

# Web Spam Detection: link-based and content-based techniques

Luca Becchetti<sup>2</sup>, Carlos Castillo<sup>1</sup>, Debora Donato<sup>1</sup>, Stefano Leonardi<sup>2</sup>, and Ricardo Baeza-Yates<sup>1</sup>

<sup>1</sup> Yahoo! Research Barcelona  
C/Ocata 1, 08003 Barcelona  
Catalunya, SPAIN

<sup>2</sup> Università di Roma “La Sapienza”  
via Ariosto 25, 00185  
Roma, Italia

**Abstract.** The Web is both an excellent medium for sharing information as well as an attractive platform for delivering products and services. This platform is, to some extent, mediated by search engines in order to meet the needs of users seeking information. Search engines are the “dragons” that keep a valuable treasure: information [13]. Given the vast amount of information available on the Web, it is customary to answer queries with only a small set of results (typically 10 or 20 pages at most). Search engines must then **rank** Web pages, in order to create a short list of high-quality results for users.

Web spam can significantly deteriorate the quality of search engine results. Thus there is a large incentive for commercial search engines to detect spam pages efficiently and accurately. Here we present the main techniques recently introduced for Web Spam detection e demotion.

## 1 Introduction

Web Spam is a well known phenomenon for all the users that, on regular basis, browse the Web by the means of a search engine. It is an annoying experience since it forces users to load pages whose content is often completely unrelated with the query they submitted to the search engine. Nevertheless there does not exist a common definition, over which the scientific community agrees. It is indeed difficult to decide till which point we can consider “licit” the efforts devoted to increase pages rankings in the list of a search engine results. There are many and relatively easy-to-implement techniques used to attract and/or redirect traffic. We have observed a number of typical aspects that characterize Web Spam pages:

- Inclusion of many unrelated keywords and links.
- Use of many keywords in the URL.
- Redirection of the user to another page.
- Creation of many copies with substantially duplicate content.
- Insertion of hide text by writing in the same color as the background of the page.

For all the reasons we have mentioned, Web spam detection is a challenging problem. Techniques for spam detection are comprised by two main step. In the first step, all the aspects enumerated above are formalized into a set of link-based and/or content-based features. These features are used to build a classifier able to detect different kind of Web spam pages.

During the last few years a lot of work has been devoted to this task. At the beginning of our study we focused on link-based spam detection [4, 3], investigating which

features are good for discovering malicious link structures on large Web graphs. The main contributions in this area are:

- We introduce a damping function for rank propagation [1] that provides a metric that helps in separating spam from non-spam pages.
- We propose a new technique for link spam detection that exploits the distribution of the number of Web page supporters with respect to distance. To this purpose, we present an improved approximate neighborhood counting algorithm [19].
- We test several metrics as degree-degree correlations, edge-reciprocity, host-base count of neighbors, PageRank and TrustRank [16].
- We suggest an automatic classifier that only uses link attributes, without looking at Web page content, still achieving a precision that is equivalent to that of the best spam classifiers that use content analysis. This is an important point, since in many cases spam pages exhibit pretty “normal” contents.

All the experiments were done on a large sample of the .uk domain where thousands of domains have been inspected and manually classified as spam or non-spam domains by only one person (one of the authors of this paper). This sample was downloaded in 2002 by the *Dipartimento di Scienze dell’Informazione, Università degli studi di Milano*.

