New use of the HITS algorithm for fast web page classification

Mohamed Nadjib Meadi, Chaouki Babahenini, Abdelmalik Taleb-Ahmed

Turkish Journal of Electrical Engineering and Computer Sciences

Volume 25 | Number 3 Article 32

1-1-2017

New use of the HITS algorithm for fast web page classification

MOHAMED NADJIB MEADI

MOHAMED CHAOUKI BABAHENINI

ABDELMALIK TALEB AHMED

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 2015 – 2032

doi:10.3906/elk-1501-236

New use of the HITS algorithm for fast web page classification

Mohamed Nadjib MEADI[1,*], **Mohamed Chaouki BABAHENINI**[1], **Abdelmalik TALEB AHMED**[2]

[1]Department of Computer Science, LESIA Laboratory, University of Biskra, Biskra, Algeria
[2]LAMIH Laboratory, University of Valenciennes, Valenciennes, France

Received: 31.01.2015 • **Accepted/Published Online:** 18.08.2016 • **Final Version:** 29.05.2017

Abstract: The immense number of documents published on the web requires the utilization of automatic classifiers that allow organizing and obtaining information from these large resources. Typically, automatic web pages classifiers handle millions of web pages, tens of thousands of features, and hundreds of categories. Most of the classifiers use the vector space model to represent the dataset of web pages. The components of each vector are computed using the term frequency inversed document frequency (TFIDF) scheme. Unfortunately, TFIDF-based classifiers face the problem of the large-scale size of input data that leads to a long processing time and an increase in resource requests. Therefore, there is an increasing demand to alleviate these problems by reducing the size of the input data without influencing the classification results. In this paper, we propose a novel approach that improves web page classifiers by reducing the size of the input data (i.e. web pages and feature reduction) by using the hypertext induced topic search (HITS) algorithm. We employ HITS results for weighting remaining features. We evaluate the performance of the proposed approach by comparing it with the TFIDF-based classifier. We demonstrate that our approach significantly reduces the time needed for classification.

Key words: Hypertext induced topic search, link analysis, support vector machine, web mining

1. Introduction

The number of pages published on the World Wide Web is estimated to be in the billions (http://www.internetlivestats.com/total-number-of-websites/). The mining of these pages requires incredible intellectual efforts that exceed human capacities.

Web pages as a whole have no unifying structure; i.e. there is variability of structuring style and content creation is much greater than in traditional collections of textual documents [1]. Therefore, it is impossible to apply dataset management and traditional information retrieval techniques to web pages for information and knowledge extraction. Web mining has emerged as a solution that handles the previous challenges. Its principal objective is to use methods and techniques of data mining to extract knowledge contained in web pages, taking into account their unstructured nature, particularly during preprocessing and feature extraction phases.

The most common models used to represent web pages for classification are the Boolean model, the vector space model, and the probabilistic model [2]. In the literature, the vector space model is widely used. The vector model represents each document as a set of terms. It associates to each term a weight calculated by a given method such as the TFIDF scheme. By the use of this model, a collection of documents is represented as a relational table (or matrix), where each term is an attribute and each weight is an attribute value.

*Correspondence: mmnadjib@yahoo.fr

The support vector machine (SVM) method is also a widely used method for text and web page categorization, because of its high generalization performance and its tolerance ability to process high-dimensional inputs. Although the dimensionality representation is high, each of the document vectors contains only a few nonzero elements and most text categorization problems are linearly separable [3].

Experimentally (see Section 5), we have found that SVM web page classifiers that use the TFIDF scheme have some disadvantages. Among others, we have found that the learning of such classifiers is very time-consuming. In addition, it produces a large set of support vectors, which may slow down the classification of new pages. Therefore, before starting the process of building an automatic web page classifier, we have to solve the problem of the immense size of the input datasets. Thus, efficient mechanisms for dataset reduction and feature selection should be proposed.

In this work, we propose a new approach that aims to construct a web page classifier using the HITS algorithm. First, we use the HITS algorithm to reduce the size of the learning set in both axes, web pages and features, at once. Second, we use the outputs of the HITS algorithm, particularly the authority vector, to calculate the weights of features.

The use of the HITS algorithm in this work is motivated by the ability of HITS to rank pages according to the query topic, which may provide more relevant authority and hub pages. In addition, compared to PageRank, HITS is a general algorithm for calculating authority and hubs in order to rank retrieved data. Moreover, the ranking achieved by HITS may also be combined with information retrieval-based rankings [2]. Hence, resulting HITS ranks may improve the selection of best features and web pages.

The remainder of this paper is organized as follows: Section 2 presents some related works. Related theoretical definitions are given in Section 3. Section 4 introduces the proposed approach. Section 5 describes conducted experiments. The paper ends with a conclusion and some research perspectives.

2. Related work

Dimensionality reduction techniques can be categorized mainly into feature extraction and feature selection [4]. Feature extraction approaches project features in a new feature space with lower dimension and the newly constructed features are usually combinations of original features. Examples of feature extraction techniques include principal component analysis (PCA), linear discriminant analysis (LDA), and canonical correlation analysis (CCA). The feature selection approaches select a small subset of features that minimize redundancy and maximize relevance to the target such as class labels in classification. Representative feature selection techniques include information gain, relief, Fisher score, Lasso, chi-square, document frequency, orthogonal centroid feature selection, comprehensive measurement feature selection, deviation from Poisson feature selection, improved Gini index, and mutual information.

There are three major paradigms of feature selection: filter methods, wrapper methods, and embedded methods. First, filter methods [5] evaluate each feature independently with respect to the class labels in the learning set and determine a ranking of all features, and then the top ranked features are selected.

Second, wrapper methods use classical search methods to find the best features subset, repeatedly evaluating different feature subsets. Wrapper methods have traditionally sought specific combinations of individual features from the power set of features, but this approach scales poorly for a large number of features inherent with classifying text [6].

Finally, embedded methods [6] build a usually linear prediction model that simultaneously tries to maximize the goodness-of-fit of the model and minimizes the number of input features. Some variants build a

classifier using the full dataset and then iteratively remove features until obtaining the minimal set of features without decreasing system accuracy.

In the following, we present some recent works that proposed new methods for feature selection in text classification that they tested on WebKb and Reuters datasets. Roberto et al. proposed a filtering method for feature selection called ALOFT (at least one feature) [7]. This approach ensures that every document in the learning set is represented by at least one feature and the number of selected features is determined in a data-driven way. In [8], the same authors proposed an improved version of the previous approach. In this new version two filtering methods are used for feature selection in text categorization, namely maximum f features per document and maximum f features per document – reduced. Both algorithms determine the number of selected features f in a data-driven way using a global ranking feature evaluation function (FEF), whereas the second algorithm analyzes only the documents with high FEF-valued features to select fewer features, therefore, avoiding unnecessary ones.

