A novel split and merge technique for hypertext classification

Suman Saha, C. A. Murthy and Sankar K. Pal
{ssaha_r, murthy, sankar}@isical.ac.in

Center for Soft Computing Research, Indian Statistical Institute

Abstract. As web grows at an increasing speed, hypertext classification is becoming a necessity. While the literature on text categorization is quite mature, the issue of utilizing hypertext structure and hyperlinks has been relatively unexplored. In this paper, we introduce a novel split and merge technique for classification of hypertext documents. The splitting process is performed at the feature level by representing the hypertext features in a tensor space model. We exploit the local-structure and neighborhood recommendation encapsulated in the this representation model. The merging process is performed on multiple classifications obtained from split representation. A meta level decision system is formed by obtaining predictions of base level classifiers trained on different components of the tensor and actual category of the hypertext document. These individual predictions for each component of the tensor are subsequently combined to a final prediction using rough set based ensemble classifiers. Experimental results of classification obtained by using our method is marginally better than other existing hypertext classification techniques.

keywords: Hypertext classification, tensor space model, rough ensemble classifier

1 Introduction

As the web is expanding, where most web pages are connected with hyperlinks, the role of automatic categorization of hypertext is becoming more and more important. The challange of retrieval engine is, it need to search and retrieve toolarge number of web pages. By categorizing documents a priori, the search space can be reduced dramatically and the quality of ad-hoc retrieval improved. Besides, web users often prefer navigating through search directories as in portal sites.

Vector Space Model (VSM), the footstone of many web mining and information retrieval techniques [1], is used to represent the text documents and define the similarity among them. Bag of word (BOW) [2] 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. For web page categorization, a frequently used approach is to use hyperlink information, which improves categorization accuracy [3]. 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 exist many articles using different kinds of features (URL, anchortext, meta-tags, neighborhood, etc.....), but finally they are represented in a single vector, thereby losing 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. Based on the assumption that each source of information provides a different viewpoint, a combination has the potential to have better knowledge than any single method. Methods utilizing different sources of information are combined to achieve further improvement, especially when the information considered is orthogonal. In web classification, combining link and content information is quite popular. A common way to combine multiple information is to treat information from different sources as different (usually disjoint) feature sets, on which multiple classifiers are trained. After that, these classifiers are combined together to generate the final decision.

In this article we have proposed a novel split and merge classification of hypertext documents. The splitting process relies on different types of features, which are extracted from a hypertext document and its neighbors. The split features are represented in a tensor space model, which consists of a sixth order tensor for each hypertext document that is a different vector for each of the different types of features. In this representation the features extracted from URL or Title or any other part are assigned to different tensor components (vector). Note that, unlike a matrix, components of a tensor differ in size, and feature space of each tensor component may be different from others. This representation model does not ignore the information about internal markup structure and link structure of the hypertext documents. In each tensor component a base level classification has been performed. The different types of classifications have been combined using rough set based ensemble classifier [4]. Our experiments demonstrate that splitting the feature set based on structure improves the performance of a learning classifier. By combining different classifiers it is possible to improve the performance even further.

In order to realize the specified objectives, the features of hypertext documents are discussed in section 2. Sections 3 and 4 present tensor space model and rough set based ensemble classifier respectively. Section 5 covers the proposed methodology. Finally, the experimental results are reported in section 6.

2 Hypertext features

A hypertext document consists of different types of features which are found to be useful for representing a web page [5]. Written in HTML, web pages contain additional information other than text content, such as HTML tags, hyperlinks 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.