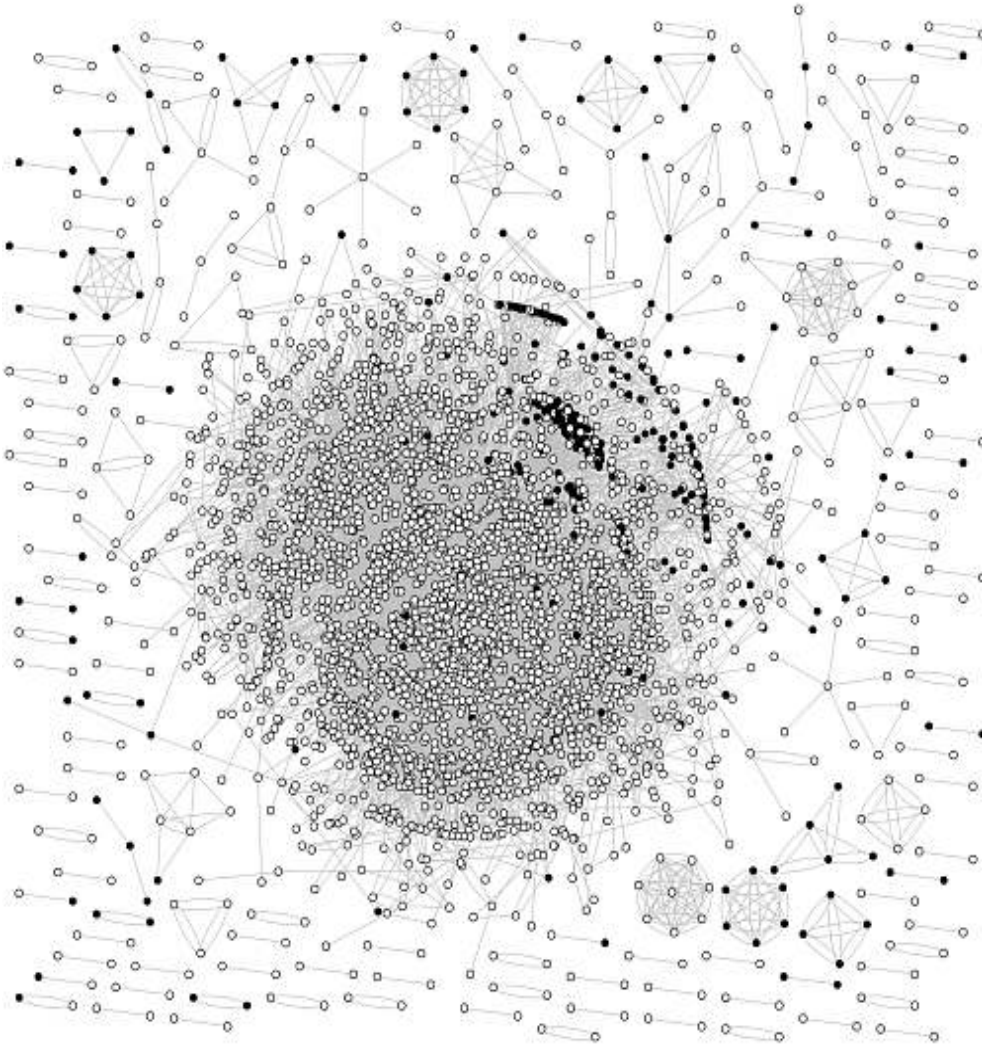
The experience gained put in evidence the need of a reference collection over which comparing all the approaches presented so far. This observation led us to build such a collection with the help of a group of volunteers recruited at the beginning of July 2006: The WEBSPAM-UK2006 is the first publicly available<sup>3</sup> Web spam collection that includes page contents and links, and that has been labeled by a large and diverse set of judges. A description of this data set can be found in [6].

In a recent work [7] we analyze the performance of an automatic classifier that combines a set of link-based and content-based features. Great improvements in the classification can be obtained exploiting the dependencies that exist among pages in the Web: links are not placed at random and in general, similar pages tend to be linked together more frequently than dissimilar ones [10].

Such a dependency holds also for spam pages and hosts: spam tends to be clustered on the Web. One explanation for this behavior is that spam pages often adopt link-based rank-boosting techniques such as link-farming. These techniques can be as simple as creating a pool of pages linking to a page whose rank is to be raised. In practice spammers use sophisticated structures that are difficult to detect.

We investigate techniques that exploit the connections between spam hosts in order to improve the accuracy of our classifiers. We assume that hosts that are well-linked together are likely to have the same class label (spam or non-spam). More generally, we can assume that two hosts in the same class should be connected by short paths going mostly through hosts in the same class. Figure 1 shows a visualization of the host graph in the WEBSPAM-UK2006 collection. An edge between two hosts is shown only if there are at least 100 links between the two hosts. In the figure, black nodes are spam and white nodes are non-spam. The layout of the nodes in the figure was computed using a spring model.