Another scheme was proposed in [9] to improve the well-known filter feature selection methods. This scheme weakens the influence of the imbalanced factors occurring in the corpus. In addition, an approach composed of two algorithms, a fair feature subset selection algorithm and an adaptive fuzzy learning network, was proposed by Lee et al. in [10].

Dasgupta et al. proposed an unsupervised feature selection strategy [11]. This strategy gives the worst case theoretical guarantee on the generalization power of the resultant classification function f with respect to the classification function f obtained when keeping all the features. Mladenic et al. in [12] used the linear SVM for feature selection. First, they train the linear SVM on a subset of learning data and retain only features that correspond to "highly weighted" components of the normal to the resulting hyperplane. The reduced feature space is then used to train the classifier over a larger learning set. Tu et al. [13] proposed a particle swarm optimization (PSO) technique to implement a feature selection phase, where the fitness function is implemented using multiclass SVMs of type one-versus-rest (1&R). Moreover, Kim et al. adopted a dimension reduction method that reduces the dimension of document vectors [14]. The authors also introduced decision functions for the centroid-based classification algorithm. Their method uses the SVM classifier to handle the problem where a document may belong to multiple classes.

Chen et al. proposed a fuzzy ranking analysis paradigm together with a new relevance measure called the discriminating power measure, to reduce the input dimension [15]. The first algorithm is used to reduce the dimensionality. This algorithm gives fair treatment to each category and identifies useful features. The second one is used for classification.

Unfortunately, in the case of SVM-based web page classification, the feature selection only has poor influence on the acceleration of classifiers because SVM classifiers support high-dimensional data due to two reasons: 1) the complexity of the SVM classifier (SMO for instance) equates to $O(N^3)$, where N is the number of learning examples; (2) in addition, Joachims in [3] stated that one remarkable property of SVMs is that their ability to learn is independent of the dimensionality of the feature space data. This means that the SVM can generalize even in the presence of a very large set of features. These methods do not provide any enhancement in the overall acceleration of the classification. The aim of our work is to accelerate the classification by simultaneously reducing features and learning examples.

3. Theoretical definitions

3.1. Web mining

Web mining [2] is the application of data mining techniques to discover consistent schemes or models in Internet resources. Web mining aims to discover useful information or knowledge from web hyperlinks, contents, and usage logs (or web logs). Based on the data types used in the process of exploration, the tasks of web mining can be classified into three main types: web structure mining, web content mining, and web usage mining.

Exploring the structure of the web is to discover knowledge from the hyperlinks that represent the structure of the web. Exploring web content aims to extract useful knowledge by exploiting the content of web pages. The web usage mining allows the creation of user access patterns from the web log files, which record the clicks from each user.

3.2. Link analysis

Web pages are connected by hyperlinks, which carry important information. These links often provide an implicit means of transportation of the authority to the pointed pages. Therefore, pages that are pointed at by many other pages are likely to contain reliable information. These links should obviously be used to evaluate the ranking of the page in search engines.

In 1997 and 1998, two search algorithms based on hyperlink analysis were introduced, which are the PageRank [16] and HITS [17] algorithms. These algorithms exploit the hyperlink structure of the web to rank web pages according to their levels of authority.

3.3. HITS

HITS (or the hubs authorities algorithm) is a link analysis algorithm, developed by Jon Kleinberg [17], which helps the rating of web pages. Unlike PageRank, which is a static ranking algorithm, HITS depends entirely on the search query. When the user issues a search query, HITS first expands the list of relevant pages returned by a search engine and produces two rankings of the expanded pages set: the ranking authority and the ranking hub.

An authority is a page with a lot of inbound links (see Figure 1). The idea is that a given page may have good content on a particular topic, so many pages can express their confidence by sending a link to this page. A hub is a page with many outgoing links (see Figure 1). The hub page serves as an organizer of information on a given topic and points to many good authority pages on the particular subject. When a user comes to a hub page, he or she will find many useful links to the best pages on the subject [2,18].

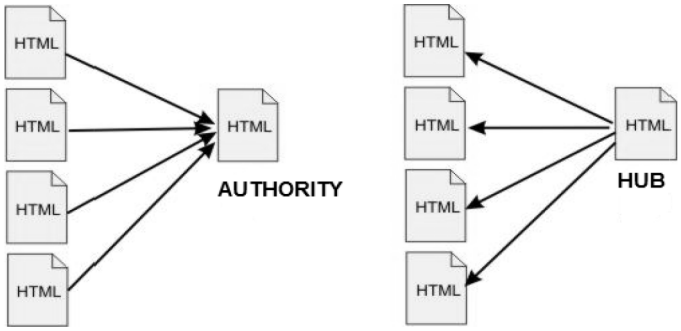

Figure 1. Meaning of hub and authority pages.

Since its emergence, the HITS algorithm has undergone many improvements. For example, Borodin et al. in [19] presented a theoretical framework based on the HITS algorithm for the study of link analysis ranking algorithms. In addition, HITS has been used in different fields. For example, Xiujuan et al. in [20] proposed a credit scoring algorithm based on link analysis ranking with a support vector machine. Their algorithm decides automatically whether a bank should provide a loan to an applicant. Moreover, Deguchi et al. used this algorithm to investigate the economic hubs and authorities of the world trade network from 1992 to 2012 [21].

3.4. Support vector machines

The SVM is based on the principle of structural risk minimization theory of statistical learning [22,23]. The main idea of the SVM is to construct a hyperplane that maximizes the separation margin between positive and negative examples:

$$(x_1, y_1), \ldots (x_i y_i)\}, \; x_i \in R^n, \; y_i \in \{+1, -1\} \tag{1}$$

Consider the problem of separating the two classes represented by n examples. We need to find a linear function that separates the two classes (Figure 2):

$$f(x) = y = w \cdot x + b \tag{2}$$

That is, we must find the widest margin between the two classes, which is to minimize $\frac{1}{2} \|w\|^2$.

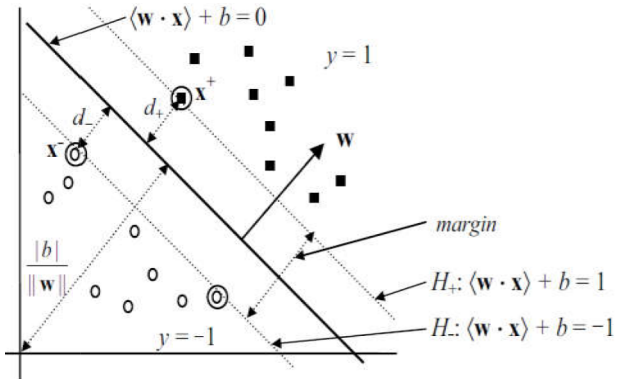

Figure 2. Principle of support vector machines.

