Customizable Instance-Driven Webpage Filtering Based on Semi-Supervised Learning

Mingliang Zhu, Weiming Hu, Xi Li and Ou Wu
National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
{mlzhu, wmhu, lixi, wuou}@nlpr.ia.ac.cn

Abstract

The World Wide Web has been growing rapidly in recent years, along with increasing needs for content-based webpage filtering. But most existing filtering systems cannot easily satisfy the personalized filtering demands from different users at the same time. In this paper, a customizable instance-driven webpage filtering strategy is proposed. For different users, different webpage filters are produced by our system through mining the certain webpage classes they focus on. A semi-supervised learning (SSL) approach is applied for obtaining a precise description of the webpage class which a user wants to filter based on the small sized user instance set he or she provided. Subsequently, a feature selection step is performed and a Bayes classifier is created over the enlarged training set. Experimental results show the great stability and high performance of our proposed method, and it outperforms existing methods.

1. Introduction

Along with the rapid growing of the Web, the webpage filtering has been in demand and been studied for a long time. On one hand, some people spread harmful or illegal contents, such as pornography which is not appropriate for children. On the other hand, people may want their accessibilities to the web to be limited in a certain range in which they are interested, for there are too many of webpages. These demands require a good and customizable webpage filtering method.

Although filtering in different fields is in demand, current methods mainly focus on the pornography filtering. Early techniques [1] include: 1). PICS (Platform for Internet Content Selection), which allows the publishers voluntarily assign category labels to webpages. 2). URL blocking, which blocks webpages in a manually maintained URL black-list. 3). Key-word counting, which blocks a webpage if the "key-words" frequencies in it exceeds a threshold. These traditional methods are not suitable for today's fast growing Web.

Recent efforts rely more on the content analysis of webpages. Lee [1] made use of neural networks to filter pornography webpages. Numbers of occurrences of 55 key-words as well as a few whole page statistics served as the classification features. Wu [2] proposed a CNN-like word net that describes different types of key-words and their interactions to recognize objective webpages. The utilities of these key-word based methods are usually limited to a certain field: the construction of the key-word list requires much human labor, and it may not be easy to find enough discriminative key-words in some fields other than pornography.

Du [3] proposed a text classification method to filter pornography webpages, which can be applied to other fields. Top n% of cosine similarities between the input webpage and all samples in the labeled webpage set are averaged and compared with a threshold to determine the nature of the new page. However, this method requires a well representative labeled training set, which is difficult to construct. Besides, the similarity calculations during the classification stage may be quite time-consuming. Furthermore, the threshold value is corpus-dependant, and thus should be carefully tuned for different fields of webpages to filter.

Related works also include a couple of traditional text classification methods, such as Naïve Bayes [4], k-nearest neighbors (kNN) [5], Decision Trees [6] and Support Vector Machines (SVM) [7].

As discussed above, existing webpage filtering systems have their shortcomings. They are either key-word based, which makes them less customizable, or require a large and representative labeled sample set which is difficult to construct. For real-time webpage filtering systems, the classification time cost is another critical drawback of some existing methods.

In this paper, we propose a new instance-driven filtering strategy to solve these problems. A certain user's personalized filtering demand is expressed by providing a set of webpage instances, indicating which pages are ones to be filtered and which are not. A customized webpage filter is then created out of our system. Our method is key-word free, customizable and can be

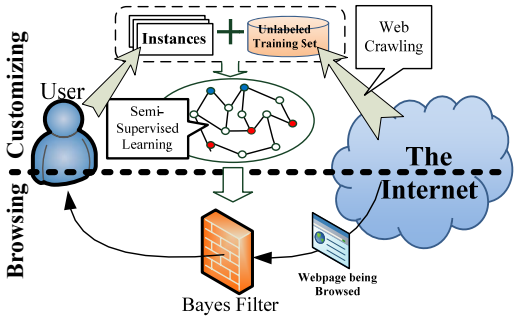

Figure 1. Outline of the framework

applied to filter webpages of any category or a combination of multiple categories.

We use the graph-based semi-supervised learning (SSL) [8] to extend the small numbered user instances onto a large webpage training set, which can be constructed automatically by a web crawler. Unlike Du's method [3], our large training set does not need to be labeled. We then build a Bayes classifier over the extended training set, because of the stability and high performance of this classier. The overview of the framework of our method is shown in Figure 1.

