11-2002

Web classification using Support Vector Machine

Aixin SUN
Nanyang Technological University

Ee Peng LIM
Singapore Management University, eplim@smu.edu.sg

DOI: https://doi.org/10.1145/584931.584952

Citation

SUN, Aixin and LIM, Ee Peng. Web classification using Support Vector Machine. (2002). *WIDM '02: Proceedings of the 4th international workshop on Web information and data management*. 96-99. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/969

Web Classification Using Support Vector Machine[*]

Aixin Sun
School of Computer Engineering
Nanyang Technological University, Singapore
sunaixin@pmail.ntu.edu.sg

Ee-Peng Lim[†]
School of Computer Engineering
Nanyang Technological University, Singapore
aseplim@ntu.edu.sg

Wee-Keong Ng
School of Computer Engineering
Nanyang Technological University, Singapore
awkng@ntu.edu.sg

ABSTRACT

In web classification, web pages from one or more web sites are assigned to pre-defined categories according to their content. Since web pages are more than just plain text documents, web classification methods have to consider using other context features of web pages, such as hyperlinks and HTML tags. In this paper, we propose the use of Support Vector Machine (SVM) classifiers to classify web pages using both their text and context feature sets. We have experimented our web classification method on the WebKB data set. Compared with earlier FOIL-PILFS method on the same data set, our method has been shown to perform very well. We have also shown that the use of context features especially hyperlinks can improve the classification performance significantly.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval; H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms

Experimentation

Keywords

Web Classification, Web Mining, SVM

1. INTRODUCTION

With the rapid development of World Wide Web (WWW), huge amount of information are now accessible by the web users. Low cost, high accessibility, and publishing freedom are the characteristics of the web that have contributed to its popularity. In this paper, we focus the classification problem for web pages, also known as the **web classification** problem.

Web pages are essentially hypertext. Besides text and multimedia components, web pages include context features such as hyperlinks, HTML tags and meta data. Most research efforts have assumed that the text components of web pages provide the primary information for web classification while the other non-text components can be used to improve the classification performance [1, 2, 14, 19]. Classical text classification techniques therefore have been widely adopted and extended for web classification. A good text classification survey is given in [15].

In this paper, we propose to use the SVM classifier to perform web classification. Our objectives is to study the impact of different web page features on the performance of web classification. We use text features alone as the baseline features and try out different combinations of text, title and hyperlink features. We have evaluated our classification method on the WebKB data set which is commonly used for web classification experiments [2, 7]. Compared to the results of the FOIL-PILFS method by Craven and Slattery on the same data set [2], our SVM-based methods performed much better in terms of F_1 measure. We have also shown that by considering the text, title and anchor words as features, the best classification performance can be obtained.

The rest of the paper is organized as follows. In Section 2, we give a survey of the previous work in hypertext classification. Our web classification method is described in Section 3. The experiments results are reported in Section 4 and finally we conclude our work in Section 5.

2. RELATED WORK

Web pages consist of both text and context features such as HTML Tags and hyperlinks. As the text features are believed to provide the primary content information about web pages, the simplest approach (also known as the *Text Only* approach) is to use text features only in web classification [3, 12]. Web classification methods using this approach set the baseline performance results for other methods that also consider other context features. Depends on the context features are used, we divide the other works on web classification into the *hypertext* approach, *link analysis* approach and *neighborhood category* approach.

In the *hypertext* approach, web pages are represented by

[*] The work is partially supported by the SingAREN 21 research grant M48020004.

[†] Dr. Ee-Peng Lim is currently on leave at Dept. of SEEM, Chinese University of Hong Kong, China.

WIDM'02, November 8, 2002, McLean, Virginia, USA.
Copyright 2002 ACM 1-58113-593-9/02/0011 ...$5.00.

both the content (text) and context features. One kind of context features are the document structures defined by the HTML tags, i.e., the layout of the web pages [5]. The other kind of context features are from the neighboring pages, for example, the text on the links (i.e., anchor text), the paragraph where the anchor text appears, the headings of the section where the link occurs [6, 8, 1]. The *link analysis approach* involves the application of learning algorithms to handle both the text components in web pages and the linkage among them, for example, the FOIL-PILFS classifier [2]. The *neighborhood category approach* web classification approach exploits the category assignment of the already classified neighboring pages to determine the category(ies) for a web page [1, 14].

3. WEB CLASSIFICATION

3.1 Support Vector Machine

Invented by Vapnik [16], SVM is a method to learn functions from a set of labelled training data. Classifiers built on SVM have shown promising results in text classification [10, 4]. Moreover, compared with other types of classifiers, SVM is both efficient and effective [10]. In all our experiments (see Section 4), the classification methods are implemented using the SVM^{light} package[1] by Joachims [9]. The output file of the SVM^{light} contains decision function output value for each classified document.

