	87.69	88.89	87.63	88.74	92.57
WEBKB	81.00	84.32	81.45	86.46	85.16	84.81	86.23	91.85	87.86
YAHOO	80.70	83.17	81.77	83.45	81.69	84.93	83.34	88.17	88.02

5.7. Experimental results of TSM using different similarity measures in the different components of tensor.

We have compared the performance of tensor framework using semantic similarity based distance measure for some selected components. Note that, semantic similarity based distance measure is generally used to measure similarity between titles or URLs, because these components contents minimum number of terms and semantic similarity based technique extends the term space using semantic network [13]. Tensor similarity between two tensor corresponding to two hypertext documents is computed using semantic similarity for some selected components and cosine similarity for other components. In our experiment we have considered title, URL and anchor text for computing semantic similarity. K-means clustering and k-NN classification have been performed on four data sets using these newly obtained similarity values. Details regarding the selection of components for computing semantic similarity or cosine similarity are stated below.

$A_2)$ Here K-means clustering on TSM is considered using cosine similarity in the all components of a tensor.

$B_2)$ Here K-means clustering on TSM is considered using semantic similarity in URL and cosine similarity in all other components of a tensor.

$C_2)$ Here K-means clustering on TSM is considered using semantic similarity in URL and title, and cosine similarity in all other components of a tensor.

$D_2)$ Here K-means clustering on TSM is considered using semantic similarity in URL, anchor and title, and cosine similarity in all other components of a tensor.

$E_2)$ Here k-NN classification on TSM is considered using cosine similarity in the all components of a tensor.

$F_2)$ Here k-NN classification on TSM is considered using semantic similarity in URL and cosine similarity in all other components of a tensor.

G_2) Here k-NN classification on TSM is considered using semantic similarity in URL and title, and cosine similarity in all other components of a tensor.

H_2) Here k-NN classification on TSM is considered using semantic similarity in URL, anchor and title, and cosine similarity in all other components of a tensor.

We have studied the advantage of using semantic measure, for some components of tensor, instead of using usual cosine similarity measure for all components. Results on precision, recall, micro-F_1 and macro-F_1 of A_2, B_2, C_2, D_2, E_2, F_2, G_2 and H_2 has been reported in tables 15, 16, 17, 18 respectively. It can be observed from the tables that using semantic similarity for URL and title generally provides better results of classification and clustering for all the data sets.

Table 15. Experimental results of clustering and classification using different similarity measures in the different components of a tensor in terms of precision.

DATA SET	A_2	B_2	C_2	D_2	E_2	F_2	G_2	H_2
DMOZ	61.86	62.30	62.99	62.28	96.44	97.35	97.16	96.94
LOOKSMART	76.41	76.62	77.39	76.72	96.97	97.53	97.47	97.14
WEBKB	56.10	56.33	58.29	58.56	95.79	96.01	96.57	96.85
YAHOO	69.79	70.70	70.93	69.82	91.97	92.34	92.36	92.24

Table 16. Experimental results of clustering and classification using different similarity measures in the different components of a tensor in terms of recall.

DATA SET	A_2	B_2	C_2	D_2	E_2	F_2	G_2	H_2
DMOZ	54.68	55.02	55.12	55.86	86.79	87.43	87.61	87.65
LOOKSMART	68.08	69.12	69.17	69.80	87.89	87.98	88.22	88.54
WEBKB	51.81	52.50	52.87	52.50	86.13	86.45	86.22	86.81
YAHOO	64.43	64.09	64.59	64.74	84.69	84.87	85.06	85.37

Table 17. Experimental results of clustering and classification using different similarity measures in the different components of a tensor in terms of micro average F_1.

DATA SET	A_2	B_2	C_2	D_2	E_2	F_2	G_2	H_2
DMOZ	58.04	58.43	58.79	58.89	91.36	92.12	92.13	92.06
LOOKSMART	72.00	72.67	73.04	73.09	92.20	92.50	92.61	92.64
WEBKB	53.86	54.34	55.44	55.36	90.70	90.97	91.10	91.55
YAHOO	67.00	67.23	67.61	67.18	88.17	88.44	88.55	88.67

Table 18. Experimental results of clustering and classification using different similarity measures in the different components of a tensor in terms of macro average F_1.

DATA SET	A_2	B_2	C_2	D_2	E_2	F_2	G_2	H_2
DMOZ	56.33	57.25	57.08	58.46	89.61	91.41	90.52	90.94
LOOKSMART	71.31	72.43	72.50	72.14	90.57	91.30	92.47	92.04
WEBKB	50.22	52.55	53.58	53.92	87.86	88.57	90.09	90.85
YAHOO	63.84	64.68	64.54	64.77	86.02	87.65	88.20	88.24

6. Conclusion

We proposed the tensor framework for representing hypertext documents. The proposed model consists of a sixth order tensor for each hypertext document and a vector space for each of the different types of feature, the tensor is defined on the product of these vector spaces. In this representation the features extracted from URL or Title is assigned in different vector spaces. A tensor similarity measure is also defined in this article, which computes component wise similarity between hypertexts and sums up the similarities to obtain the similarity between two tensors. Our results provide evidence that tensor based model is very efficient for clustering and classification of hypertext documents compared to traditional vector based models.

Acknowledgments

The authors would like to thank the Department of Science and Technology, Government of India, for funding the Center for Soft Computing Research: A National Facility. This paper was done when one of the authors, S. K. Pal, was a J.C. Bose Fellow of the Government of India.

