Improving Web Spam Classifiers Using Link Structure

Qingqing Gan and Torsten Suel
CIS Department
Polytechnic University
Brooklyn, NY 11201, USA
qq_gan@cis.poly.edu, suel@poly.edu

ABSTRACT

Web spam has been recognized as one of the top challenges in the search engine industry [14]. A lot of recent work has addressed the problem of detecting or demoting web spam, including both content spam [16, 12] and link spam [22, 13]. However, any time an anti-spam technique is developed, spammers will design new spamming techniques to confuse search engine ranking methods and spam detection mechanisms. Machine learning-based classification methods can quickly adapt to newly developed spam techniques. We describe a two-stage approach to improve the performance of common classifiers. We first implement a classifier to catch a large portion of spam in our data. Then we design several heuristics to decide if a node should be relabeled based on the preclassified result and knowledge about the neighborhood. Our experimental results show visible improvements with respect to precision and recall.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Filtering

General Terms

Algorithms, Design, Experimentation.

Keywords

Search engines, web spam detection, classification, link analysis, machine learning, web mining.

1. INTRODUCTION

Given the large number of pages on the web, most users now rely on search engines to locate web resources. A high position in a search engine's returned results is highly valuable to commercial web sites. Aggressive attempts to obtain a higher-than-deserved position by manipulating search engine ranking methods are called search engine spamming. Besides decreasing the quality of search results, the large number of spam pages (i.e., pages explicitly created for spamming) also increases the cost of crawling, indexing, and storage in search engines.

There are a variety of spamming techniques currently in use on the web, as described in [12]. Here we discuss spam falling into one of the following two major categories - content spam and link spam. A large amount of recent work has focused on web spam, including a number of studies on link analysis methods and machine learning-based classification methods for detecting spam. For example, propagating distrust from known spam pages by reversing links [19] is believed to be used by some search engines, while [13] proposes the idea of promoting trust from good sites in order to demote spam. A study of statistical properties of spam pages in [11] showed that spam pages typically differ from non-spam pages on a number of features; this observation was subsequently used in [16] to build a classifier for detecting spam. Some recent work integrates certain link-based features, such as in-degree and out-degree distributions, into classifiers in order to discover more spam. For example, the Spamrank algorithm is implemented in [3] by using the Pagerank value distribution in the in-coming pages as one of the features in classification.

In our work, we first implement a basic (baseline) classifier and then propose two methods for enhancing this classifier by integrating additional neighborhood features. Our basic classifier consists of more than twenty features, including both content-based and link-based ones, and its performance is comparable to other machine learning-based classifiers, e.g., the one discussed in [16]. Then we present two ideas for improving the results of the basic classifier.

We call the first one *relabeling*. This method may change a site's label assigned by the basic classifier according to several features in the neighborhood of the site (where by neighborhood of a site A we mean a small subgraph cut from the sites pointing to A and the sites pointed to by A). The other method, called *secondary classifier*, takes both the results from the basic classifier and features extracted from the neighborhood as input attributes. Our experiments show that either of the two refinements obtains visible improvements compared to the basic classifier, and that the secondary classifier performs best.

The rest of the paper is organized as follows. Section 2 discusses related work on general spam techniques, classification methods to detect web spam, and trust and distrust propagation. In Section 4, we implement a classifier with both content and link features. Section 5 analyzes the distribution of spam in the neighborhood of known spam and non-spam sites. Section 6 presents the two methods for enhancing the basic classifier. Finally, Section 7 discusses some open problems for future work.

2. DISCUSSION OF RELATED WORK

Given a user query, successful search engines measure not only content relevance between the query and a candidate page, but also the position of the page according to some link-based ranking algorithm. For this reason, content spam is created in order to obtain a high relevance score, and link spam is often used to confuse link-based ranking algorithms such as PageRank [17] and HITS [15]. A taxonomy of spamming techniques is described in [12], including attacks such as keyword stuffing, link farms, invisible text, and page redirecting. Numerous studies have discussed how to automatically detect web spam or prevent search results from being overly affected by spam.