Therefore, the decision rule for classification is:

$$f(x) = sign(\sum_{i=1}^{n} y_i \alpha_i k(x_i, x) + b) \tag{3}$$

where the function k is called the kernel.

4. Proposed approach

Our approach, schematized in Figure 3, aims to create a web page classifier based on the HITS algorithm. This algorithm was originally applied to classify web pages according to the quality of web pages linked to it. In this work, we propose to use HITS for reducing the size of the learning set and weighting the remaining features.

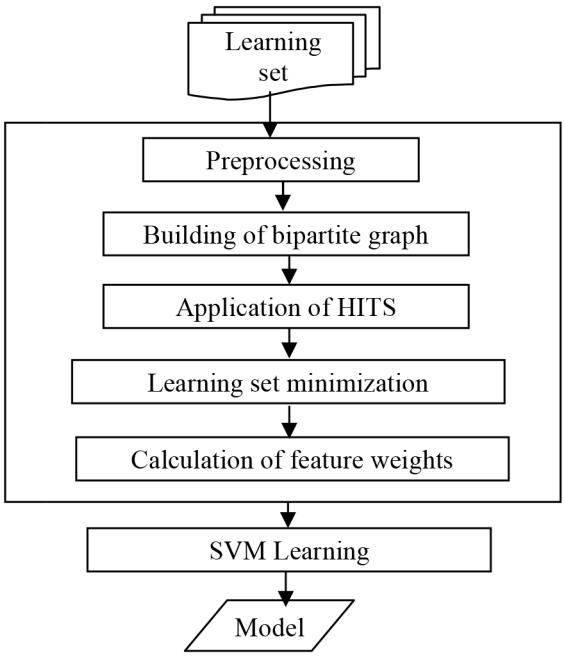

Figure 3. The steps of the proposed approach.

4.1. Preprocessing

The first step in the proposed approach is preprocessing. In this step, we perform several tasks in order to represent a web page as a bag of words. These tasks include:

1. Tags removal: delete all the HTML and scripting tags and extract only the original text.
2. Lowercase conversion: transform all characters in the text to lowercase.
3. Number deleting: remove all numbers from the text.
4. Stop words removal: clean the text from all stop words such as “for”, “while”, “and”...
5. Word selection: take only the words that consist of more than two letters.
6. Word stemming: stem words using the Porter stemming algorithm [24].

4.2. Building of bipartite graph

Originally, the HITS algorithm was proposed to analyze a graph of web pages, where nodes (web pages) can be hubs or authorities. It is under the assumption that there is a relation between web pages and their features, which can be represented by a bipartite graph where hubs are web pages and authorities are features. In graph theory, a graph is called bipartite if there is a partition of the set of nodes into two subsets U and V where each edge has an end in U and the other in V (Figure 4).

In this phase, we represent the cleaned learning set obtained from the previous step as a bipartite graph. The departure nodes are the web pages and the arrival nodes are the features. Figure 5 shows a general matrix (A[N,M]) that represents the graph where:

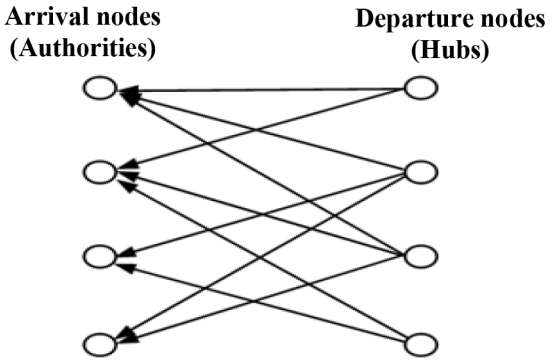

Figure 4. Example of a bipartite graph.

	f1	f2	f3	f4	…….....	fm
d_1	1	1	0	0	…….....	0
d_2	0	1	1	1	…….....	0
.					…….....	.
.					…….....	.
.					…….....	.
.					…….....	.
d_n	1	0	0	0		1

Figure 5. Example of a matrix form of a bipartite graph.

- N: represents the number of the web pages.
- M: represents the size of the feature vector.

The values of the matrix cells are binary values, where:

- A[i, j] is set to 1 if the feature number j is held in the page number i.
- A[i, j] is set to 0 otherwise.

4.3. Application of HITS algorithm

The third step in the proposed approach is the application of the HITS algorithm. We use this algorithm to rank web pages using the weight of features that they contain. In addition, we use the same algorithm to calculate weights of features based on the weight of their web pages. Thus, we consider each web page as a hub and each feature as an authority, where hubs point to authorities.

Weights of web pages are described as a vector h of dimension N (N: number of documents), where h_i is the weight hub of the page p_i. The weights of the features are described as a vector a of dimension m, where

Algorithm 1 HITS algorithm.

$h_0 \leftarrow (1, 1, \ldots, 1)$; $a_0 \leftarrow (1, 1, \ldots, 1)$;k$\leftarrow$1;
Repeat:
$a_k \leftarrow A^T Aa_{k-1}$;
$h_k \leftarrow AA^T h_{k-1}$;
$a_k \leftarrow a_k / ||a_k||$; // normalization
$h_k \leftarrow h_k / ||h_k||$; // normalization
k $\leftarrow$k + 1;
Until $||a_k - a_{k-1}|| < \varepsilon_a$ and $||h_k - h_{k-1}|| < \varepsilon_h$
return a_k and h_k;

a_i indicates the authority value of the feature i. We have two formulas to apply [2,25]:

$$a_i = \sum_{i \to j} h_j$$

$$h_j = \sum_{i \to j} a_i \tag{4}$$

The two previous relations of Eq. (4) are written as follows:

$$a = A^t \times h$$

$$h = A \times a \tag{5}$$

Here, A is a matrix that represents the bipartite graph. Algorithm 1 lists the HITS algorithm [2].

4.4. Learning set minimization

The fourth step in the proposed approach consists of reducing the learning set based on the two ranking lists, hub and authority. Practically, we reduce the learning set with two successive phases: web page reduction and feature selection.

4.4.1. Web pages reduction

At the beginning of this phase, we determine a threshold Th, and we delete all web pages that have hub values less than this threshold (hub values $< Th$).

The reduction of web pages from the learning set can produce a situation where there are features that do not belong to any web pages from the remaining pages (orphan features). Thus, we remove them from the feature vector.

At the end of this phase, we find that we achieve two types of minimization: vertical and horizontal. We mean by vertical minimization the reduction of features and we mean by horizontal minimization the reduction of web pages. These minimizations will accelerate the learning and the testing phases.

4.4.2. Features selection

Our approach proposes a filter-based method for feature selection, i.e. selects the features independently of the classifier. Thus, we focus on reducing features that have an authority value less than a threshold *Ta*.