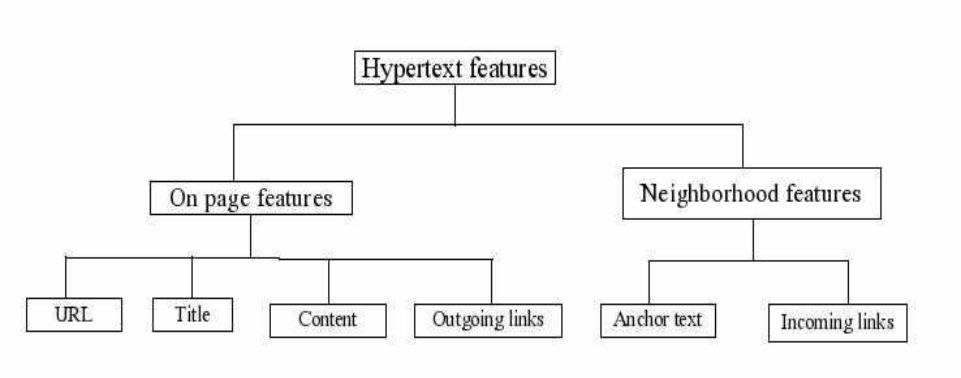


Fig. 1. Different type of features of hypertext document

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 locators (URLs), which mark the address of a resource on the world wide web, provides valuable information about the document and can be used to predict the category of the resource [6]. 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 [7].
3) Text content: The text on a page is the most relevant component for categorization. However, due to a variety of uncontrolled noise 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, vector of features, N-gram representation, which includes not only single terms, but also up to 5 consecutive words [2]. 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 [2].
4) Title and headers: Title and headers can be the 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 [8].
5) In-links: 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 [9]. Researchers have developed several link-analysis algorithms over the past few years. The most popular link-based web analysis algorithms include PageRank [10] and HITS [11].
6) Out-links: 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 modelling dependencies among neighbors and utilizing known class labels [8]. Such explanations may also apply to why web page classification benefits from utilizing features of neighbors.

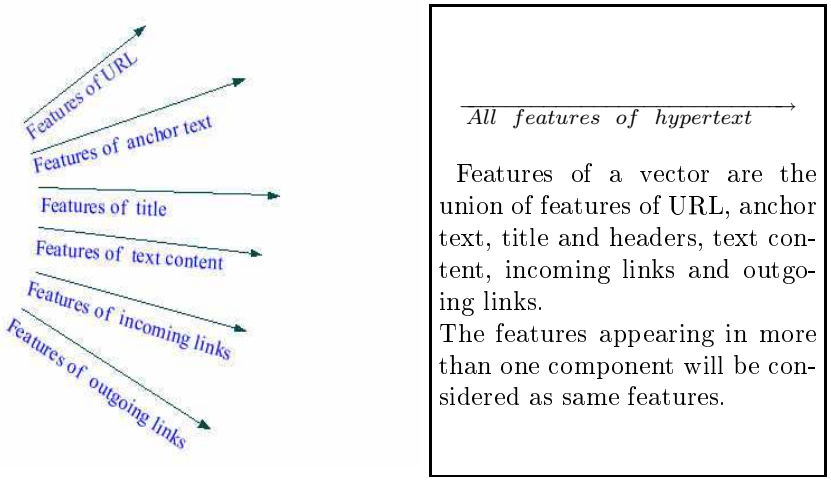
3 Split representation of hypertexts in tensor space model

Tensors provide a natural and concise mathematical framework for formulating and solving problems in high dimensional space analysis [12].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. [13–19].

An n-order tensor in m-dimensional space is a mathematical object that has n indices and each ranges from 1 to m, i.e., each index of a tensor ranges over the number of dimensions of space. 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) to an arbitrary number of indices.

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.



(a) TSM (b) VSM

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

3.1 Mathematical formulation of TSM

Let $\mathcal{H}$ be any hypertext document. Let, $S^{\mathcal{H}u} = \{e_1^{\mathcal{H}u}, e_2^{\mathcal{H}u}, \ldots, e_{n_1}^{\mathcal{H}u}\}$ be the set corresponding to features of URL, $S^{\mathcal{H}a} = \{e_1^{\mathcal{H}a}, e_2^{\mathcal{H}a}, \ldots, e_{n_2}^{\mathcal{H}a}\}$ be the set corresponding to features of anchor text, $S^{\mathcal{H}h} = \{e_1^{\mathcal{H}h}, e_2^{\mathcal{H}h}, \ldots, e_{n_3}^{\mathcal{H}h}\}$ be the set corresponding to features of title and headers, $S^{\mathcal{H}c} = \{e_1^{\mathcal{H}c}, e_2^{\mathcal{H}c}, \ldots, e_{n_4}^{\mathcal{H}c}\}$ be the set corresponding to features of text content, $S^{\mathcal{H}out} = \{e_1^{\mathcal{H}out}, e_2^{\mathcal{H}out}, \ldots, e_{n_5}^{\mathcal{H}out}\}$ be the set corresponding to features of outgoing links and $S^{\mathcal{H}in} = \{e_1^{\mathcal{H}in}, e_2^{\mathcal{H}in}, \ldots, e_{n_6}^{\mathcal{H}in}\}$ be the set corresponding to features of incoming links.