3.2 Web Page Features

To evaluate the effect of using context features on SVM classifiers, we have decided to adopt the hypertext approach. Two kinds of context features have been identified, namely, *title* and *hyperlink*. The contribution of these context features to web classification will be examined against the baseline method using text only features. The feature extraction method and web page representation used are given below.

3.2.1 Text Only (X)

Similar to all the other works using only text features, each web page is represented using a set of words extracted from its text component only. The plain text is obtained by removing HTML tags from a web page. The stop-words are removed and all the remaining words are stemmed. Formally, a page P is represented using the word $x.w_i$ extracted from the text component X, i.e., $P = \{x.w_i \mid x.w_i \in P.X\}$.

3.2.2 Text + Title (T)

Most of web pages have a title element enclosed by the <title> and </title> tags. The title of a page usually gives a good summary of the content of the page and therefore it could potentially provide more important information. In our web classification method, words extracted from title element of page P, say $t.w_j$'s, are used as features in addition to the features from text only approach $x.w_i$'s. As the result, a web page P is represented using two sets of features, $P = \{x.w_i, t.w_j \mid x.w_i \in P.X, t.w_j \in P.T\}$, where T represents the title element. Note that if a word w appears in the title element, two index terms ($x.w$ and $t.w$) will be created and they are assigned different word ids, $w.id$'s.

3.2.3 Text + Anchor Words (A)

Various experiments have shown that the inclusion of all words from the neighboring pages worsens web classification results compared to the web classification method using text only features [1, 19]. Nevertheless, this conclusion is not surprising if the assumption of text component representing the primary content of web pages holds. Inclusion of all words from the neighboring pages actually brings the *primary information* of the neighboring pages to the local page generating more noticeable noise for web classification. On the other hand, the assumption also suggests that the local text should not be simply discarded as done in [6]. We consider the anchor words of the incoming links as additional context features. The anchor words of the incoming links are prefixed. Given a page P, let $P.A$ be the set of anchor words on the links leading to P. P is represented by both the local words and anchor words, $P = \{x.w_i, a.w_k \mid x.w_i \in P.X, a.w_k \in P.A\}$.

3.2.4 Text + Title + Anchor Words (TA)

To complete the study of the effect of different context features, we also consider the web page representation using all features, i.e., local text, title words and anchor words, i.e, $P = \{x.w_i, t.w_j, a.w_k \mid x.w_i \in P.X, t.w_j \in P.T, a.w_k \in P.A\}$.

4. EXPERIMENTS

4.1 Data Set

The WebKB data set[2] contains web pages collected from computer science departments of 4 universities (Cornell, Texas, Washington and Wisconsin) in January 1997. The 4159 web pages collected were manually classified into 7 categories: student, faculty, staff, department, course, project and other. Similar to work in [2], only four categories were experimented, i.e., student, faculty, course and project. All pages from the remaining categories were used as negative training and test pages in the experiments. All experiments used *leave-one-university-out* cross-validation to conduct training and evaluation [11]. In our experiments, web pages are represented by binary feature vectors. All the HTML tags are discarded beforehand and the words are stemmed.

4.2 Experimental Setting

In *leave-one-university-out* cross-validation, we conduct multiple train-and-test experiments on Web->Kb dataset. The training data for the SVM classifiers are highly unbalanced as the proportion of positive training web pages ranges from 2% to 18%. In our experiments, we solved the problem by adjusting the *cost-factor* (parameter j in SVM^{light}) which defines the number of times the training errors on positive examples outweight the errors on negative examples. Similar to the work in [13], we defined the *cost-factor* to be the ratio of the number of negative training examples over positive ones. We used the default settings for the other parameters in SVM^{light}.

$$j = \frac{Tr-}{Tr+} \tag{1}$$

In addition to the cost factor, we also used *SCut* thresholding strategy [17, 18] to improve the accuracy of SVM

[1] SVM^{light}: http://svmlight.joachims.org/

[2] http://www-2.cs.cmu.edu/∼webkb/

Table 1: Classification results of different methods

Method	Course			Faculty			Project			Student			F_1^M
	Pr	Re	F_1	Pr	Re	F_1	Pr	Re	F_1	Pr	Re	F_1	
Foil-Pilfs	0.526	0.533	0.530	0.550	0.36	0.435	0.277	0.274	0.275	0.655	0.462	0.542	0.445
SVM(X)	0.501	0.692	0.581	0.497	0.642	0.560	0.0921	0.267	0.137	0.65	0.733	0.689	0.492
SVM(T)	0.553	0.746	0.635	0.566	0.702	0.627	0.0995	0.353	0.155	0.693	0.727	0.709	0.532
SVM(A)	0.55	0.721	0.624	0.63	0.706	0.666	0.319	0.31	0.314	0.671	0.782	0.722	0.582
SVM(TA)	0.637	0.734	0.682	0.63	0.691	0.659	0.299	0.357	0.325	0.735	0.726	0.730	0.599

