A Novel Approach for Web Page Classification using Optimum features

J. Alamelu Mangai, V. Santhosh Kumar

Department of Computer Science Engineering, BITS, Pilani – Dubai, International Academic City, Dubai, U.A.E

Abstract
The boom in the use of Web and its exponential growth are now well known. The amount of textual data available on the Web is estimated to be in the order of one terra byte, in addition to images, audio and video. This has imposed additional challenges to the Web directories which help the user to search the Web by classifying selected Web documents into subject. Manual classification of web pages by human expertise also suffers from the exponential increase in the amount of Web documents. Instead of using the entire web page for classifying it, this article emphasizes the need for automatic web page classification using minimum number of features in it. A method for generating such optimum number of features for web pages is also proposed. Machine learning classifiers are modeled using these optimum features. Experiments on the bench marking data sets with these machine learning classifiers have shown promising improvement in classification accuracy.

Keywords:
Web page Classification, Web directories, features, machine learning

1. Introduction

There are basically three different forms of searching the Web. The first is to use search engines that index a portion of the Web documents as a full-text database. The second is to use Web directories, which classify selected Web documents by subject. The third is to search the Web exploiting its hyperlink structure[1]. Since traversing the Web pages using their hyperlinks to satisfy the user's requirements is tedious, search engines are used frequently for this.

Search engines are broadly classified into two, based on the search methodology, namely, robot style search engines and directory-style search engines. Robot style search engines collect all terms including words and phrases to index Web pages. Users can retrieve the Web pages through these representative keywords of a Web page. The benefits of such robot style search engines are: 1) the algorithms of extracting terms and indexing pages are relatively simple. 2) they do not require too much human effort or maintenance. 3) they usually have compact or friendly interfaces [2]. Directory style search engines require man power to classify a large number of Web pages into each appropriate category with their themes. Therefore it costs much time and care. Also, the effort to classify every increasing number of Web pages becomes increasingly difficult. So, automatic Web page classification is the better solution to reduction in costs and man power successfully.

There are many additional challenges to Web page classification than conventional text classification. Web can be seen as a very large, unstructured but ubiquitous database. This triggers the need for efficient tools to manage, retrieve and filter information from this data base. This problem is also becoming important in huge intranets where we want to extract or infer new information to

support a decision process. With the exponential increase in the amount of Web data, automatic classification of Web pages into predefined categories is essential to assist the Web directories. This also helps in focused crawling. Web pages can be classified by two methods: syntactic and semantic. This article emphasizes syntactic classification, which uses a set of words or patterns in a web page to classify it. The alternative approach uses natural language analysis of the text. The techniques to preprocess natural language and extract the text semantics are too costly for large amounts of data. In addition, they are only effective with well structured text, a thesaurus and other conceptual information. Since classification is a supervised machine learning task, to get better classification results, data has to be pre-processed with utmost care. With no quality data, there is no quality mining results. Much of the work in this area have been focusing on fine tuning the data to be mined, rather than the classifier itself. This article suggests a method to classify a Web page with only minimum number of representative features or terms extracted from it without using the entire Web page. The optimum number of features are selected using a three step procedure, by filtering the features in each subsequent step. The experimental results have shown good classification accuracy with these optimum features.

The rest of the article is organized as follows. Section 2 is a brief of the related work, Section 3 highlights the proposed work, Section 4 is a glimpse of the experimental results and discussion and Section 5 concludes this study.

Manuscript received May 5, 2011
Manuscript revised May 20, 2011

2. Related Work

Web-page classification is much more difficult than pure-text classification due to a large variety of noisy information embedded in them. In [3], the authors have proposed a new web-page classification algorithm based on web page summaries generated by human experts. Experimental results have shown that the proposed summarization-based classification algorithm achieves an approximately 8.8% improvement as compared to pure-text-based classification algorithm. Traditional machine learning algorithms have also been tweaked to automatically classify the web pages as in [4]. Both text and the context-features of a web page such as hyperlinks and HTML tags are used to model a SVM classifier to classify the web pages [5]. Positive Example Based Learning (PEBL) [6] is a framework for Web page classification which eliminates the need for manually collecting negative training examples in preprocessing.