Let $S^{\mathcal{H}}$ be the set representing all features of $\mathcal{H}$ in a vector space, $\mathcal{V}$. Then, $S^{\mathcal{H}} = S^{\mathcal{H}u} \cup S^{\mathcal{H}a} \cup S^{\mathcal{H}h} \cup S^{\mathcal{H}c} \cup S^{\mathcal{H}out} \cup S^{\mathcal{H}in}$. Let $S_1^{\mathcal{H}}$ be the set of features which are present in more than one component. So, $S_1^{\mathcal{H}} = \cup_{(x,y \in F) \& x \neq y} S^{\mathcal{H}x} \cap S^{\mathcal{H}y}$, where, $F = \{u, a, h, c, out, in\}$. Note that, features present in more than two components is already considered in the above expression. Let s be an element of $S_1^{\mathcal{H}}$. That is s has occurred in more than one component of the hypertext documents. For each appearance of s in different components, s may have different significance regarding the categorization of the hypertext documents. Now the multiple appearance of s is ignored in $S^{\mathcal{H}}$, as it is a set of union of the sets corresponding to the components of hypertext.

In the vector space model for hypertext representation, vectors are constructed on $S^{\mathcal{H}}$, that is, occurrence of $s \in S_1^{\mathcal{H}}$ in different components is ignored. In some advanced vector space models elements of different components are tagged [8], that is $S_1^{\mathcal{H}} = \phi$. Let $|.|$ denote cardinality of a set. Number of features of different components may have a large variance value. For example, $|S^{\mathcal{H}u}| << |S^{\mathcal{H}c}|$. In this representation, importance of the elements corresponding to the components with low cardinality, is ignored during magnitude normalization.

In tensor space model the features corresponding to different components of hypertext are represented as different components of a tensor. Let $\mathcal{T}$ be the tensor space corresponding to hypertext documents. Each member T of $\mathcal{T}$ is of the form $T = T_{xi}$ where, $x \in F$ and $1 \leq i \leq |S^{\mathcal{H}x}|$, i.e. the value of T at (x, i) is $e_i^{\mathcal{H}x}$. Note that i depends on x, so it is not just a matrix.

Similarity measures on TSM Cosine similarity is a measure of similarity between two vectors of n dimensions by finding the angle between them, 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 A, B , represented 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.

Let $\mathcal{T}$ be the tensor space corresponding to hypertext documents. Each member T of $\mathcal{T}$ is of the form $T = T_{rs}$ where r ranges on the types of features considered and s ranges on number of terms extracted of particular types. The tensor similarity between two tensors T_i and T_j of $\mathcal{T}$ is defined as $sim(T_i, T_j) =$

$\sum_r sim(T_i r, T_j r)$, where $sim(T_i r, T_j r)$ is the similarity between r^{th} component of T_i and T_j. Now, for each r, the r^{th} components of a tensor T_i is a vector. So, $sim(T_i r, T_j r)$ is basically the similarity between two vectors. Note that, here, cosine similarity is considered as vector similarity measure.

Computational complexity on TSM Let n be the total number of features of hypertext documents. 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 holds.

Theorem: Computational complexity of an algorithm performing on tensor space model 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.