At the end of this task, we find some web pages with zero features, i.e. web pages that do not have any feature presenting in the feature vector. Therefore, we remove these web pages from the learning set. Consequently, we achieve the second minimization on both axes.

4.5. Features weighting

A document in the vector model is represented as a weight vector in which each component is a weight of a feature. Originally, these weights are calculated using the TFIDF scheme. In the fifth step of this approach, we propose to calculate the weights using the authority vector, which is the output of the HITS algorithm, instead of using the TFIDF scheme.

Let a_i be the authority value of the term (feature) t_i and n_{ij} be the frequency of term t_i in document d_j. Then the weight of the term t_i in d_j (denoted w_{ij}) is given by:

$$w_{ij} = \begin{cases} \text{if } n_{ij} \neq 0 & \text{then } \log(a_i \times n_{ij}) \\ \text{else} & 0 \end{cases} \tag{6}$$

4.6. Support vector machine learning

Among the learning and data classification algorithms, we use SVM because SVM-based classifiers have shown promising results in the classification of text and web pages as reported in the literature [3,26].

To obtain good results, we have to go through a normalization phase where new weights are normalized to range between –1 and 1.

The normalization value $Norm_w$ of a weight w is obtained by the following formula:

$$Norm_w = 2 \times \frac{w - min_w}{max_w - min_w} - 1 \tag{7}$$

where max_w and min_w represent the maximum and the minimum weights respectively in the original corpus.

5. Experimental results and discussion

To validate our approach, we have carried out a comparative study between the TFIDF-based web page classifier and our approach that suggests the reduction of the learning set and the weighting of the features using the HITS algorithm. Our experiments are conducted on different datasets, namely WebKB-courses [27], WebKB-student, WebKB-faculty, Reuters-21578 R8, and Reuters-21578 R52 (http://www.cs.umb.edu/∼smimarog/textmining/datasets/). We focus on the support vector machines, decision tree, and naive Bayes as classification algorithms. The experiments were run on an i5 Intel PC with 4 GB main memory under the Microsoft Windows 7 operating system.

Instead of setting the threshold values manually, we proposed to remove web pages with hub values lower than the average of all hub values, as follows:

$$Th = \frac{\sum_{i=1}^{N} Hub_i}{N} \tag{8}$$

where N is the number of web pages

In the case of features, we keep only those that have authority values higher than or equal to half of the average of the authorities vector (Eq. (9)).

$$Ta_j = \frac{\sum_{i=1}^{M} auth_i}{2*M} \tag{9}$$

where M is the number of features

Web page classifiers are evaluated using standard measures originally used for the evaluation of information retrieval systems, which are accuracy, precision, recall, and F-measure. They are defined as follows [28]:

$$\text{Accuracy} = \frac{\text{true\_positives} + \text{true\_negatives}}{\text{positives} + \text{negatives}} \tag{10}$$

$$F - \text{mesure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{11}$$

$$\text{Precision} = \frac{\text{true\_positivs}}{\text{true\_positives} + \text{false\_positives}} \tag{12}$$

$$\text{Recall} = \frac{\text{true\_positives}}{\text{true\_positives} + \text{false\_negatives}} \tag{13}$$

Where:

- *Positives*: All web pages belonging to the desired class.
- *Negatives*: All web pages that do not belong to the desired class.
- *TruePositives* and *TrueNegatives*: documents correctly classified by the classifier.
- *FalsePositives* and *FalseNegatives*: documents that were misclassified by the classifier.

Experiment 1 *The web pages used in these experiments are extracted from the WebKB project or the Four Universities dataset [27]. The latter contains web pages collected in January 1997 from IT departments of four universities (Cornell, Texas, Washington, and Wisconsin). These web pages were manually classified into seven categories: student, faculty, staff, department, course, project, and others. Among these web pages, we have chosen a subset that has been partitioned into two classes: course and not the course (see Table 1). The desired class course contains a set of web pages of computer science lessons, and not course class contains web pages about various other subjects.*

The learning algorithm used was the SMO (sequential minimal optimization) [29]. We have chosen the SVM's parameters after a series of tests and we used those that afford us the best accuracy and F-measure.

To validate our approach we used n-cross-validation, where we split up our learning set into n $(n = 10)$ parts and we learn our system on $(n - 1)$ parts of the observations. The model validation will be on the nth part of the samples. Experiment rounds are repeated n times. The results of this experiment are summarized in Tables 2–5.

Table 1. Description of the used datasets.

	Learning dataset			Testing dataset		
	Total	Positives	Negatives	Total	Positives	Negatives
WebKB-course	2277	801	1476	253	89	164
WebKB-faculty	2785	745	2040	1383	372	1011
WebKB-student	2785	1085	1700	1383	540	843
Reuters-21578 R8	5485	1596	3889	2189	696	1493
Reuters-21578 R52	6532	1596	4936	2568	696	1872

Table 2. Summary of the obtained results using TFIDF scheme without reducing.

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	Average
Features	19251	19084	19146	19166	19491	19358	19428	19308	19490	19186	/
S Vs	883	747	728	722	730	698	718	942	709	697	/
Learning time (min:s:ms)	50:10:217	16:7:234	43:53:15	24:48:513	31:25:909	39:34:97	25:29:929	45:5:328	42:49:709	36:37:35	/
Accuracy	96.00%	99.20%	97.20%	97.60%	96.40%	99.20%	95.30%	95.70%	96.80%	95.70%	96.91 ± 1.4%
F-measure	0.94	0.99	0. 96	0.97	0.95	0.99	0.93	0.93	0.95	0.93	0.95 ± 0.02

Table 3. Summary of the obtained results using HITS model without reducing.

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	Average
Features	19251	19084	19146	19166	19491	19358	19428	19308	19490	19186	/
SVs	335	338	432	421	428	355	370	442	347	313	/
Learning time (min:s:ms)	19:25:866	14:33:55	17:35:528	19:14:74	23:39:76	20:7:138	19:13:259	28:50:327	17:4:593	14:50:533	/
Accuracy	96.00%	99.60%	97.20%	96.80%	97.60%	99.20%	96.00%	98.40%	98.00%	94.50%	97.48 ± 1.59%
F-measure	0.94	0.99	0.96	0.95	0.97	0.99	0.94	0.98	0.97	0.92	0.96 ± 0.02

Table 4. Summary of the obtained results after reduction using HITS model.

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	Average
Web pages	1002	1001	980	978	981	968	978	965	967	961	/
Features	3344	3374	3324:	3328	3407	3345	3406	3346	3401	3352	/
SVs	71	124	60	69	56	59	114	126	89	122	/
Learning time (min:s:ms)	0:21:30	0:19:918	0:20:600	0:22:692	0:26:218	0:21:421	0:22:672	0:20:740	0:23:323	0:26:338	
Accuracy	96.80%	100.00%	97.60%	98.00%	98.00%	98.00%	97.60%	97.60%	96.40%	96.80%	97,68 ± 0.99%
F-measure	0.95	1.00	0.97	0.97	0.97	0.97	0.97	0.97	0.95	0.95	0.97 ± 0.01

Table 5. Summary of the obtained results after reduction using TFIDF scheme.