Many spam detection techniques can be described as using learning-based classification to identify spam. In [11], the au-

AIRWeb '07, May 8, 2007 Banff, Alberta, Canada.

thors show that compared to normal pages, spam pages exhibit different trends in several distributions such as the out-degree and average URL length. In subsequent work [16], they extracted several features from web sites and apply them to a machine learning-based classifier. In [1], it is shown that sites with similar site structure often have the same functionality (e.g., e-commerce site, community site, company site), thus providing another potential approach for spam detection. The features we later describe in Section 4 are inspired by this work. Another example of such a machine learning approach is [9].

Another direction of web spam research has studied link spam in terms of trust and distrust propagation. Work in [21] first finds a seed set of spam pages, and then expands it to neighboring pages in the graph. The TrustRank approach [13] proposes to propagate trust from good sites. BadRank [19] is the idea of propagating badness through inverted links, i.e., pages should be punished for pointing to bad pages. Work in [23] proposes propagating distrust through outgoing links. There are several other studies [2, 24] that investigate link-based features to identify spam. Other spam techniques, such as cloaking [22] or blog spam, have also been discussed. Detection of duplicated content, discussed in [10], can also be used to identify copied or automatically created web content.

A general observation in web search has been that properties of neighboring nodes are often correlated with those of a node itself, as, e.g., observed for page topics in [6, 8, 7]. This suggests applying similar ideas to spam detection, i.e., a node is more likely to be spam if other nodes pointing to it or pointed to by our node are also spam. This idea was discussed in [4], where measures such as co-citation are used to classify unknown pages. We also use properties of a node's neighbors in the web graph, though in a somewhat different way. Finally, very recent unpublished work in [5], encountered while preparing this paper, proposes an approach very similar to ours.

3. DATA AND EXPERIMENTAL SETUP

For our experiments, we used web sites in the Swiss `ch` top-level domain crawled in 2005 using the PolyBot web crawler [18]. This data set includes about 12 million pages located on 239,272 hosts. The pages are connected by 234 million links.

In order to build the training data set used later, we repeatedly picked random sites from these 239,272 sites and categorized them manually, until we had around 4000 spam sites and 3000 non-spam sites. After combining these with a list of 762 known spam sites made available by `search.ch`, we had 4794 sites that we know to be spam. From these, we chose a sample of 1000 sites, with half of them randomly picked from the spam sites and the other half from the non-spam sites. These 1000 nodes are used in Section 4 to train a classifier.

4. BASIC CLASSIFIER

Features. The basic classifier uses both content and link features. The content features are extracted from the pages, while link features are based on the site-level graph. To justify our site-level approach, we also checked different pages from the same site and observed that they are usually either all spam or all non-spam. For this reason, we decided to base our classifier on site-level features and links. We first extracted eight content features for each page. Then, among all pages located in one site, we select the median value for each feature to be representative for the whole site. The list of content features we used are as follows (all of these were also used in [16]):

- number of words in a page.
- average length of words in a page.
- fraction of words drawn from globally popular words.
- fraction of globally popular words used in page, measured as the number of unique popular words in a page divided by the number of words in the most popular word list.
- fraction of visible content, calculated as the aggregate length (in bytes) of all non-markup words on a page divided by the total size (in bytes) of the page.
- number of words in the page title.
- amount of anchor text in a page. This feature would help to detect pages stuffed full of links to other pages.
- compression rate of the page, using `gzip`.

The following link features were calculated for each site. These features were also used in [1].

- percentage of pages in most populated level
- top level page expansion ratio
- in-links per page
- out-links per page
- out-links per in-link
- top-level in-link portion
- out-links per leaf page
- average level of in-links
- average level of out-links
- percentage of in-links to most popular level
- percentage of out-links from most emitting level
- cross-links per page
- top-level internal in-links per page on this site
- average level of page in this site

In addition, we add three other features listed as follows.

- number of hosts in the domain. We observed that domains with many hosts have a higher probability of spam.
- ratio of pages in this host to pages in this domain.
- number of hosts on the same IP address. Often spammers register many domain names to hold spam pages.