Most Web page classification models typically apply the bag of words (BOW) model to represent the feature space. The original BOW representation, however, is unable to recognize semantic relationships between terms. One possible solution is to apply the topic model approach based on the Latent Dirichlet Allocation algorithm to cluster the term features into a set of latent topics. Terms assigned into the same topic are semantically related. A novel hierarchical classification method based on a topic model and by integrating additional term features from neighboring pages is proposed in [7]. A web page classification method for creating a high quality collection of researchers' homepages is proposed in [8]. This method reduces manual assessment required for assuring given precision/recall using a recall-assured and a precision-assured classifier. A structure based approach for automatic classification of web pages is proposed in [9]. Apart from the text contents, the structure of a web page, the images and the links in it are used to automatically classify web pages into a few broad categories.

Ant-Miner, the first Ant Colony algorithm is used in [10] for discovering classification rules in the field of web content mining, and have proved that it is more effective than C5.0 in two sets of BBC and Yahoo web pages used in their experiments. The URL features of a web page along with the features of its sibling pages are combined using Bayesian algorithm to classify the web pages [11]. Most of the automatic web page classification algorithms ignore the conflict between the fixed number of categories and the growing number of web pages going into the system. It is also required to search through all existing categories to make any classification. A dynamic and hierarchical classification system that is capable of adding new categories as required, organizing the web pages into a tree structure, and classifying web pages by searching through only one path of the tree structure is proposed in [12]. The test results show that the proposed single-path search technique reduces the search complexity and increases the accuracy by 6% comparing to related algorithms.

To effectively classify web pages solving the synonymous keyword problem, a web page classification method is proposed in [13] based on support vector machine using a weighted vote schema for various features. Latent semantic analysis is used to find the semantic relations between keywords, and between documents. The latent semantic analysis method projects terms and a document into a vector space to find latent information in the document. At the same time, the text features from web page content are also extracted. Through these text features, web pages are classified into a suitable category. These two features are sent to the SVM for training and testing respectively. Based on the output of the SVM, a voting schema is used to determine the category of the web page. Experimental results indicate that this method is more effective than traditional methods. There are several difficulties in integrating traditional classification approaches in a search engine. The authors in [14] present an Entity-Based Web Page Classification Algorithm, which can be embedded in search engines easily. In the algorithm, an Entity System is built to classify web pages immediately before indexing jobs. [15] presents a novel ontology-based web page classification method for the knowledge grid environment, which utilizes generated metadata from web pages as the inter medium to classify the web pages by ontology concepts. All these works have concentrated on improving an existing Web page classifier or developing a novel Web page classifier.

3. Proposed Work

3.1 Architecture of the proposed Web page classifier

Many researches have focused on either improving a Web page classifier or developing a novel classifier. This work suggests that preprocessing the web pages to ease the classification is one way of achieving better classification results. A method to identify the best and optimum number of features that better represent the category of a Web page is proposed. Various machine learning classifiers are modeled using only these features and not the entire Web page. These classifiers are used to predict the category of new Web pages. Fig 1 shows the architecture of the proposed Web page classifier.

As stated earlier, this work concentrates more on preprocessing the web pages to be classified, rather than the classifier itself. More emphasize is on selecting the best and representative features of a Web page category. Since real time data sets are of higher dimensions, and classifiers learn from the training features, the time taken to model the

classifier will be more, if proper feature selection is not done. This work attempts to minimize the time taken to build the classifier by using a series of feature selection steps. The algorithm of the proposed work is illustrated in Fig2.

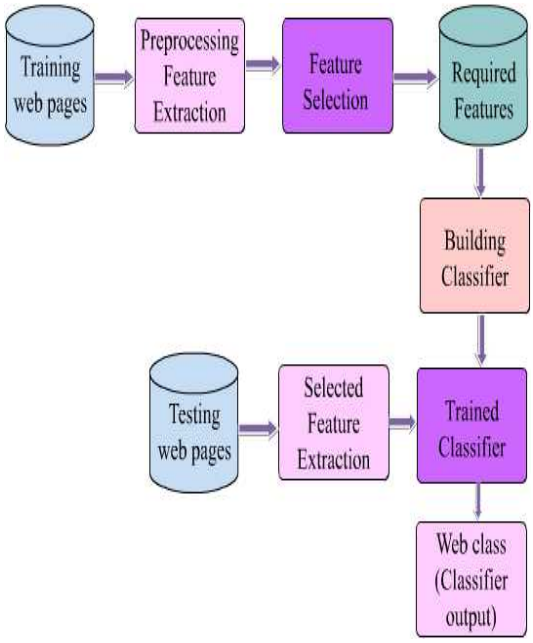

Fig 1: Architecture of the proposed Web page Classifier

1. Preprocessing
   Remove HTML tags, stop words, punctuation symbols, digits and hyphens.
2. Feature Set Extraction, F1
   for each new term in the web page find its term frequency and inverse document frequency among the web page collection.
3. Feature Selection, F2
   Select features from F1 using CfsSubset Evaluator method.
4. Train a C4.5 decision tree classifier using F2.
5. The features in the pruned decision tree form the final set of features F3.
6. Evaluate the performance of the machine learning classifiers using the final set of features F3.