2. Extending the User Instances

2.1 Making use of the Unlabeled Training Set

Suppose we are given a set $\boldsymbol{L}$ of n_l instances of webpages from the user, including n_{l+} positive (to filter) instances ($y_i = +1$) and $n_{l-}=n_l-n_{l+}$ negative instances ($y_i = -1$). Our goal is to assign a label $y \in Y=\{+1, -1\}$ to a new coming page x. Usually the user instance set size n_l is quite small, so L may not be well representative of the distribution of the webpages to filter. To solve this problem, we make use of a large unlabeled webpage set (denoted set $\boldsymbol{U}$) to obtain a precise representation, using semi-supervised learning. The SSL methods train classifiers from both labeled and unlabeled training samples. They are very effective when labeled samples are hard to obtain, and has been successfully applied to many tasks, including the text classification [9].

The unlabeled training set can be constructed with the help of a typical web crawler [10]. When this set grows large, it can be regarded as an approximate of the distribution of the real web. Note that this set tends to be highly *unbalanced*, i.e. the number of negative samples overwhelms that of the positives.

2.2 Graph-Based Semi-Supervised Learning

We adopt the graph-based SSL [8] to extend the small instance set onto the large unlabeled training set., The graph approaches make better use of the data distribution revealed by the unlabeled data, compared with other SSL methods [11].

We create a graph whose nodes are all the data points ($L \cup U$). Webpage samples are presented as bag-of-word vectors and the weight of the edge between nodes i and j is the *binary* cosine similarity (considering only the presence of a term but ignoring its frequency) of nodes i and j:

$$w_{ij} = Sim(i,j) = \frac{n_{i \wedge j}}{\sqrt{n_i \cdot n_j}} \qquad (1)$$

where n_i and n_j denote the numbers of terms present in i and j respectively, and $n_{i \wedge j}$ denotes the number of terms that present in both webpages. Instead of a fully connected graph, we create a kNN graph in our algorithm, in which nodes i and j are connected if i is in j's k-nearest-neighborhood or vice versa. The kNN graph is more time efficient while propagating (to be described in the following paragraphs) and perform much better on the highly unbalanced training set.

Then we propagate the labels through edges, using the Label Propagation algorithm. It is naturally that we assume data points that are similar to have the same label, so large edge weights allow labels to travel through more easily. An $n \times n$ probabilistic transition matrix P is defined as follows:

$$P_{ij} = P(i \to j) = \frac{w_{ij}}{\sum_{l=1}^{k} w_{il}} \qquad (2)$$

where P_{ij} is the probability of transit from node i to j. We compute soft labels f for nodes. f is an $n \times 2$ matrix, where f_{i1} and f_{i2} are the beliefs that the node i has the label -1 and $+1$, respectively. The Label Propagation algorithm runs as follows:

Step i. Initialize f arbitrarily
Step ii. Clamp the labeled data points:
for each labeled node i, let f_{i1}=0 and f_{i2}=1 if i is labeled negative, and vice versa.
Step iii. Propagate: $f \leftarrow Pf$
Step iv. Goto Step ii until f converges

It can be proven that f converges to a fixed point, and the propagation actually minimizes the quadratic energy function over the graph [4]:

$$E(f) = \frac{1}{2}\sum_{i,j} w_{ij}(f_i - f_j)^2 \qquad (3)$$

3. Classification

The graph-based SSL is originally a transductive method. It only works on the training data ($L \cup U$). But induction is required in order to classify new coming webpages. Common inductive efforts classify a new point x by "freezing" the converged graph and propagating labels from its nearest neighbor [8], kNN-s, or even all the points [12] in the frozen graph. But these

methods need to calculate the similarities between x and each point in the graph, which has high computational cost during classification when the unlabeled training set is large. So we build a Bayes classifier for classification because of its stability and high performance.