Classification Methods. We initially trained this classifier by using the decision tree C4.5, included in Weka 3.4.4 [20]. To address the overfitting problem, we tried different values for the parameter called the confidence threshold for pruning. The resulted precision and recall scores stayed the same, while resulted decision trees show slight changes for each setting. Therefore we decided to take the default value of 0.25 for later experiments. Ten-fold cross validation is used here to evaluate the classifier. The result is described in Table 1. In addition, we show in Table 2 the results of applying a Support Vector Machine (instead of C4.5) to our training data. Here, we use the polynomial kernel and the complexity constant is set to 1. By comparing F-measures for both classes, we see that C4.5 slightly wins over SVM. We thus used C4.5 for later experiments.

	Precision	Recall	F-measure
spam	0.897	0.812	0.852
non-spam	0.882	0.925	0.903

Table 1: C4.5 Results

	Precision	Recall	F-measure
spam	0.879	0.812	0.844
non-spam	0.863	0.913	0.887

Table 2: SVM Results

5. NEIGHBORHOOD STRUCTURE OF SPAM

In this section, we look at the following question: What does a site's neighborhood look like? Our expectation is that the neighborhood is a strong indicator about that site with respect to it being spam or non-spam. An example of a site and its neighborhood is shown in Figure 1. The number next to each node represents the confidence score for the label from the basic classifier described in Section 4. The target node is marked in grey, which means it is considered spam, while some of the neighbor nodes are non-spam. (We omit incoming links to neighbors.) We are interested in the distributions of several properties of the neighbors.

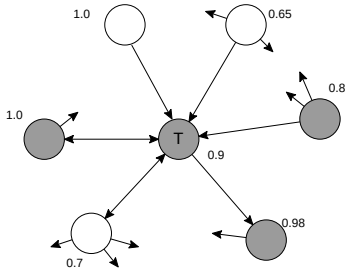

Figure 1: Neighborhood

Incoming spam distribution: We define incoming neighbors of site A as the sites directly pointing to site A. In Figure 2, a site falls into one of 12 buckets (X axis) according to the fraction of spam nodes among its incoming neighbors. The Y axis represents the percentage of total spam/non-spam sites falling into each bucket. (Thus, the site in our example would fall into the bucket for the range from 40% to 50%.) As we expected, a large portion of spam sites have predominantly spammy neighbors, while non-spam sites have more non-spam neighbors (but also some spammy neighbors). Note that we only show sites with in-degree larger than five in Figure 2.

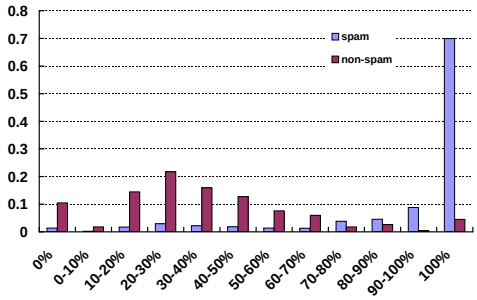

Figure 2: In-link spam distribution for spam and non-spam sites.

Outgoing spam distribution: We observe a similar, but even more pronounced, effect when looking at outgoing links. Many spam sites exclusively point to other spam, while essentially no non-spam pages point only to spam. Again, we only look at sites with out-degree larger than five.

Weighted incoming distribution: Finally, we looked at the case where each in-link is weighted by the out-degree of the pointing site; i.e., as in Pagerank, we weigh it by $1/w$ where w is the out-degree; the result is shown in Figure 4.

Note that the distributions described above are based on the judgments of the basic classifier, which means the charts may not represent the actual situation in reality. However, we believe that trend is representative, given the large number of nodes in our data set. In the following, we describe two methods to exploit these observations to improve our classifier.