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	Average
Web pages	1002	1001	980	978	981	968	978	965	967	961	/
Features	3344	3374	3324:	3328	3407	3345	3406	3346	3401	3352	/
S Vs	291	280	243	269	236	276	297	341	260	285	/
Learning time (min:s:ms)	2:6:352	1:16:61	0:56:361	0:33:488	0:34:79	1:49:47	2:27:381	2:19:779	0:40:819	1:33:4	/
Accuracy	96.80%	99.60%	97.60%	97.20%	96.80%	98.00%	95.30%	96.40%	96.40%	95.70%	96.98 ± 1.23%
F-measure	0.96	0.99	0.97	0.96	0.96	0.97	0.93	0.95	0.95	0.94	0.96 ± 0.01

The SVM-based classifiers that use HITS as a weighting scheme are more accurate (average: 97.48%) with a small set of support vectors than those that uses TFIDF as a weighting scheme (average: 96.91%). Tables 2 and 3 show that the HITS algorithm can be used as a competitor to the TFIDF model in the weighting of features contained in web pages.

The results in Table 4 indicate that our proposition accelerates significantly the learning time and minimizes the number of support vectors. In addition, this approach improves the classification accuracy. If we take as an example the first experiment (E1), we find that the learning time has been decreased from 19:25:866 (min:s:ms) before reduction (see Table 3) to 21:30 (s:ms) after reduction (see Table 4). Moreover, the number of support vectors was reduced from 335 to 71 and the accuracy has slightly improved from 96% to 96.80%.

We can conclude that the combination of the HITS algorithm, as a reduction method, with TFIDF, as a weighting scheme, is feasible and can give good results (Table 5).

Experiment 2 *We have applied our approach on two additional datasets: Reuters-21578 R8 and Reuters-21578 R52. The first holds 7674 documents on eight classes of topics (acq, crude, earn, grain, interest, money-fx, ship, trade). We have arbitrarily chosen the class acq as positive class and the rest as the negative class (see Table 1). The Reuters-21578 R52 dataset contains 9100 documents partitioned on 52 classes of topics (acq, earn, etc.). In this case, the positive class was acq also (see Table 1). The obtained results are summarized in Tables 6 and 7.*

Table 6. The obtained results using SVM classifier before reducing.

Weighting model			TFIDF				HITS			
	Pages	Features	F-M	Acc	SV	Learning time (h:min:s:ms)	F-M	Acc	SV	Learning time (h:min:s:ms)
Reuters-21578 R8	5485	14623	0.97	98.40%	1071	2:26:31:957	0.97	97.90%	693	1:41:31:418
Reuters-21578 R52	6532	15868	0.97	98.40%	1210	4:0:27:405	0.95	97.50%	796	0:55:9.410

Table 7. The obtained results using SVM classifier after reducing.

Weighting model			TFIDF				HITS			
	Pages	Features	F-M	Acc	SV	Learning time (h:min:s:ms)	F-M	Acc	SV	Learning time (h:min:s:ms)
Reuters-21578 R8	2165	3487	0.97	97.80%	616	0:22:25:960	0.95	96.90%	562	0:1:51:782
Reuters-21578 R52	2769	3775	0.95	97.50%	626	0:11:38:807	0.93	96.50%	587	0:9:19:292

Table 6 shows a comparison between two SVM classifiers where the first uses a TFIDF scheme as a weighting model and the second uses HITS for weighting the features. Though we slightly lose the classification accuracy using the HITS scheme, we significantly accelerated the learning step.

The obtained results, depicted in Table 7, support those obtained in Tables 4 and 5. Hence, we confirm that HITS algorithm can be used successfully as a weighting scheme as well as a reduction method when we use SVM-based classifiers.

Experiment 3 *To prove the efficiency of our approach, we attempted to replace the SVM by other classification algorithms (decision tree and naive Bayes) to classify the previously mentioned datasets (Table 1). The different experiments in this section are performed using the Weka (http://www.cs.waikato.ac.nz/ml/weka/) tool and obtained results are given in Tables 8 and 9.*

Table 8. The obtained results using the naive Bayes classifier.

	Before reducing				After reducing			
Weighting model	TFIDF		HITS		TFIDF		HITS	
	F-M	Acc	F-M	Acc	F-M	Acc	F-M	Acc
Reuters-21578 R8	0.96	96.07%	0.96	96.03%	0.903	90.13%	0.924	92.33%
Reuters-21578 R52	0.93	92.8%	0.97	96.57%	0.8	88.20%	0.93	92.64%
WebKB-course	0.9	89.72%	0.91	90.51%	0.879	87.75%	0.924	92.49%

Table 9. The obtained results using Decision Tree classifier

	Before reducing				After reducing			
Weighting model	TFIDF		HITS		TFIDF		HITS	
	F-M	Acc	F-M	Acc	F-M	Acc	F-M	Acc
Reuters-21578 R8	0.94	94.24%	0.94	94.29%	0.935	93.56%	0.94	93.92%
Reuters-21578 R52	0.95	94.82%	0.94	93.92%	0.93	93.11%	0.92	92.36%
WebKB-course	0.92	92.09%	0.913	91.30%	0.91	90.91%	0.9	90.12%

From Tables 8 and 9, we observe that the feature weights calculated using the HITS algorithm (authority values) give good accuracy and F-measure results when we apply SVM, naive Bayes, and decision tree classifiers. Thus, we can state that the authority values are reliable for calculating the weights of web page features.

Experiment 4 *In this section, we compare the HITS algorithm as a feature selection algorithm (i.e. without applying the web page reduction step) with chi-square and information gain methods where we have removed all the features that have ranking values equal to or less than zero. To conduct better performance comparison analysis, we applied these methods (our approach, chi-square, and information gain) on previous datasets (Table 1), with different classification algorithms (naive Bayes, decision tree, SVM). Moreover, for this comparison, we used also two different weighting schemes, which are TFIDF and HITS, with a variation of the threshold value. We report the obtained results in Tables 10–19.*

Table 10. The results after reduction of WebKB-course dataset using TFIDF scheme.