For a new page x, we calculate $P(+1|x)$ and $P(-1|x)$. By the conditional version of Law of Total Probability:

$$P(Y \mid x) = \sum_{t \in x} P(Y \mid t, x) P(t \mid x) \quad (4)$$

where t is a term in webpage x. Suppose:

$$P(Y \mid t) = P(Y \mid t, x) \quad (5)$$

which means the probability of t to be positive or negative is independent of which webpage t is in, or in another word, the label is *conditional* independent of the webpage if given the term (Note that the label is obviously dependent of the webpage without given the term). Then the probabilities can be calculated as:

$$P(Y \mid x) = \sum_{t \in x} P(Y \mid t) P(t \mid x) \quad (6)$$

where $P(t|x)$ is the normalized frequency of term t in x:

$$P(t \mid x) = \frac{freq\ of\ term\ \ t\ in\ x}{sum\ of\ \ freq\ of\ \ all\ terms\ in\ x}$$

$P(Y \mid t)$ can be estimated from the training set. Let F_{t+} be the number of the positive webpages in the training set containing term t and F_{t-} be the number of the negative ones, then $P(Y|t)$ can be calculated as:

$$P(+1 \mid t) = \frac{\varepsilon + F_{t+}}{2\varepsilon + F_{t+} + F_{t-}}, \quad P(-1 \mid t) = \frac{\varepsilon + F_{t-}}{2\varepsilon + F_{t+} + F_{t-}} \quad (7)$$

where ε is a small constant preventing zero frequencies.

After estimating the $P(Y|t)$, a feature selection step is performed. We sort the $P(Y|t)$ list by the $P(+1|t)$ values in descending order (or by the $P(-1|t)$ values in ascending order in another word), and only *nTermTop* terms on top of the sorted list and *nTermBottom* terms on bottom are kept for classification. All other probability values are set to 0.5 and thus ignored (terms with the $P(Y|t)$ values equal to 0.5 make the same contribution to $P(+1|x)$ and $P(-1|x)$ in Equation (6)). So both space and time demands for classification are reduced after the feature selection step.

Finally, the decision is made by comparing $P(+1|x)$ and $P(-1|x)$. If $P(+1|x) > P(-1|x)$, the target webpage is classified positive and filtered, or otherwise negative.

4. Testing Results

In order to evaluate the performance of our proposed method, we compare our filtering results with several other methods: Du's method [3], SVM, Transductive SVM (TSVM) [13], and graph based SSL with common kNN induction. SVM is reported the best among traditional text classification algorithms [14] while TSVM is one of its SSL versions. We used SVM^{light} [15] to implement the SVM and TSVM.

We also choose the task of pornographic webpage filtering to make our method comparable with others. We collected 3000 normal Chinese webpages from web sites under 10 categories of the sohu web site directory [16] and 500 ones from different pornographic sites. We intended to include a number of sex-related (such as sex education) webpages in the normal set, which usually contain words in common key-word lists and used to be misclassified by previous filtering methods. In order to demonstrate the instance-driven webpage filtering strategy, we randomly divide our dataset into some subsets, summarized in Table 1. The TSVM works even poorer than SVM on the unbalanced training set in our experiments. So we also create a balanced scheme for TSVM by selecting 297 out of the 1967 unlabeled normal samples (although this will probably not be the case of real applications).

Table 1. Summary of dataset

	Normal Set	Porn. Set	Total
Labeled Instances (L)	33	33	66
Unlabeled Samples (U)	1967	297	2264
Training ($L + U$)	2000	330	2330
Test Set	1000	170	1170
Total	3000	500	3500

The filtering rate *FR* and the over-filtering rate *OR* are used as accuracy criteria. Let n_{pn} be the number of webpages in the test set that are positive (pornographic in our case) but classified negative (normal) by the filter, and n_{pp}, n_{np}, n_{nn} are defined similarly. Then:

$$FR = \frac{n_{pp}}{n_{pp} + n_{pn}}, \quad OR = \frac{n_{np}}{n_{pp} + n_{np}} \quad (8)$$

Texts in all webpages are first passed into the ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System) [17] to segment and extract Chinese words. Then stop-words are removed, as well as barely used words, which are ones appear in less than 5 webpages out of our whole dataset.