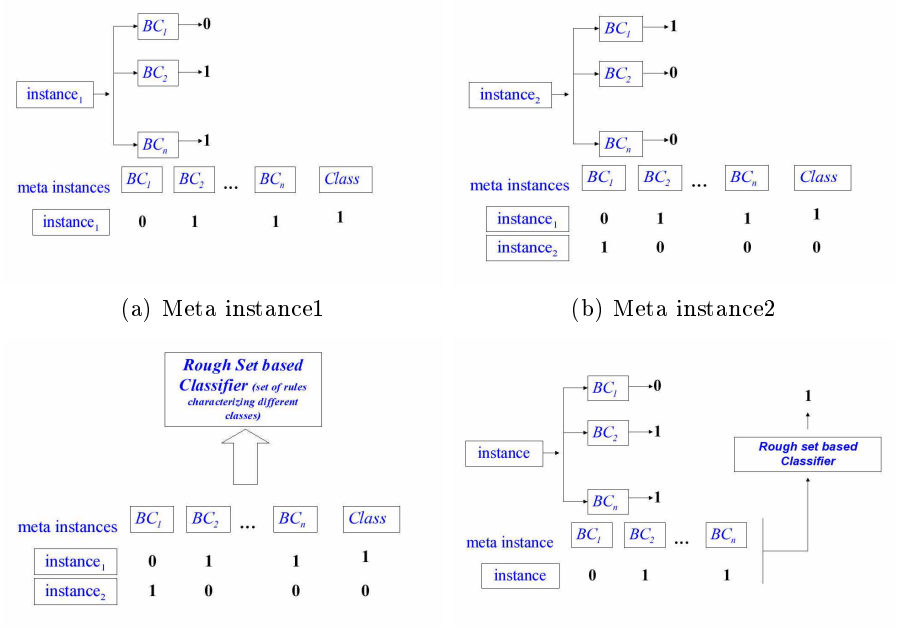
4 Merging classifications using rough ensemble classifier

The rough ensemble classifier (REC) is designed to extract decision rules from trained classifier ensembles that perform classification tasks [4]. REC utilizes trained ensembles to generate a number of instances consisting of prediction of individual classifiers as conditional attribute values and actual classes as decision attribute values. Then a decision table is constructed using all the instances with one instance in each row. Once the decision table is constructed, rough set attribute reduction is performed to determine core and minimal reducts. The classifiers corresponding to a minimal reduct are then taken to form classifier ensemble for REC classification system. From the minimal reduct, the decision rules are computed by finding mapping between decision attribute and conditional attributes. These decision rules obtained by rough set technique are then used to perform classification tasks (Fig.3). Following theorems exists in this regard.

- **Theorem 1:** Rough set based combination is an optimal classifier combination technique [4].
- **Theorem 2:**The performance of the rough set based ensemble classifier is at least same as every one of its constituent single classifiers [4].

5 Split and merge classification of hypertexts

Here we propose a split and merge classification of hypertexts. A hypertext document is split on the basis of the different types of features existing in it. Tensor



(a) Meta instance1 (b) Meta instance2

(c) Training REC (d) Testing REC

Fig. 3. Example describing different steps of REC. Subfigures (a) and (b) show the classification of instances by different base classifiers (denoted as BC). Outputs of the base classifiers along with the actual class have been considered to construct meta data. Sub figure (c) show the training of REC. Subfigure (d) show the output of REC.

space model has been used to represent the hypertext using the information of text content, internal mark-up structure and link structure. Classification of hypertext documents, represented as tensor, can be obtained in two ways: (1) by integrating classifier's parameters of different tensor components and (2) by integrating classifiers output obtained on different tensor components. In the first way, a K-NN classification has been performed using tensor similarity measure. In the second way ensemble classification has been performed (Fig. 4). For ensemble classification, base level classification has been carried out on individual tensor components and combined classification has been obtained using rough set based ensemble classifier.

5.1 Preprocessing for split representation

Hypertext documents are tokenized with syntactic rules and canonical forms. First we select a set of relevant features from a HTML document. For each type of feature an individual tensor component is constructed. A tensor component is

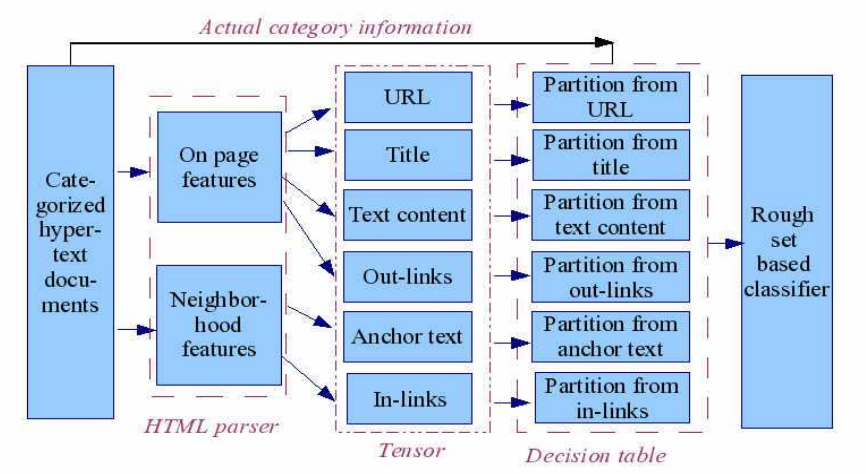


Fig. 4. Block diagram of proposed method.

a vector, which represents the terms of particular type corresponding to the component. Note that the tensor space model captures the structural representation of hypertext document.

