Improving Web Spam Detection with Re-Extracted Features

Guang-Gang Geng
Institute of Automation
Chinese Academy of Sciences
Beijing 100080, P. R. China
guanggang.geng@ia.ac.cn

Chun-Heng Wang
Institute of Automation
Chinese Academy of Sciences
Beijing 100080, P. R. China
chunheng.wang@ia.ac.cn

Qiu-Dan Li
Institute of Automation
Chinese Academy of Sciences
Beijing 100080, P. R. China
qiudan.li@ia.ac.cn

ABSTRACT
Web spam detection has become one of the top challenges for the Internet search industry. Instead of using some heuristic rules, we propose a feature re-extraction strategy to optimize the detection result. Based on the predicted spamicity obtained by the preliminary detection, through the host level web graph, three types of features are extracted. Experiments on WEBSPAM-UK2006 benchmark show that with this strategy, the performance of web spam detection can be improved evidently.

Categories and Subject Descriptors
H.5.4 [Information Interfaces and Presentation]: Hypertext/ Hypermedia; K.4.m [Computer and Society]: Miscellaneous; H.4.m [Information Systems Applications]: Miscellaneous

General Terms
Measurement, Experimentation, Algorithms.

Keywords
Link spam, Content spam, Web spam, Machine learning.

1. INTRODUCTION
The practices of crafting web pages for the sole purpose of increasing the ranking of these or some affiliated pages, without improving the practicability to the surfers, are called web spam[4]. Web spam seriously deteriorates search engine ranking results. Finding effective methods for spam detection has become increasingly urgent.

Analogous to the methods used in fighting email spam, [1] detected content spam via a number of statistical content based attributes. [5] implemented a classifier to catch a large portion of spam, then several heuristics rules were designed to decide whether a node should be relabeled. [2] summarized the existing content and link based method, detected web spam with machine learning algorithms, then gave some heuristic rules to improve the performance.

Both [2] and [5] achieved good results with the preliminary machine learning algorithms, but they optimized the detection result with some heuristic rules. As we all know, effective spam detection is essentially an "arms race" between search engines and spamers. Heuristic rules based detection system can be more easily manipulated by the spamers. Compared with other optimization methods mentioned in [2][5], stack graph learning (*Sgl*) can mine topological dependencies for spam detection more reasonably[2], which is a simple two-stage learning method.

In this paper, we propose a feature re-extraction strategy to build a robust and high-performance detection system. Based on the Web topology and preliminary detection results, a series of re-extracted features are computed for the second stage learning, which can be seen as the expansion of *Sgl* algorithm. With this strategy, the complete detection process can be executed in the machine learning framework. Experiments on the WEBSPAM-UK2006 benchmark shows that with both original and re-extracted features, the performance of spam detection can be improved evidently.

2. PROPOSED DETECTION STRATEGY
Figure. 1 is the flow chart of our proposed two-stage web spam detection strategy. Preliminary detection is carried out based on the original extracted features, then the predicted spamicity will be used for the result optimization. Instead of smoothing the result with heuristic rules, feature re-extraction strategy is adopted. Detection algorithm on the expanded eigenspace will be implemented in the optimization stage.

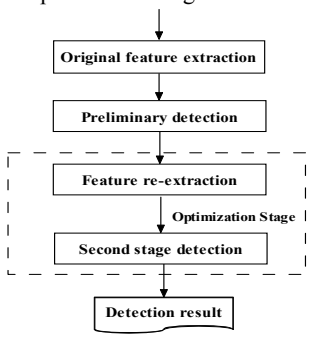

Figure 1: Flow chart of the proposed strategy.

2.1 Original Feature Extraction
Original features(*of*) are the features used for preliminary detection, which consist of hyperlink related features and content based features. Similar with [2], most of the link related features are computed for the home page and page with the maximum PageRank in each host. We don't use any linear combined features, since they are redundant from the perspective of feature selection. The content-based features consist of the number of words in page, amount of anchor text, and fraction of visible content etc [1][2]. Based on the *of*, preliminary detection is

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

carried out, and the predicted spamicity[3] will be used for features re-extraction.

2.2 Features Re-Extraction

Based on the host level link graph and the predicted spamicity, clustering features, propagation features and neighbor features are extracted.

2.2.1 Clustering Features

The host level undirected graph G is defined as $G = (V, E, w)$, where V is the set of hosts, $w = f(n)$ is a weighting function, n is the number of links between any page in host u and any page in host v, and E is the set of edges with non-zero weight. We cluster the graph G using the METIS graph partitioning algorithm. With such algorithm, we can partition all the hosts into K clusters. After partitioning, the clustering features(*cf*) can be computed as

$$cf(H) = \frac{\sum_{h \in C(H)} spamicity(h)}{|C(H)|} \quad (1)$$

where $C(H)$ is the cluster that host H belongs to, and *spamicity*(h) is the predicted spamicity of h with preliminary detection. With different $f(n)$ and K, we can get several clustering features. Here we chose $K = 1000, f(n) \in \{1, n, log(n)\}$. 3 clustering features are extracted.