		Chi-2 TH = 0	Info-gain TH = 0	HITS					
				Ta = 0.00005	Ta = 0.00006	Ta = 0.00007	Ta = 0.00008	Ta = 0.0001	Ta = 0.0002
Web pages		2274	2274	2275	2275	2275	2275	2275	2275
Features		2033	2033	2424	2094	1944	1759	1550	755
Naive Bayes	Accuracy	93.28%	93.28%	90.12%	90.51%	90.91%	91.30%	91.30%	91.70%
	F-measure	0.933	0.933	0.903	0.91	0.91	0.914	0.914	0.917
Decision tree	Accuracy	93.28%	93.28%	92.89%	92.89%	92.89%	92.89%	92.89%	93.68%
	F-measure	0.932	0.933	0.928	0.928	0.928	0.928	0.928	0.936
SMO	SV	392	392	343	334	356	368	351	278
	Accuracy	95.65%	95.65%	94.86%	96.05%	95.65%	96.05%	95.65%	96.05%
	F-measure	0.957	0.957	0.949	0.96	0.957	0.96	0.957	0.96

Table 11. The results after reduction of WebKB-course dataset using HITS scheme.

		Chi-2 TH = 0	Info-gain TH = 0	HITS					
				Ta = 0.00005	Ta = 0.00006	Ta = 0.00007	Ta = 0.00008	Ta = 0.0001	Ta = 0.0002
Naive Bayes	Accuracy	92.89%	92.89%	90.91%	92.49%	92.09%	91.70%	92.89%	92.89%
	F-measure	0.929	0.929	0.91	0.925	0.921	0.917	0.929	0.929
Decision tree	Accuracy	93.68%	93.68%	92.49%	92.49%	92.49%	92.49%	92.49%	92.89%
	F-measure	0.936	0.936	0.924	0.924	0.924	0.924	0.924	0.929
SMO	SV	259	259	290	261	259	273	254	247
	Accuracy	96.44%	96.44%	96.84%	96.44%	96.44%	97.23%	97.23%	96.84%
	F-measure	0.964	0.964	0.969	0.964	0.964	0.972	0.972	0.969

Table 12. The results after reduction of Reuters-21578 R8 dataset using TFIDF scheme.

		Chi-2 TH = 0	Info-gain TH = 0	HITS					
				Ta = 0.00007	Ta = 0.00009	Ta = 0.0001	Ta = 0.00011	Ta = 0.00014	Ta = 0.0003
Pages		5485	5485	5485	5485	5485	5485	5485	5485
Features		1528	1528	1832	1629	1505	1369	1214	613
Naive Bayes	Accuracy	96.48%	96.48%	95.66%	95.48%	95.20%	94.88%	94.93%	94.38%
	F-measure	0.965	0.965	0.957	0.955	0.952	0.949	0.95	0.944
Decision tree	Accuracy	95.12%	94.61%	94.24%	94.88%	94.88%	94.79%	94.88%	94.75%
	F-measure	0.951	0.946	0.942	0.949	0.949	0.948	0.949	947
SMO	SV	967	758	1003	972	942	1089	973	688
	Accuracy	98.29%	98.21%	98.63%	98.49%	98.58%	98.40%	98.40%	98.54%
	F-measure	0.982	0.982	0.986	0.985	0.986	0.984	0.984	0.985

Table 13. The results after reduction of Reuters-21578 R8 dataset using HITS scheme.

		Chi-2 TH = 0	Info-gain TH = 0	HITS					
				Ta = 0.00007	Ta = 0.00009	Ta = 0.0001	Ta = 0.00011	Ta = 0.00014	Ta = 0.0003
Naive Bayes	Accuracy	96.12%	96.12%	96.25%	96.48%	96.30%	96.85%	96.67%	96.67%
	F-measure	0.961	0.961	0.963	0.965	0.963	0.969	0.967	0.967
Decision tree	Accuracy	95.11%	94.84%	94.84%	94.84%	94.70%	94.84%	94.93%	95.43%
	F-measure	0.951	0.948	0.948	0.948	0.947	0.948	0.949	0.954
SMO	SV	777	491	832	911	633	811	874	651
	Accuracy	98.22%	97.72%	98.17%	98.04%	97.90%	97.94%	97.99%	97.62%
	F-measure	0.982	0.977	0.982	0.98	0.979	0.974	0.98	0.976

Table 14. The results after reduction of Reuters-21578 R52 dataset using TFIDF scheme.

		Chi-2 TH = 0	Info-gain TH = 0	HITS					
				Ta = 0.00006	Ta = 0.00008	Ta = 0.00009	Ta = 0.0001	Ta = 0.00012	Ta = 0.00032
Pages		6532	6532	6532	6532	6532	6532	6532	6532
Features		1853	1853	1940	1716	1572	1433	1271	639
Naive Bayes	Accuracy	93.61%	93.61%	93.30	93.22%	92.99%	92.91%	92.72%	92.99%
	F-measure	0.938	0.938	0.935	0.932	0.931	0.929	0.927	0.931
Decision tree	Accuracy	94.70%	94.70%	94.86%	94.86%	94.85%	94.86%	94.86%	94.78%
	F-measure	0.947	0.947	0.948	0.948	0.948	0.948	0.948	0.948
SMO	SV	800	800	1364	1327	1242	1094	1012	845
	Accuracy	97.90%	97.90%	98.33%	98.36%	98.40%	98.48%	98.44%	98.33%
	F-measure	0.979	0.979	0.983	0.984	0.984	0.985	0.984	0.983

Table 15. The results after reduction of Reuters-21578 R52 dataset using HITS scheme.

		Chi-2 TH = 0	Info-gain TH = 0	HITS					
				Ta = 0.00006	Ta = 0.00008	Ta = 0.00009	Ta = 0.0001	Ta = 0.00012	Ta = 0.00032
Naive Bayes	Accuracy	96.50%	96.50%	96.26%	96.30%	96.26%	96.26%	95.79%	95.52%
	F-measure	0.965	0.965	0.963	0.963	0.963	0.963	0.958	0.956
Decision tree	Accuracy	94.00%	94.00%	94.12%	94.12%	94.12%	94.12%	94.12%	94.55%
	F-measure	0.94	0.94	0.941	0.941	0.941	0.941	0.941	0.946
SMO	SV	905	905	855	840	842	820	1002	931
	Accuracy	97.82%	97.82%	97.62%	97.70%	97.74%	97.70%	97.70%	97.51%
	F-measure	0.978	0.978	0.976	0.977	0.977	0.977	0.977	0.975

Table 16. The results before and after reduction of WebKB-student dataset using TFIDF scheme.