Fig 2: Algorithm of the proposed Work

3.2 Term-Frequency – Inverse Document Frequency

Features that appear rarely in a web page will better reflect the category of a web page. In information retrieval, to identify such discriminative terms, a statistical measure known as term frequency – inverse document frequency is used. A measure of how significant a term is in a document collection is given by the term frequency. In a document collection term frequency of a term (ti) is defined as

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

where ni,j is the number of occurrences of the considered term (ti) in document dj, and the denominator is the sum of the number of occurrences of all terms in document dj. The inverse document frequency is a measure of the importance of the term in the entire document collection.

$$\mathrm{idf}_i = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

where | D | : total number of documents in the corpus and $|\{d : t_i \in d\}|$: number of documents where the term ti appears. Then,

$$(\text{tf-idf})_{i,j} = \mathrm{tf}_{i,j} \times \mathrm{idf}_i$$

when inverse document frequency factor is incorporated , the weight of terms that occur very frequently in the collection diminishes and the weight of terms that occur rarely increases. Since a web page is converted to a text file after preprocessing, tf-idf statistical measure is used to select the initial set of informative rare features F1 of the web page.

3.3 Feature Selection using CfsSubsetEval

The initial set of features will be of high dimension. As the number of features increases, the time required to model a classifier also increases. Hence, selecting the best features for building the classifier is more significant than modeling the classifier itself. Feature selection algorithms use two objects namely, a feature evaluator and a search method. The evaluator determines a method to assign a worth to each candidate subset of features. The search method determines the style of search on the feature subsets. CfsSubsetEval uses a correlation measure to evaluate the worth of a subset of features by considering the individual predictive ability of each feature along with the degree of redundancy between them. The subset of features that are highly correlated with the class while having low inter-correlation are preferred. CfsSubsetEval is used with the rank search method to select the features F2. The rank search method uses a subset evaluator like GainRatio to rank all subsets. A decision tree classifier C4.5 is modeled using the selected features F2.

3.4 C4.5 Decision Tree Classifier

The decision tree induction algorithms like C4.5 have incorporated in their learning phase an automatic feature selection strategy [16]. They are best suited to extract meaningful features from large measurement spaces [17] [18]. They are also proven to perform better in domains that involve correlated features. It is non-parametric and computationally fast. If the entire set of initial features F1 is used by C4.5, it takes a longer time to construct the tree and it is likely to be too complex. Also, the features with high information gain are nearer to the root in the constructed tree. So, features are selected from F1 using CfsSubsetEvaluator. A C4.5 classifier is built only on F2. Only those features that C4.5 uses in its pruned decision tree are selected as the final set of features F3.

3.5 Classification

Classification is a supervised learning task where, the classifier learns how to predict the category of a test example from a set of labeled examples. The final set of features selected by the proposed method are evaluated using three machine learning classifiers namely, Decision tree, k Nearest Neighbour, oneR, multilayer perceptron and RBF[19]. The performance of the classifiers are evaluated using the metrics like CCI and macro F- measure. CCI refers to number of correctly classified instances. F-Measure is used in Information Retrieval to characterize the performance of the classifier.

$$F-measure = \frac{2 \times recall \times precision}{recall + precision} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

where, TP = number of true positives
FP = number of false positives
FN = number of false negatives

$$recall = \frac{\text{number of relevant documents retrieved}}{\text{total number of relevant documents}}$$

$$precision = \frac{\text{number of relevant documents retrieved}}{\text{total number of documents retrieved}}$$

Precision and recall measure how precise and how complete the classification is on the positive class. They are introduced using a confusion matrix. A confusion matrix contains information about actual and predicted results given by a classifier

Experiments are conducted using WebKB data set available in UCI repository. It is a benchmarking dataset for machine learning research. This data set contains WWW-pages collected from computer science departments of various universities. The pages are manually classified into the following categories student, faculty, staff, department, course, project and others. For the analysis of the proposed work course web pages are considered as positive examples (p) and non course web pages as negative examples (n). Two sample sizes of p70-n30 and p350- n150 are experimented. Features are extracted in subsequent steps as stated in Section 3. The worth of these optimum features are estimated using the different classifiers in WEKA, a suite of tools and functions for data mining tasks [20]. The classifiers are tested using 10 – fold cross validation. The time taken to model each classifier with the initial set of features and the final optimum features for each sample size are also evaluated.