1) Preprocessing text content:

- The text is stemmed using Porter's stemming algorithm and stop words are removed.
- Unique words present in the text are represented as a tensor component. This tensor component corresponds to the text content of the hypertext documents.

2) Preprocessing URL:

- 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.
- These segmented substrings are treated as words. All these words found in a URL will be represented as a tensor component corresponding to features of URLs.

3) Preprocessing anchor text:

- Anchor text is a small text content. The text is stemmed using Porter's stemming algorithm and stop words are removed.
- This is computed in the same way as text content, except substituting each document by a virtual document consisting of all the anchor text inside that document.

– Unique words present in this virtual document are represented as a tensor component corresponding to features of anchor text.

4) Preprocessing of title and headers:

– Title and headers are text contents. The text is stemmed using Porter's stemming algorithm and stop words are removed.
– Unique words present in these text are represented as a tensor component corresponding to features of title and headers.

5) Preprocessing in-links:

– 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 segmentation of the in-links are stored in a tensor component corresponding to features of in-links.

6) Preprocessing out-links:

– All out-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 out-links are stored in a tensor component corresponding to features of out-links.

The above methodologies are applied on split hypertext documents. The merging will take place during classification, which is described below.

5.2 Merging of classifications

We now describe how to merge the classifications obtained from each one of the components of the tensor using naive bayes as base level classifiers. To generate the initial classifications for rough ensemble classifier, we assume a base level classifier and train it on each tensor component. These trained classifiers provide different classifications on the tensor space. Outputs of the base classifiers and the actual class information are used to construct meta level decision table. Output of a base level classifier contributes to the existence of an attribute, values of this attribute can be any class label that is determined by the base classifier corresponding to the tensor component. This meta data represented in the form of decision table is the input of rough set based ensemble classifier and its output is the merged classification. So the number of attributes is the same as number of tensor components. Rough set based attribute reduction techniques eliminate superfluous attributes and create a minimal sufficient subset of attributes for a decision table. Such minimal sufficient subset of attributes is called a reduct. Once the reduct is computed we remove redundant classifiers from the ensemble and construct new reduced decision table. Rough set based decision rules extracted from this reduced decision table are applied to obtain final classification. These decision rules perform merging of the base level decisions into a final decision.

6 Experimental Results

We performed a large number of experiments to test the output of proposed methods. We now describe the data corpuses, methodologies and results.

6.1 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.

(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
S_1*	51399

*S_1 includes the features appearing in at least two components

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

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.

(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
S_1*	30900

*S_1 includes the features appearing in at least two components

Table 2. Class distribution and features of the looksmart 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

*S_1 includes the features appearing in at least two components

(b)

Components	# features
URL	12898
Anchor	10515
Title	16193
Text	23582
In-link	14529
Out-link	14094
Total	91811
Union	72071
S_1*	19740

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

Another data set consists of 40000 web pages crawled from the Yahoo topic directory (http://dir.yahoo.com). This is a large hypertext corpous, 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 category of the Yahoo data set is given in the Table 4.

We processed the data sets to remove images and scripts followed by stopwords removal and stemming. Link graph has been constructed for each of the datasets for extracting neighborhood features. URLs have been segmented for

(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
NewsMedia	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
SocialScience	2859	8.59	3493	6.30
SocietyCulture	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
S_1*	65560

*S_1 includes the features appearing in at least two components

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

extracting URL features. Finally features extracted from all the components of hypertext have been represented using both the models (i.e., tensor space model and vector space model) in our experiments. We have considered vector space model for the purpose of comparison.

6.2 Evaluation Measure

We have employed the standard measures to evaluate the performance of hypertext classification (i.e. precision, recall and F_1 measures). Precision (P) is the proportion of actual positive class members returned by the system among all positive class members. 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:

$$F_1 = \frac{2PR}{P+R}$$

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 classification performance.