		Without reducing	Chi-2 TH = 0	Info-gain TH = 0	HITS					
					Ta = 0.00008	Ta = 0.00014	Ta = 0.00018	Ta = 0.00023	Ta = 0.00028	Ta = 0.00047
Web pages		2785	2782	2782	2785	2785	2784	2784	2781	2780
Features		7139	1314	1314	2160	1399	1127	913	784	463
Naive Bayes	Accuracy	86.19%	87.56%	87.56%	85.76%	85.18%	84.24%	83.95%	83.22%	82.00%
	F-measure	0.862	0.876	0.876	0.858	0.853	0.844	0.841	0.834	0.822
Decision tree	Accuracy	90.02%	90.24%	90.24%	90.02%	89.44%	87.64%	88.58%	88.00%	87.35%
	F-measure	0.9	0.902	0.902	0.9	0.895	0.876	0.885	0.88	0.873
SMO	SV	1186	841	841	1055	984	744	951	802	724
	Accuracy	92.19%	92.48%	92.48%	92.55%	92.77%	92.19%	92.26%	91.61%	91.40%
	F-measure	0.921	0.925	0.925	0.925	0.927	0.922	0.922	0.916	0.914

In Table 10, chi-square and information gain achieve 93.28% accuracy while our method achieves accuracy that ranges in the interval of 90.12%–93.68% with naive Bayes and decision tree. In addition, our method reaches 96.05% accuracy while chi-square and information gain achieve lower accuracy (95.65%) with SMO. From these

Table 17. The results before and after reduction of WebKB-student dataset using HITS scheme.

		Without reducing	Chi-2 TH = 0	Info-gain TH = 0	HITS Ta = 0.00008	Ta = 0.00014	Ta = 0.00018	Ta = 0.00023	Ta = 0.00028	Ta = 0.00047
Naive Bayes	Accuracy	83.66%	86.55%	86.55%	83.01%	80.48%	79.97%	79.32%	80.33%	79.18%
	F-measure	0.835	0.865	0.865	0.828	0.801	0.796	0.791	0.801	0.791
Decision tree	Accuracy	88.36%	90.89%	90.89%	88.43%	89.37%	89.01%	88.50%	86.62%	85.76%
	F-measure	0.883	0.909	0.909	0.884	0.894	0.89	0.885	0.866	0.857
SMO	SV	1503	1141	1141	1391	1342	1264	1240	1204	1147
	Accuracy	94.65%	94.50%	94.50%	94.79%	94.79%	94.29%	94.22%	94.29%	93.71%
	F-measure	0.946	0.945	0.945	0.948	0.948	0.943	0.942	0.943	0.937

Table 18. The results before and after reduction of WebKB-faculty dataset using TFIDF scheme.

		Without reducing	Chi-2 TH = 0	Info-gain TH = 0	HITS Ta = 0.00008	Ta = 0.00014	Ta = 0.00018	Ta = 0.00023	Ta = 0.00028	Ta = 0.00047
Web pages		2785	2781	2781	2785	2785	2784	2784	2781	2780
Features		7139	714	714	2160	1399	1127	913	784	463
Naive Bayes	Accuracy	76.50%	77.51%	77.51%	74.98%	75.05	76.43%	78.96%	77.66%	72.89%
	F-measure	0.776	0.785	0.785	0.762	0.763	0.776	0.799	0.786	0.743
Decision tree	Accuracy	89.30%	89.88%	89.88%	89.44%	89.95%	89.88%	89.23%	89.23%	88.58%
	F-measure	0.892	0.986	0.986	0.894	0.897	0.897	0.891	0.891	0.885
SMO	SV	1459	746	746	1170	1205	1201	1052	1192	1095
	Accuracy	92.99%	92.84%	92.84%	92.84%	92.84%	93.13%	91.97%	92.91%	93.06%
	F-measure	0.929	0.928	0.928	0.928	0.926	0.93	0.92	0.928	0.929

Table 19. The results before and after reduction of WebKB-faculty dataset using HITS scheme.

		Without reducing	Chi-2 TH = 0	Info-gain TH = 0	HITS Ta = 0.00008	Ta = 0.00014	Ta = 0.00018	Ta = 0.00023	Ta = 0.00028	Ta = 0.00047
Naive Bayes	Accuracy	78.81%	82.79%	82.79%	78.16%	80.33%	80.84%	81.34%	79.18%	80.33%
	F-measure	0.798	0.835	0.835	0.791	0.809	0.813	0.819	0.8	0.813
Decision tree	Accuracy	88.86%	89.15%	89.15%	88.86%	90.31%	90.31%	89.88%	89.73%	89.59%
	F-measure	0.888	0.891	0.891	0.888	0.902	0.902	0.898	0.896	0.895
SMO	SV	1236	1146	1146	1137	1141	1098	1111	1091	1099
	Accuracy	94.58%	94.50%	94.50%	94.14%	94.29%	94.29%	94.29%	94.14%	93.93%
	F-measure	0.944	0.945	0.945	0.94	0.942	0.942	0.942	0.94	0.938

results, we conclude that our approach improves accuracy using SMO and decision tree in the case of the use of the TFIDF scheme.

Table 11 shows that chi-square and information gain achieve better accuracy (93.68%) than our approach (92.89%) with decision tree. However, our method reaches the same accuracy (92.89%) of both chi-square and information gain with naive Bayes. Our approach reaches better accuracy (97.23%) than the two other methods (96.44%) using SMO with the HITS scheme (see Eq. (6)).

In Table 12, chi-square and information gain achieve 96.48% accuracy while our approach achieves lower accuracy (95.66%) with naive Bayes using the TFIDF scheme. In addition, chi-square achieves the best accuracy (95.12%), followed by our approach (94.88%) and then information gain (94.61%) with decision tree. Similarly to previous tables (Tables 10 and 11), our approach achieves the best accuracy (98.63%) compared to the two other methods (98.29% and 98.21%) with SMO using the TFIDF scheme.

In Table 13, our approach achieves 96.85% (95.43%) accuracy, while chi-square and information gain achieve a lower accuracy of 96.12% (95.11%, 94.84%) with naive Bayes (decision tree, respectively). In addition, there is a slight deviation of 0.05% between the accuracy obtained by the chi-square approach (98.22%) and our approach (98.17) with SMO using the HITS scheme.

As we can see from Table 14, chi-square and information gain achieve 93.61% accuracy, while our approach achieves 93.30% accuracy with naive Bayes using TFIDF. However, our approach achieves better accuracy than chi-square and information gain with both decision tree and SMO.

In Table 15, chi-square and information gain achieve 96.50% accuracy, while our approach achieves

96.30% accuracy with naive Bayes. However, with the decision tree, our approach improves the accuracy (94.55%) more than the two other methods (94.00%). Our approach achieves lower accuracy (97.74%) than the two other methods (97.82%) with SMO using the HITS scheme.

From the previous tables, we may safely conclude that:

- The HITS algorithm can be successfully applied to feature selection and dimensionality reduction methods.
- The weighting scheme that is based on the HITS algorithm (see Section 4.5) is reliable to calculate the weights of features contained in web pages.
- The combination of the HITS algorithm for feature selection and TFIDF may improve the accuracy of SVM-based classifiers.