Table 2: The correct F_1 values

Method	Course	Faculty	Project	Student	F_1^M
SVM(X)	0.577	0.555	0.136	0.683	0.488
SVM(T)	0.628	0.626	0.149	0.699	0.525
SVM(A)	0.612	0.661	0.258	0.717	0.562
SVM(TA)	0.671	0.643	0.264	0.723	0.575

classifiers after they have been constructed. With *SCut*, the original training data of each train-and-test experiment (for a specific university and category pair) is further divided into two subsets, one is used to train a classifier and the other one (known as validation set) is used to learn the optimized threshold. The locally optimized threshold for the category is the score (or value) where the best pre-defined performance measure can be achieved on the validation set, e.g., F_1. In our experiments, as we use *leave-one-university-out* cross-validation for each category, web pages in the training set of each classification task can be naturally split, i.e., the three universities. Another *leave-one-university-out* cross-validation is used to find the optimized threshold for F_1 measure with respect to the output score of the SVM classifiers. For each category, threshold values from 0 to -1 at step of 0.05 have been tested, the optimized threshold is the value at which the averaged F_1 value of the training set cross-validation is the highest. Note that the optimized threshold for each category is obtained purely from the *training set* of the category.

4.3 Results and Discussion

Four sets of classification results using SVM classifiers are presented in Table 1. The four runs of the classification experiments were based on four kinds of web page representations, i.e., X (text only), T (text + title), A (text+anchor words) and TA (text + title + anchor words). Results of Foil-Pilfs taken from [2] are included in Table 1 for easy comparison. This comparison is possible because the same dataset was used in [2], and the computation of F_1 based on the *macro-averaged* Pr and Re for each category using leave-one-university-out cross-validation was also used in [2]. However, the "correct" way of computing F_1 values pointed by Yang in [18] is to compute the F_1 value for each category followed by averaging of the per category F_1 values. In our work, the "correct" F_1 values were also computed and reported in Table 2. The "correct" F_1 values are usually slightly lower than the corresponding values reported in Table 1 as pointed out in [18].

Note that although the same data set and the way of selection of training and test web pages in Foil-Pilfs method were used in our experiments, the feature representation and feature selection are different. In our experiments, *set-of-words* representation was used and no feature selection employed while *bag-of-words* and frequency-based vocabulary pruning were used by Foil-Pilfs (see [2] for more details).

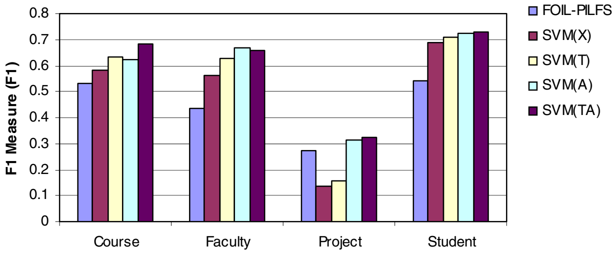

Figure 1: F_1 Comparison

Comparing the precision, clearly, SVM(X) is a loser compared with Foil-Pilfs, especially when very few positive training examples and positive test web pages were given, e.g., project and faculty. Improvements were observed for all categories except project when the title components of the web pages were used, i.e., SVM(T). In those improvement cases, SVM classifiers performed slightly better than Foil-Pilfs. When the anchor words were used together with the local text components (SVM(A)), the precisions for all the four categories exceeded those of SVM(X) and Foil-Pilfs. However, compared to the results of SVM(T), only the categories with relatively fewer positive training and test pages (i.e., project and faculty) observed increases in precision. SVM(TA) delivered the best performance as it considered both the titles and anchor words. Other than the project category, SVM(TA) outperformed all other methods. For the project category, the precision returned by SVM(TA) was quite close to that of SVM(A) which did well in the category.

By looking at the recall values, the following conclusions can be drawn. Firstly, the SVM methods performed far better than Foil-Pilfs in recall for categories with larger number of positive training examples. Only the recall value of SVM(X) for the project category is not as good as Foil-Pilfs. Secondly, the use of context features such as titles and anchor words led to increases in recall in most cases compared with the methods using text components only. The only exception is the student category. Thirdly, it is hard to tell which combinations of context features contributed most to the recall. We observed that the method using text and title features (SVM(T)) did well for the faculty category, the method using text and anchor words (SVM(A)) was the best for the student category, and the method using text, title and anchor words emerged for the course category.