6.3 Classification results on TSM

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 tensor space model for hypertext representation, we have constructed two k-NN classifiers. In the first case, k-NN classification on vector space representation for hypertext document is considered and vector similarity measure is used to compute nearest neighbor. In the second case, k-NN classification on tensor space model for hypertext representation is considered and tensor similarity measure is used to compute nearest neighbour. The performances of these two classifiers have been observed on four different datasets, Yahoo, webKB, Looksmart and Dmoz. The classification results of comparisons are shown in tables 4(a), 4(b), 4(c) and 4(d). The results has been shown in terms of Precession, recall, micro-average-F_1 and macro-average-F_1. It can be observed from the tables that classification results are better when tensor space model for hypertext representation is considered compared to classification results when vector space model for representation is considered.

(a)

Data set	VSM	TSM	Better?
Dmoz	92.94	95.07	√
Looksmart	90.13	93.55	√
WebKB	91.85	94.67	√
Yahoo	88.24	89.12	√

(b)

Data set	VSM	TSM	Better?
Dmoz	83.27	88.89	√
Looksmart	87.36	90.26	√
WebKB	85.80	84.91	√
Yahoo	82.36	86.42	√

(c)

Data set	VSM	TSM	Better?
Dmoz	87.83	91.87	√
Looksmart	88.72	91.87	√
WebKB	88.72	89.52	√
Yahoo	85.19	87.74	√

(d)

Data set	VSM	TSM	Better?
Dmoz	85.69	89.32	√
Looksmart	83.18	87.63	√
WebKB	84.28	87.23	√
Yahoo	84.50	86.34	√

Table 5. Results of k-NN classification on VSM and TSM

6.4 Classification results on individual components and combined results.

In this subsection we have provided the results of experiments regarding classifications of hypertext documents. Classifications of hypertext have been performed on different components of tensor space model corresponding to different types of feature sets using naive bayes classifier. Here naive bayes classifier is used for its simplicity in implementation. We have also provided the results of classification with tensor space model using k-NN classifier where tensor similarity measure

is distance. In addition, the combined results of classification are provided using rough set based ensemble classifier. The cases considered are given below.

A) Classification based on URL features (2, 5.1) using naive bayes classifier.

B) Classification based on Anchor text features (2, 5.1) using naive bayes classifier.

C) Classification based on features of Title and headers (2, 5.1) using naive bayes classifier.

D) Classification based on features of Text content (2, 5.1) using naive bayes classifier.

E) Classification based on features of In-coming links (2, 5.1) using naive bayes classifier.

F) Classification based on features of Out-going links (2, 5.1) using naive bayes classifier.

G) Classification based on Tensor similarity measure (6.3) using using k-NN classifier.

H) Classification based on split marge classification (5.2).

Results on precision, recall, micro-F_1 and macro-F_1 of A, B, C, D, E, F, G and H have been reported in tables 6, 7, 8, 9 respectively. It can be observed that classification results are poor for link based features than the text based features, and combined results corresponding to proposed methods are far better.

Table 6. Classification results on individual components and their rough set based combination in terms of precision.

Data set	A	B	C	D	E	F	G	H
Dmoz	62.01	70.30	76.61	81.96	66.37	68.42	95.07	95.32
Looksmart	67.50	70.11	74.51	82.64	62.29	61.28	93.55	94.02
WebKB	64.90	72.02	69.92	86.16	67.30	61.71	94.67	95.72
Yahoo	61.64	68.98	67.93	79.43	59.93	61.38	89.12	90.24

Table 7. Classification results on individual components and their rough set based combination in terms of recall.

Data set	A	B	C	D	E	F	G	H
Dmoz	57.83	71.56	68.95	79.59	56.58	57.85	88.89	90.15
Looksmart	60.04	73.32	69.57	77.63	60.84	64.04	90.26	91.67
WebKB	56.97	65.48	67.75	79.27	54.72	56.53	84.91	85.81
Yahoo	57.90	67.51	66.86	78.85	54.41	54.39	86.42	87.14

Table 8. Classification results on individual components and their rough set based combination in terms of micro average F_1.

Data set	A	B	C	D	E	F	G	H
Dmoz	59.85	70.92	72.57	80.76	61.08	62.69	91.87	92.66
Looksmart	63.55	71.68	71.96	80.06	61.56	62.63	91.87	92.83
WebKB	60.68	68.60	68.82	82.57	60.36	59.01	89.52	90.49
Yahoo	59.71	68.24	67.39	79.14	57.04	57.67	87.74	88.66

Table 9. Classification results on individual components and their rough set based combination in terms of macro average F_1.