2.2.2 Propagation Features

Propagation features is computed as follows:

$$pf(H)^{(t)} = \sum_{h:h->H} \frac{pf(h)^{(t-1)} \times weight(h,H)}{\sum_{g:h->g} weight(h,g)} \quad (2)$$

where t is the iterative times, $pf(h)^{(0)} = spamicity(h)$, and *weight(h,H)* is the weight of host h and H. The graph can be the forward graph, the backward graph or the bidirectional graph.

In the experiments, $weight(h,H) \in \{1, n, log(n)\}$, where n is the number of links between h and H, $t = 5$, and all the above mentioned graphs are used. 9 propagation features are extracted.

2.2.3 Neighbor Features

The neighborhood is a strong indicator about that host with respect to it being spam or non-spam. Neighbor features (*nf*) can be computed as

$$nf(H) = \frac{\sum_{h \in N(H)} (spamicity(h) \times weight(H,h))}{|N(H)|} \quad (3)$$

in which $N(H)$ represents the neighbor relation set of host H, $N(H) \in \{inlink(H), outlink(H), outlink(outlink(H)), inlink(inlink(H)), inlink(outlink(H)), outlink(inlink(H))\}$, $inlink(H)$ represents the inlink set of H, and $outlink(H)$ is the outlink set of H.

In the following experiments, for $inlink(H)$ and $outlink(H)$, $weight(H, h) \in \{1, log(n)\}$ and for all the rest neighbor relation, $weight(H, h) = 1$. Total 8 neighbor features are extracted.

2.3 Detection Algorithm

The detection algorithm we used in the experiment is bagging, a famous meta-learning algorithm. The weak classifier for bagging is C4.5.

3. EXPERIMENTS

3.1 Data Collection

WEBSPAM-UK2006 [4] is used in our experiments. The collection includes 77.9 million pages, corresponding to roughly 11400 hosts. We use all the labeled data with their home page in the summarized samples [4]. Both set1 and set2 are taken into account, where 4411 hosts are marked normal and 1803 hosts are marked spam.

3.2 Experiment Results

Six times 5-fold cross-validation is run on the data set. The precision, recall, true positive rate(TP), false positive rate(FP), area under ROC curve (AUC) and F-measure are used to measure the performance.

Table 1 shows the performance of web spam detection with different strategies. The first line is the baseline, which is computed with the original features. The second line reports the results computed with stack graph learning optimization, where both inlink and outlink relations are taken into consideration. The last line gives the performance with both original and re-extracted features. The figures in brackets are the improvement compared with the baseline. From the table, we can see that when using all the features, a big improvement can be obtained on all of the measures. The experimental results indicate that the proposed strategy can mine the topological dependencies more effectively.

Table 1. Web Spam Detection Performace with Different Strategies

Features	TP	FP	Precision	Recall	F-measure	AUC
of	0.832	0.0626	0.845	0.832	0.838	0.958
Sgl	**0.885 (6.41%)**	0.0645 (-2.98%)	**0.848 (0.47%)**	**0.885 (6.41%)**	**0.867 (3.38%)**	**0.967 (0.088%)**
of + cf + pf + nf	**0.900 (8.12%)**	**0.0606 (3.08%)**	**0.859 (1.63%)**	**0.900 (8.12%)**	**0.879 (4.80%)**	**0.971 (1.26%)**

4. CONCLUSIONS

In this paper, we detect web spam in the machine learning framework. The proposed detection strategy includes two-stage feature extraction, which makes full use of the Web topological relation. Experiment shows that the strategy is robust, and can detect web spam more effectively.

5. REFERENCES

[1] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting Spam Web Pages through Content Analysis. In Proc. of the WWW'06, May, 2006.

[2] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know Your Neighbors: Web Spam Detection Using the Web Topology. SIGIR'07, July, 2007

[3] G.G. Geng, C.H. Wang, Q.D. Li, L. Xu and X.B. Jin, Boosting the Performance of Web Spam Detection with Ensemble Under-Sampling Classification, FSKD'07 China, August, 2007.

[4] Yahoo! Research: "Web Collection UK-2006". http://research.yahoo.com/ Crawled by the Laboratory of Algorithmics, University of Milan. http://www.yr-bcn.es/webspam/ URL retrieved Jan. 2008

[5] Q.Q. Gan and Torsten Suel. Improving Web Spam Classifiers Using Link Structure. AIRWeb'07, Banff,Canada, May, 2007