<sup>3</sup> <http://www.yr-bcn.es/webspam/datasets/>



**Fig. 1.** Graphical depiction of the hostgraph (undirected), pruned to include only labeled nodes with a connection of over 100 links between them. Black nodes are spam, white nodes are non-spam. Most of the spammers in the larger connected component are clustered together (upper-right end of the center portion). Most of the other connected components are single-class (either only spam nodes, or only non-spam nodes).

For the larger connected component of the graph, we can see that spam nodes tend to be clustered together (in the upper right corner of the central group of nodes of Figure 1). For the nodes that are not connected to this larger connected component (or are connected by links below the threshold), we can see that most of the groups are either exclusively spam, or exclusively non-spam.

The main contributions of [7] can be summarized as follows:

- To the best of our knowledge this is the first work that integrates link and content features for building a system to detect Web spam.
- We investigate the use of a cost sensitive classifier to exploit the inherent imbalance of labels in this classification problem. In the dataset we use, most of the Web content is not spam.
- We demonstrate improvements in the classification accuracy using dependencies among labels of neighboring hosts in the Web graph. We incorporate these dependencies by means of clustering and random walks.

– We apply stacked graphical learning [8] to improve the classification accuracy, exploiting the link structure among hosts in an efficient and scalable way.

The following of this documents is organized as follows.

In Section 2 we give an overview of the different definition of Web Spam. In Section 3 we characterize Web spam pages. Section 4 presents the two datasets we used in our experiments Section 5 and Section 6 describe the two link-based algorithm we introduced in [4]. In Section 7 we list the results of a number of link-based classifier over the UK2002 collection, meanwhile in Section 8 we show the results of a combined approach that uses both link-based and content-based features over the collection WEBSpam-UK2006.

## 2 Web Spam: a debatable problem

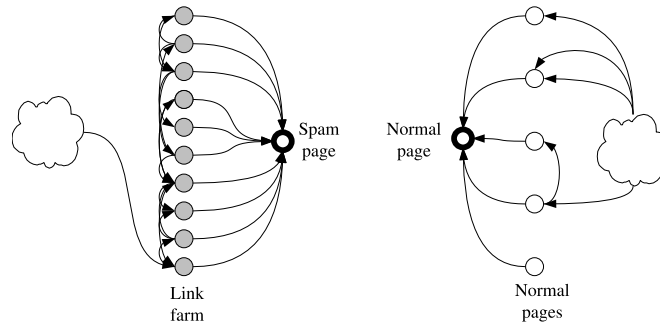
The term “**spam**” has been commonly used in the Internet era to refer to *unsolicited (and possibly commercial) bulk messages*. The most common form of electronic spam is **e-mail spam**, but in practice each new communication medium has created a new opportunity for sending unsolicited messages. The Web is not absent from this list, but as the request-response paradigm of the HTTP protocol makes it impossible for spammers to actually “send” pages directly to the users, Web spammers try to deceive search engines and thus break the trust that search engines establish with their users.

All deceptive actions which try to increase the ranking of a page in search engines are generally referred to as **Web spam** or **spamdexing** (a *portmanteau*, or combination, of “spam” and “indexing”). A **spam page or host** is a page or host that is either used for spamming or receives a substantial amount of its score from other spam pages.

An alternative way of defining Web spam could be any attempt to get “an unjustifiably favorable relevance or importance score for some web page, considering the page’s true value” [15]. A **spam page** is a page which is used for spamming or receives a substantial amount of its score from other spam pages. Another definition of spam, given in [20] is “any attempt to deceive a search engine’s relevancy algorithm” or simply “anything that would not be done if search engines did not exist”.

Seeing as there are many steps which content providers can take to improve the ranking of their Web sites, and given that there is an important subjective element in the evaluation of the relevance of Web pages, to offer an exact definition of Web spam would be misleading. Indeed, there is a large gray area between “ethical” **Search Engine Optimization** (SEO) services and “unethical” spam. SEO services range from ensuring that Web pages are indexable by Web crawlers, to the creation of thousands or millions of fake pages aimed at deceiving search engine ranking algorithms. Our main criteria for deciding in borderline cases is the perceived effort spent by Web authors on providing good content, against the effort spent on trying to score highly in search engines.