Data set	A	B	C	D	E	F	G	H
Dmoz	59.79	70.72	72.71	79.95	58.14	59.36	89.32	90.04
Looksmart	62.86	71.89	67.11	78.28	61.14	61.26	87.63	88.74
WebKB	59.37	69.15	65.68	82.46	58.81	57.71	87.23	91.85
Yahoo	58.53	66.91	65.84	77.50	54.58	58.73	86.34	88.17

6.5 Comparisons with some recent hypertext classification techniques

We have compared the performance of the proposed methods with existing classification techniques. A brief review of existing hypertext classification techniques is given below and these methods are considered for comparisons.

A_1) The article "Enhanced hypertext categorization using hyperlinks"[8], is the first hypertext classification system that combines textual and linkage features into a general statistical model to infer the of interlinked documents. Relaxation labelling technique is used for better classification by exploiting link information in a small neighborhood around documents.

B_1) The article "Improving A Page Classifier with Anchor Extraction and Link Analysis"[20], 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.

C_1) The article "Fast webpage classification using URL features"[6], explores the use of URLs for web page categorization via a two-phase pipeline of word segmentation and classification. This technique quantify its performance against document-based methods, which require the retrieval of the source documents.

D_1) The article, "Link-Local Features for Hypertext Classification"[21], demonstrates that 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.

E_1) The article "Graph based Text Classification: Learn from Your Neighbors"[22], presents a new method for graph-based classification, with particular emphasis on hyperlinked text documents but broader applicability. This approach is based on iterative relaxation labelling 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.

F_1) In the article, "Web Page Classification with Heterogeneous Data Fusion"[23], 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.

G_1) Here a k-NN classifier on tensor space model is considered where tensor similarity measure used is the distance between hypertext documents 6.3.

H_1) Proposed split and merge classification where rough ensemble classifier on tensor space model is considered 5.2.

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 and H_1 have been reported in Tables 10, 11,12 and 13 respectively. It can be observed that performance of the proposed methods (i.e., G_1 and H_1) are better than others in terms of precision, recall, micro-F_1 and macro-F_1. Among G_1 and H_1, H_1 is found to be the better than G_1.

Table 10. Comparison of the eight 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
Dmoz	86.4	87.79	92.11	87.62	93.17	93.98	95.07	95.32
Looksmart	91.82	86.15	89.81	87.27	87.78	88.9	93.55	94.02
WebKB	87	89.9	86.43	90.5	93.18	87.39	94.67	95.72
Yahoo	82.34	83.78	83.51	84.68	86.12	87.72	89.12	90.24

Table 11. Comparison of the eight 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
Dmoz	82.62	86.78	85.18	84.21	83.95	84.71	88.89	90.15
Looksmart	83.44	88.56	84.62	82.11	87.26	91.29	90.26	91.67
WebKB	80	81.38	83.61	84.33	83.26	83.46	84.91	85.81
Yahoo	80.5	82.54	83.61	85.11	81.56	83.13	86.42	87.14

Table 12. Comparison of the eight 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
Dmoz	84.46	87.28	88.50	85.88	88.32	89.10	91.87	92.66
Looksmart	87.42	87.33	87.13	84.61	87.51	90.07	91.87	92.83
WebKB	83.35	85.42	84.99	87.30	87.94	85.37	89.52	90.49
Yahoo	81.40	83.15	83.55	84.89	83.77	85.36	87.74	88.66

Table 13. Comparison of the eight 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
Dmoz	81.26	85.78	87.82	82.43	83.28	87.99	89.32	90.04
Looksmart	86.09	85.45	83.25	83.87	87.69	88.89	87.63	88.74
WebKB	81	84.32	81.45	86.46	85.16	84.81	87.23	91.85
Yahoo	80.7	83.17	81.77	83.45	81.69	84.93	86.34	88.17

7 Conclusion

We proposed a split and merge classification of hypertext documents. In the split process the hypertext documents is represented in tensor space model. Tensor space model consists of a sixth order tensor for each hypertext document, that is a different vector for each of the different types of features. In this representation, the features extracted from URL or Title are assigned in different tensor components. In the merging process, base level classification has been performed on individual tensor components and combined classification has been obtained by using rough set based ensemble classifier. Two step improvement on the existing classification results of hypertext has been shown. In the first step we achieve better classification results by merging similarity distances. In the second step further improvement of the results has been obtained by merging the output of base level classifiers using rough ensemble classifier. Improvement took place because of the initial split process.