To consolidate the performance of all the methods, we now examine their F_1 values. The F_1 value comparison is shown in Figure 1. It is clear that the SVM classifiers delivered better performance than FOIL-PILFS for larger categories course, student and faculty, even when only the text components were used. A small increase in F_1 measure was observed when title components were used for all the four categories. This suggested that the title components help in little way in classifying the Web->Kb dataset. Compared with the F_1 results of method using text alone (SVM(T)), the use of anchor words (SVM(A)) led to significant increase in F_1, especially for the small category project. Although there is a slight drop for faculty category when both title and anchor words were used compared with using anchor words only, the use of both title and anchor words clearly yielded the best results among all the methods tested.

In summary, SVM performed very well in web page classification and the use of context features especially anchor words indeed improved the classification performance. Our results are consistent with the ones reported in [8] where methods using text only and anchor words were evaluated on a set of YAHOO! pages.

5. CONCLUSION AND FUTURE WORK

In this paper, we studied the effect of using context features in web classification using SVM classifiers. Compared with FOIL-PILFS, our experiments have shown that SVM-based web classification methods performed very well on the WebKB data set even using the text components only. It was shown that context features consisting of title components and anchor words, improved the classification accuracy significantly. However, the method without using anchor words could not deliver consistently good classification for all the categories. If we exclude the PROJECT category which has very few positive training web pages, our SVM-based web classification methods achieved consistently above 0.6 values for the F_1 measure.

While our results are encouraging, there are still much improvement to be made. More context features can be experimented to examine their usefulness in web classification. In particular, context features obtained from link analysis and neighborhood categories can be added to our SVM-based method.

6. REFERENCES

[1] S. Chakrabarti, B. E. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proc. of the ACM SIGMOD1998*, pages 307–318, Seattle, USA, 1998.

[2] M. Craven and S. Slattery. Relational learning with statistical predicate invention: Better models for hypertext. *Machine Learning*, 43(1-2):97–119, 2001.

[3] S. T. Dumais and H. Chen. Hierarchical classification of Web content. In *Proc. of the SIGIR2000*, pages 256–263, Athens, GR, 2000.

[4] S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proc. of the CIKM1998*, pages 148–155, 1998.

[5] F. Esposto, D. Malerba, L. D. Pace, and P. Leo. A machine learning apporach to web mining. In *Proc. of the 6th Congress of the Italian Association for Artificial Intelligence (IA*AI1999)*, pages 190–201, Bologna, Sep 1999.

[6] J. Furnkranz. Exploiting structural information for text classification on the WWW. In *Proc. of the 3rd Symposium on Intelligent Data Analysis (IDA99)*, pages 487–498, Amsterdam, NL, 1999.

[7] L. Getoor, E. Segal, B. Taskar, and D. Koller. Probabilistic models of text and link structure for hypertext classification. In *Proc. of the Int. Joint Conf. on Artificial intelligence Workshop on Text Learning: Beyond Supervision*, Seattle, WA, Aug 2001.

[8] E. Glover, K. Tsioutsiouliklis, S. Lawrence, D. Pennock, and G. Flake. Using web structure for classifying and describing web pages. In *Proc. of the WWW2002*, Hawaii, USA, May 2002.

[9] T. Joachims. SVM^{light}, An implementation of Support Vector Machines (SVMs) in C. http://svmlight.joachims.org/.

[10] T. Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proc. of the ECML1998*, pages 137–142, Chemnitz, DE, 1998.

[11] D. D. Lewis. Applying support vector machines to the TREC-2001 batch filtering and routing tasks. In *Proc. of the TREC2001*, Gaithersburg, Maryland, Nov 2001.

[12] D. Mladenic. Turning Yahoo to automatic web-page classifier. In *Proc. of the 13th European Conf. on Artificial Intelligence*, pages 473–474, Brighton, UK, Aug 1998.

[13] K. Morik, P. Brockhausen, and T. Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *Proc. of the 16th Int. Conf. on Machine Learning*, pages 268–277, Bled, Slowenien, 1999.

[14] H.-J. Oh, S. H. Myaeng, and M.-H. Lee. A practical hypertext categorization method using links and incrementally available class information. In *Proc. of the 23rd ACM SIGIR2000*, pages 264–271, Athens, GR, 2000.

[15] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[16] V. N. Vapnik. *The nature of statistical learning theory.* Springer Verlag, Heidelberg, DE, 1995.

[17] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1-2):69–90, 1999.

[18] Y. Yang. A study on thresholding strategies for text categorization. In *Proc. of the ACM SIGIR2001*, pages 137–145, New Orleans, USA, Sep 2001.

[19] Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2-3):219–241, 2002.