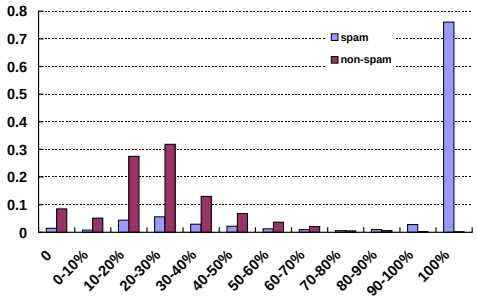

Figure 3: Out-link spam distribution

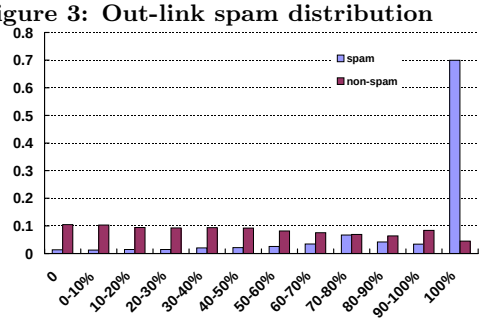

Figure 4: Weighted-in-link distribution

6. IMPROVING THE BASIC CLASSIFIER

Relabeling Approach. By relabeling we mean the process of changing the label of a site from spam to non-spam or vice versa following some rules. In particular, we first decide the label of a site's neighborhood according to one of the heuristics described further below. This label is also attached with a confidence score. We compare this label to the one we obtain from running the baseline classifier. If these two disagree with each other and the neighborhood is stronger in terms of confidence score, we flip that site's label. In any other cases, the label will stay the same. Here are the features we used to produce the neighborhood label and confidence score. Since they are the same as the ones plotted in the figures in 5, we omit detailed descriptions.

- H1: Relabeling according to the fraction X of spam sites in the total incoming neighborhood. If X is larger than 0.5, the indicated label from the neighborhood is spam with confidence X; otherwise, the indicated label is non-spam with confidence $(1 - X)$.
- H2: Relabeling according to the fraction of spam in the weighted incoming neighborhood. The label and confidence is calculated in the same way as above.
- H3: Relabeling according to the fraction of spam in the outgoing neighborhood.

To evaluate these policies, we collect the prediction for all instances in the testing sets as we train and test the baseline classifier using ten-fold cross validation in Section 4. Then we apply relabeling to this prediction. By comparing the relabeled result to the true label of a site, we compute the precision and recall scores for both classes. In Figure 5, we see improvements when using H2 or H3 (but not when using H1). A natural question is if we can do better by using all features.

Secondary Classifier Approach. A simple method to achieve this goal is to use another classifier. We present the following features to this classifier.

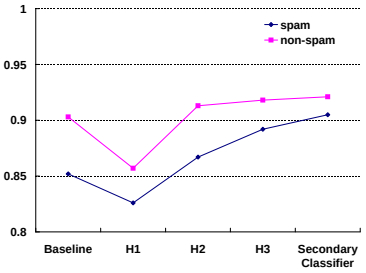

Figure 5: F-measure for different methods

- F1: The label by the basic classifier
- F2: The confidence score associated with F1
- F3: The percentage of incoming links from spam sites.
- F4: The percentage of outgoing links pointing to spam.
- F5: The fraction of weighted spam in the incoming neighbors, where the weight is proportional to the confidence score of the neighbor.
- F6: The fraction of weighted spam in the outgoing neighbors, where the weight is as in F5.
- F7: The percentage of weighted incoming spam, where the weight is given by $1/w$.

A classifier integrating all features above is implemented again by using C4.5. The results are also shown in Figure 5. The results show additional improvements compared to using only the baseline classifier or using H2 or H3.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented some preliminary results from a set of experiments on automatic detection of web spam sites. In particular, we studied how the results of a baseline classifier for this problem can be improved by adding a second-level heuristic or secondary classifier that uses the baseline classification results for neighboring sites in order to flip the labels of certain sites. Our results showed promising improvements on a large data set from the Swiss web domain.

Spam detection is an adversarial classification problem where the adversary can modify properties of the generated spam pages to avoid detection by anti-spam techniques. Possible modifications include, for instance, changing the topology of a link farm, or hiding text and links in more complicated ways. There are also many web sites whose design is optimized for search engines, but which also provide useful content. Any spam detection and demotion methods must deal with the grey area between ethical search engine optimization and unethical spam, and should give feedback on what is acceptable and what not. We believe that a semi-automatic approach mixing content features, link-based features, and end user input (e.g., data collected via a toolbar or clicks in search engine results) with actions and judgments by an experienced human operator will be better in practice.