Table 2 shows the results. The *standard deviation* of *FR* and *OR* are included to show the stability of each method on the small sized labeled set. The page parsing time is ignored in the classification time since it is common in all methods. Focusing on the filtering accuracy, the methods which make use of unlabeled samples, including ours, the balanced TSVM and the kNN induction, generally perform better than those do not. For a real-time webpage filter, the classification time cost is another critical consideration. The SVM based methods have the least time costs because the texts are usually linear separable so that the model generated by SVM is simple. Our method, as well as Du's, is also excellent in time performance. But both Du's method and the kNN

Table 2. Testing results (Avg. of 10 runs for each method)

	FR	std(*FR*) ($\times 10^{-2}$)	*OR*	std(*OR*) ($\times 10^{-2}$)	Avg. classification time (ms)
Our Method	**98.12%**	0.823	**1.82%**	1.01	22.3
Du Scheme1 (Threshold = 0.09)	97.35%	0.971	15.58%	1.80	33.7
Du Scheme2 (Threshold = 0.15)	85.82%	3.84	3.32%	**0.477**	33.7
SVM	88.06%	11.9	13.48%	21.5	**5.1**
TSVM (unbalanced)	94.41%	10.3	73.92%	2.37	7.0
TSVM (balanced)	92.30%	2.58	3.21%	1.88	6.0
*k*NN Induction	94.41%	**0.588**	4.35%	1.32	1879

induction spend their classification time mainly in calculating similarities between target webpage and all webpages in the training set. Therefore, when new webpages are added into the training set (which usually leads to better classification accuracy), the time cost for those methods may become unbearable. In our method, filtering time cost of the Bayes classifier based on selected features is not relative to the number of webpages in the training set.

5. Conclusion

This paper has proposed an instance-driven strategy for webpage filtering, which is fully customizable and key-word free compared with most existing ones. Different filtering demands from users are expressed by different user-specific webpage instance sets. A semi-supervised learning based method has been presented for the task. With the help of an unlabeled webpage set which is large sized and can be crawled from the Internet automatically, our method has achieved a high filtering accuracy based on only a small number of user instances. Compared with previous techniques, our method gets a more precise description of the webpage class that is to be filtered, so its filtering accuracy significantly outperforms others. Furthermore, among all the methods tested, our method is the least affected by the truth that the negative and positive webpage sets are extremely unbalanced. The feature selection step and Bayes classifier have greatly improved the stability and classification time cost of our method.

6. Acknowledgment

This work is partly supported by NSFC (Grant No. 60520120099 and 60672040) and the National 863 High-Tech R&D Program of China (Grant No. 2006AA01Z453), and Natural Science Foundation of Beijing (Grant No. 4041004).

7. References

[1] P.Y. Lee, S.C. Hui and A. Fong, "Neural Networks for Web Content Filtering", *Intelligent Systems*, Vol. 17, Iss. 5 pp. 48-57, 2002.

[2] O. Wu and W. Hu, "Web Sensitive Text Filtering by Combining Semantics and Statistics", In *Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering*, pp. 663-667, 2005.

[3] R. Du, R. Safavi-Naini and W. Susilo, "Web Filtering Using Text Classification", In *Proceedings of the 11th IEEE International Conference on Network*, pp. 325 - 330, 2003.

[4] D.D. Lewis and M Ringuette, "A Comparison of Two Learning Algorithms for Text Categorization", In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994.

[5] Y.Yang and X. Liu, "A re-examination of text categorization methods", In *Proceedings of the 22nd Annual International ACM SIGIR conference on Research and development in information retrieval,* pp. 42-49, 1999.

[6] T. Mitchell, *Machine Learning*. McGraw Hill, 1996.

[7] V. Vapnic, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

[8] X. Zhu, "Semi-Supervised Learning with Graphs", *Doctoral dissertation, School of computer Science, Carnegie Mellon University*, 2005.

[9] K. Nigam, "Using unlabeled data to improve text classification", *Doctoral dissertation, School of computer Science, Carnegie Mellon University*, 2001.

[10] G. Pant, P. Srinivasan, and F. Menczer. "Crawling the Web", In M. Levene and A. Poulovassilis, editors, *Web Dynamics*, Springer, 2003.

[11] J. He, J. Carbonell and Y. Liu, "Graph-Based Semi-Supervised Learning as a Generative Model". *IJCAI,* pp. 2429-2497, 2007.

[12] O. Delalleau, Y. Bengio and N.L.Roux, "Efficient non-parametric function induction in semi-supervised learning", In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005.

[13] T. Joachims, "Transductive Inference for Text Classification using Support Vector Machines", In *Proceedings of the 16th International Conference on Machine Learning*, 1999.

[14] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features", In *Proceedings of the 10th European Conference on Machine Learning*, Springer, 1998.

[15] T. Joachims, *Making Large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[16] Sohu web site directory, http://dir.sohu.com/.

[17] ICTCLAS, http://www.nlp.org.cn/project/project.php?proj_id=6.