5.1. Complexity of the proposed steps

In this paper, we have proposed three additional steps compared to the standard approach of a web page classifier. These steps are applying the HITS algorithm, web page reduction, and feature selection.

The complexity of one iteration of the HITS algorithm is estimated to O(M+N). The use of a bipartite graph and the sparseness of document vectors significantly enhanced the execution time of the algorithm. By setting $\varepsilon = 10^{-7}$ the algorithm takes a maximum of six iterations to converge.

Additionally, we observed during the comparison between our method and the chi-square and information gain techniques that our method significantly reduces the processing time of feature selection.

The complexity of the second proposed step, which is documents reduction, is estimated as O(N). The complexity of the feature reduction step is estimated as O(M). Finally, we confirm that the execution time of the added steps is negligible compared to the overall performance of the classifier.

6. Conclusion and perspectives

In this paper, we proposed a novel approach for creating a web page classifier based on the HITS algorithm to reduce the size of the learning set and the feature vector, as well as the weighting of the features.

The experiments that we performed on a set of web pages and text documents provided encouraging results. Our classifier reduces the learning time and the number of the support vectors, and it improves the accuracy of the classification in the conducted experiments.

However, this approach has one drawback, which is sensitivity to the threshold value. If we make a good choice for the threshold value, we can keep or even improve the accuracy of our system; otherwise, we will slightly lose the accuracy.

In future work, we plan to improve this approach by defining the thresholds *Ta* and *Th* automatically, as well as applying this approach for multiclass classification. Finally, we will try to improve the formula that we used to calculate feature weights.

References

[1] Qi X, Davison BD. Web page classification: features and algorithms. ACM Comput Surv 2009; 41: 12-40.

[2] Bing L. Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Berlin, Germany: Springer Science and Business Media, 2007.

[3] Joachims T. Text categorization with support vector machines: learning with many relevant features. In: 10th European Conference on Machine Learning; 21–23 April 1998; Chemnitz, Germany. pp. 137-142.

[4] Tang J, Alelyani S, Liu H. Feature selection for classification: a review. In: Aggarwal CC, editor. Data Classification: Algorithms and Applications. Boca Raton, FL, USA: CRC Press, 2013. pp. 37-64.

[5] Sánchez-Maroño N, Alonso-Betanzos A, Tombilla-Sanromán M. Filter methods for feature selection–a comparative study. Lect Notes Comp Sci 2007; 4881: 178-187.

[6] Forman G. Feature selection for text classification. In: Liu H, Motoda H, editors. Computational Methods of Feature Selection. Boca Raton, FL, USA: Chapman and Hall/CRC Press, 2007. pp. 157-176.

[7] Pinheiro RH, Cavalcanti GD, Correa RF, Ren TI. A global-ranking local feature selection method for text categorization. Expert Syst Appl 2012; 39: 12851-12857.

[8] Pinheiro RH, Cavalcanti GD, Correa RF, Ren TI. Data-driven global-ranking local feature selection methods for text categorization. Expert Syst Appl 2015; 42: 1941-1949.

[9] Yang J, Qu Z, Liu Z. Improved feature-selection method considering the imbalance problem in text categorization. Scientific World Journal 2014; 2014: 625342.

[10] Lee HM, Chen CM, Tan CC. An intelligent web-page classifier with fair feature-subset selection. In: IEEE IFSA World Congress and 20th NAFIPS International Conference; 25–28 July 2001; Vancouver, Canada. pp. 395-400.

[11] Dasgupta A, Drineas P, Harb B, Josifovsk V, Mahoney MW. Feature selection methods for text classification. In: ACM 2007 13th International Conference on Knowledge Discovery and Data Mining; 12–15 August 2007; San Jose, CA, USA. pp. 230-239.

[12] Mladenic D, Brank J, Grobelnik M, Milic-Frayling N. Feature selection using support vector machines. In: ACM 2004 27th Annual International SIGIR Conference; 25–29 July 2004; Sheffield, UK. pp. 234-241.

[13] Tu CJ, Chuang LY, Chang JY, Yang CH. Feature selection using PSO-SVM. IAENG International Journal of Computer Science 2007; 33: 111-116.

[14] Kim H, Howland P, Park H. Dimension reduction in text classification with support vector machines. J Mach Learn Res 2005; 6: 37-53.

[15] Chen CM, Lee HM, Chang YJ. Two novel feature selection approaches for Web page classification. Expert Syst Appl 2009; 36: 260-272.

[16] Page L, Brin S, Motwani R, Winograd T. PageRank: Bringing Order to the Web. Stanford, CA, USA: Stanford Digital Libraries Working Paper, 1997.

[17] Kleinberg J. Authoritative sources in a hyperlinked environment. J ACM 1999; 46: 604-632.

[18] Duhan N, Sharma AK, Bhatia KK. Page ranking algorithms: a survey. In: IEEE Advance 2009 Computing Conference; 6–7 March 2009; Patiala, India. New York, NY, USA: IEEE. pp. 1530-1537.

[19] Borodin A, Rosenthal JS, Roberts GO, Tsaparas P. Link analysis ranking: algorithms, theory and experiments. ACM T Internet Techn 2005; 5: 231-297.

[20] Xu X, Zhou C, Wang Z. Credit scoring algorithm based on link analysis ranking with support vector machine. Expert Syst Appl 2009; 36: 2625-2632.

[21] Deguchi T, Takahashi K, Takayasu H, Takayasu M. Hubs and authorities in the world trade network using a weighted HITS algorithm. PLoS One 2014; 9: e100338.

[22] Vapnik VN. Estimation of Dependences Based on Empirical Data. New York, NY, USA: Springer, 1982.

[23] Vapnik VN. The Nature of Statistical Learning Theory. New York, NY, USA: Springer, 1995.

[24] Porter MF. An algorithm for suffix stripping. Program 1980; 14: 130-137.

[25] Prajapati MR. A survey paper on hyperlink-induced topic search (HITS) algorithms for web mining. International Journal of Engineering Research & Technology 2012; 1: 2278-0181.

[26] Sun A, Lim EP, Ng WK. Web classification using support vector machine. In: ACM 2002 4th International Workshop on Web Information and Data Management; 4–9 November 2002; McLean, VA, USA. New York, NY, USA: ACM. pp. 96-99.

[27] Craven M, McCallum A, PiPasquo D, Mitchell T, Freitag D. Learning to extract symbolic knowledge from the World Wide Web. In: 15th National Conference on Artificial Intelligence; 26–30 July 1998; Madison, WI, USA. pp. 509-516.

[28] Han J, Kamber M. Data Mining: Concepts and Techniques. San Francisco, CA, USA: Morgan Kaufmann, 2006.

[29] Platt JC. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical Report, Microsoft Research. Seattle, WA, USA: Microsoft, 1998.