4. Experimental Setup

4. Result and Discussion

4.1 Initial Feature Selection F1

The web pages are initially preprocessed as follows. The HTML tags, stopwords, punctuation and digits are removed. Words are reduced to their root. To diminish the weight of frequently occurring words and increase the weight of rare words in a web page, the term frequency – inverse document frequency of each word in a web page is computed. This forms the initial set of features F1 as listed in Table 1.

Table 1: Initial set of features F1

S. No.	*Sample size*	*No. of Instances*	*No. of features*
1.	p70-n30	100	2774
2.	p350-n150	500	7651

4.2 Feature Selection F2

CfsSubsetEvaluator is run on F1, to further select the features with more information gain on the class and less inter-correlation. This forms F2 as shown in table 2.

4.3 Feature Selection F3

A C4.5 decision tree classifier is built using F2. The features that C4.5 uses in its pruned tree are alone selected as the final features F3 as shown in table 3.

Table 2: Features selected, F2

S. No.	*Sample size*	*No. of Instances*	*No. of features*
1.	p70-n30	100	31
2.	p350-n150	500	42

Table 3: Features selected, F3

S. No.	*Sample size*	*No. of features*
1	p70-n30	5
2	p350-n150	13

4.4 Classification

The performance of the various machine learning classifiers are evaluated on the different feature sets F1, F2 and F3. The time taken to model the classifier in each case is also observed as illustrated in the following tables.

Table 4: CCI on the feature set F1

S. No	*Sample Size*	*DT*	*kNN*	*oneR*	*MLP*	*RBF*
1.	p70-n30	88	73	89	----	----
2.	p350-n150	455	376	423	----	----

Table 5: Time taken to model the classifiers on the feature set F1

S. No	*Sample Size*	*DT*	*kNN*	*oneR*	*MLP*	*RBF*
1.	p70-n30	1.22	0	0.48	----	----
2.	p350-n150	32.59	0.02	6.26	----	----

Table 6: CCI on the feature set F2

S. No	*Sample Size*	*DT*	*kNN*	*oneR*	*MLP*	*RBF*
1.	p70-n30	92	99	89	99	99
2.	p350-n150	463	476	425	485	481

Table 7: Time taken to model the classifiers on the feature set F2

S. No	*Sample Size*	*DT*	*kNN*	*oneR*	*MLP*	*RBF*
1.	p70-n30	0.22	0	0	2.3	0.09
2.	p350-n150	0.11	0	0.02	17.73	0.41

It can be observed that the classification accuracy has increased with all classifiers with the reduced features F2. Also the time taken to model the classifiers is also reduced. kNN has exhibited more accuracy over F2 than F1.

Table 8: CCI on the feature set F3

S. No	*Sample Size*	*DT*	*kNN*	*oneR*	*MLP*	*RBF*
1.	p70-n30	97	99	89	99	99
2.	p350-n150	472	475	425	477	479

Table 9: Time taken to model the classifiers on the feature set F3

S. No	*Sample Size*	*DT*	*kNN*	*oneR*	*MLP*	*RBF*
1.	p70-n30	0	0	0	0.2	0.02
2.	p350-n150	0.03	0	0	3.02	0.09

It can be inferred from Table 8 and 9 that the time taken to build the classifiers with F3, has significantly reduced with no compromise in the accuracy. Therefore for better accuracy with reduced resource utilization, selecting the best and optimum features is more important. This is one way of improving the performance of the classifiers.

5. Conclusion

To produce better classification results with efficient utilization of the resources, the entire web page need not be used. Since web pages have data of high dimensions they need to be preprocessed well to identify their best representative features which better reflect their category. And so, without bias to frequent terms, a multilevel feature selection is proposed in this work. And so our results show that the proposed method of feature selection helps to identify a less number of features but with high information gain, that contribute more to the classification accuracy.

References

[1] Makoto Tsukada, Takashi Washio, Hiroshi Motoda (2001), "Automatic Web page Classification using machine learning methods", Web Intelligence : Research and Development, LNCS, Springer, Vol. 2198, pp:303-313.

[2] Hao Wen, Liping Fang, Ling Guan (2008). "Automatic Web Page Classification using various Features", LNCS, Springer Verlag, Vol : 5353, pp. 368 -376.