The relationship between a Web site administrator trying to rank high in a search engine and the search engine administrator is an **adversarial** one, in which any unmerited gain in ranking for a Web site results in a loss of accuracy for the search engine. This relationship is however extremely complex in nature, both because it is mediated



**Fig. 2.** Schematic depiction of the neighborhood of a page participating in a link farm (left) and a normal page (right). A link farm is a densely connected sub-graph, with little relationship with the rest of the Web, but not necessarily disconnected.

by the non univocal attitudes of customers towards spam, and because more than one form of Web spam exists which involves search engines.

### 3 Characterizing Spam Pages

There are many techniques for spamming the index of a search engine [15], and they can be broadly classified in two groups: content (or keyword) spam, and link spam.

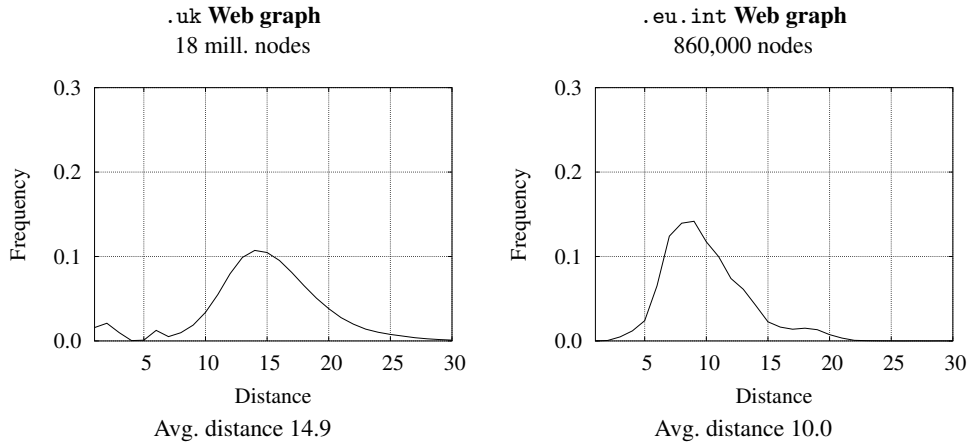
**Content spam** refers to changes in the content of the pages, for instance by inserting a large number of keywords [9, 11]. In [18], it is shown that 82-86% of spam pages of this type can be detected by an automatic classifier. The features used for the classification include, amongst others: the number of words in the text of the page, the number of hyperlinks, the number of words in the title of the pages, the compressibility (redundancy) of the content, etc.

**Link spam** includes changes into the link structure of the sites, by creating **link farms** [22, 2]. A link farm is a densely connected set of pages, created explicitly with the purpose of deceiving a link-based ranking algorithm. Zhang et. al. [22] define this form of collusion as the “manipulation of the link structure by a group of users with the intent of improving the rating of one or more users in the group”.

The targets of link-based spam-detection algorithms are the pages that receive most of their ranking by participating in link farms. A page that participates in a link farm may have a high in-degree, but little relationship with the rest of the graph. In Figure 2, we show a schematic diagram depicting the links around a spam page and a normal page. Link farms can receive links from non-spam sites by buying advertising, or by buying expired domains used previously for legitimate purposes.

Previous work in this direction includes the contribution of [12], based on “shingles”, which can be also applied in detecting some types of link farms (those that are dense graphs). It is clear that a spammer can construct link farms that exhibit statistical and spectral properties that do not differ from those of normal pages. An exclusively topological approach is clearly unfit in this case.

Link-based and content-based analysis offer two orthogonal approaches that can prove useful if combined. On one hand, in fact, link-based analysis does not capture all possible cases of spamming, since some spam pages appear to have spectral and



**Fig. 3.** Distribution of the fraction of new supporters found at varying distances (normalized), obtained by backward breadth-first visits from a sample of nodes, in two large Web graphs.

topological properties that are statistically close to those exhibited by non spam pages. In this case, content-based analysis can prove extremely useful.