Finally, we feel that spam detection research raises some methodological issues. Spam detection can be done on the page or site level, but very often large link farms are spread out over multiple sites and even domains. Moreover, in the case of the Swiss web domain, a few large farms are responsible for most of the spam, in terms of both pages and sites. Pages and sites within a farm are often very similar, and training sets selected at random from the entire domain are likely to contain representatives of many of the major spam farms, calling into question the underlying basis of evaluation via cross-validation. Moreover, a method that fails to detect say one of the few major farms but finds all the smaller ones may look quite bad when looking at the number of sites or pages (or even domains). On the positive side, such major farms are easy to detect due to their sheer size, and a person equipped with a suitable interactive spam detection and web mining platform should be able to first remove these large farms from the set, and then iteratively focus on other aspects of the problem.

8. REFERENCES

[1] E. Amitay, D. Carmel, A. Darlow, R. Lempel, and A. So. The connectivity sonar: Detecting site functionality by structural patterns. In *Proc. 14th ACM Conf. on Hypertext and Hypermedia*, 2003.

[2] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. Link-based characterization and detection of Web Spam. In *Workshop on Advers. Inf. Retrieval on the Web*, Aug. 2006.

[3] A. Benczur, K. Csalogany, T. Sarlos, and M. Uher. Spamrank - fully automatic link spam detection. In *Workshop on Advers. Inf. Retrieval on the Web*, 2005.

[4] A. Benczúr, K. C. T., and Sarlós. Link-based similarity search to fight web spam. In *Workshop on Advers. Inf. Retrieval on the Web*, 2006.

[5] C. Castillo, D. Donato, A. Gionis, V. Murdock, and F. Silvestri. Know your neighbors: Web spam detection using the web topology. Technical report, Yahoo! Research Barcelona, Nov. 2006.

[6] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 1998.

[7] B. Davison. Recognizing nepotistic links on the web. In *Workshop on Artificial Intelligence for Web Search*, 2000.

[8] B. Davison. Topical locality in the web. In *Proc. 23rd Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2000.

[9] I. Dorst and T. Scheffer. Thwarting the nigritude ultramarine: Learning to identify link spam. In *Proc. European Conf. on Machine Learning*, 2005.

[10] D. Fetterly, M. Manasse, and M. Najork. On the evolution of clusters of near-duplicate web pages. In *Proc. 1st Latin American Web Congress*, 2003.

[11] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics: using statistical analysis to locate spam web pages. In *Proc. 7th Int. Workshop on the Web and Databases*, pages 1–6, 2004.

[12] Z. Gyongyi and H. Garcia-Molina. Web spam taxonomy. In *Workshop on Advers. Inf. Retrieval on the Web*, 2005.

[13] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with trustrank. In *Proc. 30th VLDB*, 2004.

[14] M. Henzinger, R. Motwani, and C. Silverstein. Challenges in web search engines. *SIGIR Forum*, 36(2):11–22, 2002.

[15] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[16] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proc. 15th WWW*, pages 83–92, 2006.

[17] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.

[18] V. Shkapenyuk and T. Suel. Design and implementation of a high-performance distributed web crawler. In *Int. Conf. on Data Engineering*, 2002.

[19] M. Sobek. PR0 - Google's PageRank 0 penalty, 2002.

[20] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[21] B. Wu and B. Davison. Identifying link farm spam pages. In *Proc. 14th WWW*, May 2005.

[22] B. Wu and B. Davison. Detecting semantic cloaking on the web. In *Proc. 15th WWW*, pages 819–828, 2006.

[23] B. Wu, V. Goel, and B. Davison. Propagating trust and distrust to demote Web spam. In *Workshop on Models of Trust and the Web*, 2006.

[24] H. Zhang, A. Goel, R. Govindan, K. Mason, and B. V. Roy. Making eigenvector-based reputation systems robust to collusion. In *Proc. 3rd Workshop on Web Graphs*, 2004.