[3] Dou Shen, Zheng Chen, Qiang Yang, Hua-Jun Zeng, Benyu Zhang, Yuchang Lu, Wei- Ying Ma (2004) , "Web-page classification through summarization", In the Proceedings of the 27th annual international ACM SIGIR 04, conference on. Research and Development in Information Retrieval, New York, ACM Press, pp:242- 249..

[4] Oh-Woog Kwon, John-Hyeok Lee (2000) , "Web page classification based on k-nearest neighbor approach" , Proceedings of the fifth international workshop on Information retrieval with Asian languages, Hong Kong, China, ACM Press, pp:9-15..

[5] Aixin Sun, Ee-Peng Lim, Wee-Keong Ng ,(2002),"Web classification using support vector machine", Proceedings of

the 4th international workshop on Web Information and Data Management, New York, ACM Press, p: 96-99.

[6] Hwanjo Yu, Jiawei Han, Kevin Chen-Chuan Chang (2004), "PEBL: Web Page Classification without Negative Examples", IEEE Transactions on Knowledge and Data Engineering, Vol.16, No.1, pp: 70-81.

[7] Wongkot Sriurai, Phayung Meesad, Choochart Haruechaiyasak (2010), "Hierarchial web page Classification based on a Topic Model and Neighboring Pages Integration", International Jopurnal of Computer Science and Information Security, Vol. 7, No.2.

[8] Yuxin Wang and Keizo Oyama (2006), "Web Page Classification Exploiting Contents of Surrounding Pages for Building a High – quality Homepage Collection", LNCS, Springer,Vol. 4312, pp:515-518.

[9] Arul Prakash Asirvatham, Kranthi Kumar Ravi (2001), "Web Page Classification based on Document Structure", Awarded Second Prize in National Level Student Paper Contest conducted by IEEE India Council..

[10] Nicholas Holden and Alex A. Freitas, (2004), "Web Page Classification with an Ant Colony Algorithm", Parallel Problem Solving from Nature, LNCS, Springer, Vol.3242, pp:1092-1102.

[11] Sara Meshkizadeh, Amir Masoud Rahmani, Mashallah Abassi Dezfuli (2010), "Web Page Classification based on URL features and Features of Sibling Pages", IJCSIS, Vol. 8, No:2.

[12] Xiaogang Peng, Ben Choi (2002), "Automatic Web Page Classification in a Dynamic and Hierarchial Way", In Proceedings of Second IEEE International Conference on Data Mining, Washington DC, IEEE Computer Society, pp:386-393.

[13] Rung-Ching Chen, Chung-Hsun Hsieh (2006), "Web Page Classification based on a support Vector Machine using a weighted vote schema", Expert Systems with Applications, Vol. 31, Issue 2, pp:427-435.

[14] Yicen Liu, Mingrong Liu, Liang Xiang and Qing Yang, (2008), "Entity-Based Classification of Web Page in Search Engine", ICADL, LNCS, Vol. 5362, pp:411- 412.

[15] Hai Dong Hussain, F. K. Chang E (2009), "An Ontology based Web Page Classification Approach for the Knowledge Grid Environment", In Proceedings of the 5th International Conference on Semantics, Knowledge and Grid, Oct 12, China : IEEE Computer Society, pp: 120 – 127.

[16] Perner P, Aptec C (2000), "Empirical Evaluation of feature subset selection based on a real world data set", Engineering Application of AI,.Vol. 17, pp:285-288.

[17] Borak J. S. Strahler A.H (2002), "Feature Selection using decision tree an application for the MODIS land cover algorithm, IEEE Explore, pp:243-245.

[18] Piramuthu S and Shaw M J (1994), "On Using decision tree as feature selector for feed forward neural network", International Symposium on Integrating Knowledge and Neural Heuristics, pp:67-74.

[19] S.B.Kotsiantis (2007), "Supervised Machine Learning : A Review of Classification Techniques", Informatica 31, pp:249-268.

J. Alamelu Mangai received her M.E from Annamalai University, India in 2005. She is currently pursuing her Ph.D from BITS, Pilani. She is also working as a Senior Lecturer in the Department of Computer Science and Engineering in BITS, Pilani – Dubai Campus, U.A.E, since 2006. Her research interests include data, text and web mining.

V. Santhosh Kumar received his Ph.D. degree from Indian Institute of Science, Bangalore, India. He is working as Assistant Professor at BITS Pilani, Dubai Campus, U.A.E.

His research interests include Data Mining,, Performance Evaluation of Computer Systems.