Acknowledgment

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. Wong, S.K.M., Raghavan, V.V.: 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, Swinton, UK, British Computer Society (1984) 167–185
2. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. In: In AAAI-98 Workshop on Learning for Text Categorization. (1998)
3. Yang, Y., Slattery, S., Ghani, R.: A study of approaches to hypertext categorization. Journal of Intelligent Information Systems **18**(2-3) (2002) 219–241
4. Saha, S., Murthy, C.A., Pal, S.K.: Rough set based ensemble classifier for web page classification. Fundamentae Informetica **76**(1-2) (2007) 171–187
5. Furnkranz, J.: Web mining. The Data Mining and Knowledge Discovery Handbook, pages 899– 920. Springer (2005)
6. Kan, M.Y., Thi, H.O.N.: Fast webpage classification using url features. In: CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management, New York, NY, USA, ACM (2005) 325–326
7. Eiron, N., McCurley, K.S.: Analysis of anchor text for web search. In: SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, New York, NY, USA, ACM (2003) 459–460
8. Chakrabarti, S., Dom, B., Indyk, P.: Enhanced hypertext categorization using hyperlinks. In: SIGMOD '98: Proceedings of the 1998 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM (1998) 307–318
9. Chakrabarti, S., Roy, S., Mahesh, V., Soundalgekar: Fast and accurate text classification via multiple linear discriminant projections. The International Journal on Very Large Data Bases **12**(2) (2003) 170–185
10. Brin, S., Page, L.: The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems **30**(1–7) (1998) 107–117
11. Zong, X., Shen, Y., Liao, X.: Improvement of hits for topic-specific web crawler. In: Advances in Intelligent Computing, Lecture Notes in Computer Science, Springer Berlin. (September 16 2005) 524–532
12. Borisenko, A.I., Tarapov, I.E.: Vector and Tensor Analysis with Applications. Dover Publications (1979)
13. Resnik, P.: Signal processing based on multilinear algebra. PhD thesis, Katholieke, University of Leuven, Belgium (1997)
14. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis of image ensembles: Tensorfaces. In: In ECCV. (2002)
15. Kolda, T.G., Bader, B.W., Kenny, J.P.: Higher-order web link analysis using multilinear algebra. In: International Conference on Data Mining, IEEE press (2005)
16. Liu, N., Zhang, B., Yan, J., Chen, Z., Liu, W., Bai, F., Chien, L.: Text representation: From vector to tensor. In: International Conference on Data Mining. Lecture Notes in Computer Science, IEEE Computer Society (2005)
17. Cai, D., He, X., , Han, J.: Tensor space model for document analysis. In: Proceedings of ACM SIGIR06 conference, New York, NY, USA, ACM (2006) 625 – 626
18. Cai, D., He, X., , Han, J.: Beyond streams and graphs: Dynamic tensor analysis. In: International Conference on Knowledge Discovery and Data Mining (SIGKDD'06), New York, NY, USA, ACM (2006) 374 – 383

19. Plakias, S., Stamatatos, E.: Tensor space models for authorship identification. In Darzentas, J., Vouros, G.A., Vosinakis, S., Arnellos, A., eds.: SETN. Lecture Notes in Computer Science, Springer (2008) 239–249
20. Cohen, W.: Improving a page classifier with anchor extraction and link analysis (2002)
21. Utard, H., Furnkranz, J.: Link-local features for hypertext classification. In Ackermann, M., Berendt, B., Grobelnik, M., Hotho, A., Mladenic, D., Semeraro, G., Spiliopoulou, M., Stumme, G., Sv?tek, V., van Someren, M., eds.: EWMF/KDO. Volume 4289 of Lecture Notes in Computer Science., Springer (2005) 51–64
22. Angelova, R., Weikum, G.: 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, New York, NY, USA, ACM (2006) 485–492
23. Xu, Z., King, I., Lyu, M.R.: Web page classification with heterogeneous data fusion. In: WWW '07: Proceedings of the 16th international conference on World Wide Web, New York, NY, USA, ACM (2007) 1171–1172