References

[1] Ralitsa Angelova and Gerhard Weikum. Graph-based text classification: learn from your neighbors. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 485–492, New York, NY, USA, 2006. ACM.

[2] Steffen Bickel and Tobias Scheffer. Multi-view clustering. In *ICDM '04: Proceedings of the Fourth IEEE International Conference on Data Mining*, pages 19–26, Washington, DC, USA, 2004. IEEE Computer Society.

[3] A. I. Borisenko and I. E. Tarapov. *Vector and Tensor Analysis with Applications*. Dover Publications, 1979.

[4] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[5] Deng Cai, Xiaofei He, , and Jiawei Han. Beyond streams and graphs: Dynamic tensor analysis. In *International Conference on Knowledge Discovery and Data Mining (SIGKDD'06)*, pages 374 – 383, New York, NY, USA, 2006. ACM.

[6] Deng Cai, Xiaofei He, , and Jiawei Han. Tensor space model for document analysis. In *Proceedings of ACM SIGIR06 conference*, pages 625 – 626, New York, NY, USA, 2006. ACM.

[7] Soumen Chakrabarti, Byron Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pages 307–318, New York, NY, USA, 1998. ACM.

[8] Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: a new approach to topic-specific Web resource discovery. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1623–1640, 1999.

[9] W. Cohen. Improving a page classifier with anchor extraction and link analysis, 2002.

[10] Zifeng Cui, Baowen Xu, Weifeng Zhang, and Junling Xu. Web documents clustering with interest links. In *SOSE '05: Proceedings of the IEEE International Workshop*, pages 119–124, Washington, DC, USA, 2005. IEEE Computer Society.

[11] J. Furnkranz. *Web mining*. The Data Mining and Knowledge Discovery Handbook, pages 899– 920. Springer, 2005.

[12] X. He, H. Zha, C. Ding, and H. Simon. Web document clustering using hyperlink structures, 2001.

[13] A. Hotho, S. Staab, and G. Stumme. Explaining text clustering results using semantic structures, 2003.

[14] Jingyu Hou and Yanchun Zhang. Utilizing hyperlink transitivity to improve web page clustering. In *ADC '03: Proceedings of the 14th Australasian database conference*, pages 49–57, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[15] Min-Yen Kan and Hoang Oanh Nguyen Thi. Fast webpage classification using url features. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 325–326, New York, NY, USA, 2005. ACM.

[16] Tamara G. Kolda, Brett W. Bader, and Joseph P. Kenny. Higher-order web link analysis using multilinear algebra. In *International Conference on Data Mining*. IEEE press, 2005.

[17] Ning Liu, Benyu Zhang, Jun Yan, Zheng Chen, Wenyin Liu, Fengshan Bai, and Leefeng Chien. Text representation: From vector to tensor. In *International Conference on Data Mining*, Lecture Notes in Computer Science. IEEE Computer Society, 2005.

[18] A. McCallum and K. Nigam. A comparison of event models for naive bayes text classification. In *In AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[19] J. Neville and D. Jensen. Iterative classification in relational data. In *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pages 13–20. AAAI Press, 2000.

[20] Spyridon Plakias and Efstathios Stamatatos. Tensor space models for authorship identification. In John Darzentas, George A. Vouros, Spyros Vosinakis, and Argyris Arnellos, editors, *SETN*, Lecture Notes in Computer Science, pages 239–249. Springer, 2008.

[21] Philip Resnik. Signal processing based on multilinear algebra. *PhD thesis, Katholieke, University of Leuven, Belgium*, 1997.

[22] Suman Saha, C. A. Murthy, and Sankar K. Pal. Classification of web services using tensor space model and rough ensemble classifier. In Aijun An, Stan Matwin, Zbigniew W. Ras, and Dominik Slezak, editors, *ISMIS*, volume 4994 of *Lecture Notes in Computer Science*, pages 508–513. Springer, 2008.

[23] Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng Chen. A comparison of implicit and explicit links for web page classification. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 643–650, New York, NY, USA, 2006. ACM.

[24] Herve Utard and Johannes Furnkranz. Link-local features for hypertext classification. In *EWMF/KDO*, volume 4289 of *Lecture Notes in Computer Science*, pages 51–64. Springer, 2005.

[25] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.

[26] M. A. O. Vasilescu and D. Terzopoulos. Multilinear analysis of image ensembles: Tensorfaces. In *In ECCV*, 2002.

[27] Jidong Wang, Hua-Jun Zeng, Zheng Chen, Hongjun Lu, Li Tao, and Wei-Ying Ma. Recom: reinforcement clustering of multi-type interrelated data objects. In *SIGIR*, pages 274–281, 2003.

[28] S. K. M. Wong and Vijay V. Raghavan. Vector space model of information retrieval: a reevaluation. In *Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 167–185, Swinton, UK, 1984. British Computer Society.

[29] Zenglin Xu, Irwin King, and Michael R. Lyu. Web page classification with heterogeneous data fusion. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 1171–1172, New York, NY, USA, 2007. ACM.

[30] Yiming Yang, Sean Slattery, and Rayid Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2-3):219–241, 2002.

[31] Yiming Yang, Seán Slattery, and Rayid Ghani. A study of approaches to hypertext categorization. *J. Intell. Inf. Syst.*, 18(2-3):219–241, 2002.

[32] Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. In *SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia*, pages 46–54. ACM, 1998.

[33] Xiaojun Zong, Yi Shen, and Xiaoxin Liao. Improvement of hits for topic-specific web crawler. In *Advances in Intelligent Computing, Lecture Notes in Computer Science, Springer Berlin*, pages 524–532, September 16 2005.