On the other hand, content-based analysis seems less resilient to changes in spammers strategies, in much the same way that content-based techniques for detecting email spamming are. For instance, a spammer could copy an entire Web site (creating a set of pages that may be able to pass all tests for content spam detection) and change a few out-links in every page to point to the target page. This may be a relatively inexpensive task to perform in an automatic way, whereas creating, maintaining, reorganizing a link farm, possibly spanning more than one domain, is likely to be economically more expensive.

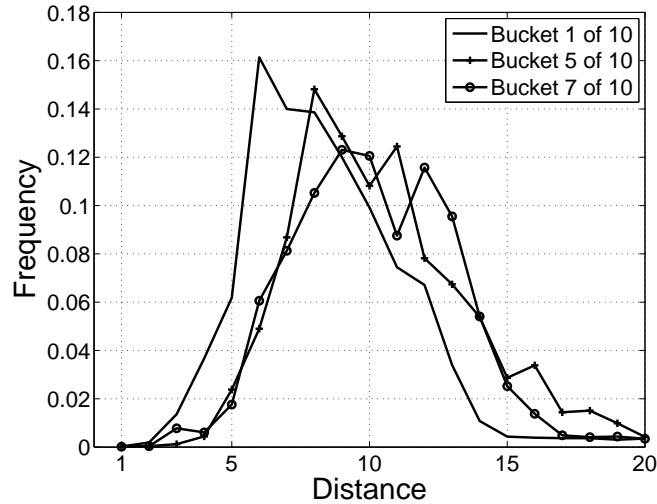
We view our set of Web pages as a **Web graph**, that is, a graph  $G = (V, E)$  in which the set  $V$  corresponds to Web pages in a subset of the Web, and every link  $(x, y) \in E$  corresponds to a hyperlink from page  $x$  to page  $y$  in the collection. For concreteness, the total number of nodes  $N = |V|$  in the full Web is in the order of  $10^{10}$  [14], and the typical number of links per Web page is between 20 and 30.

Link analysis algorithms assume that every link represents an endorsement, in the sense that if there is a link from page  $x$  to page  $y$ , then the author of page  $x$  is recommending page  $y$ . Following [5], we call  $x$  a **supporter** of page  $y$  at distance  $d$ , if the shortest path from  $x$  to  $y$  formed by links in  $E$  has length  $d$ . The set of supporters of a page are all the other pages that contribute towards its link-based ranking.

As suggested by Figure 2, a particular characteristic of a link farm is that the spam pages might have a large number of distinct supporters at short distances, but this number should be lower than expected at higher distances.

In Figure 3 we plot the distribution of distinct supporters for a random sample of nodes in two subsets of the Web obtained from the Laboratory of Web Algorithmics. (All the Web graphs we use in this paper are available from the *Dipartimento di Scienze dell'Informazione, Università degli studi di Milano* at <http://law.dsi.unimi.it/>).

We can see that the number of new distinct supporters increases up to a certain distance, between 8 and 12 links in these graphs, and then decreases, as the graph is finite in size and we approach its effective diameter. We expect that the distribution of



**Fig. 4.** Distribution of the number of new supporters at different distances, for pages in different PageRank buckets.

supporters obtained for a highly-ranked page is different from the distribution obtained for a lowly-ranked page.

To observe this, we calculated the PageRank of the pages in the `eu.int` (European Union) sub-domain. We chose this domain because it is a large, entirely spam-free, subset of the Web. We grouped the pages into 10 buckets according to their position in the list ordered by PageRank. Figure 4 plots the distribution of supporters for a sample of pages in three of these buckets having high, medium and low ranking respectively.

As expected, highly-ranked pages have a large number of supporters after a few levels, while lowly-ranked pages do not. Note that if two pages belong to the same strongly-connected component of the Web, then eventually their total number of supporters will converge after a certain distance. In that case the areas below the curves will be equal.

As shown in Figure 2, we expect that pages participating in a link-farm present anomalies in their distribution of supporters. A major issue is that computing this distribution for all the nodes in a large Web graph is computationally very expensive. A straightforward approach is to repeat a reverse breadth-first search from each node of the graph, and marking nodes as they are visited [17]; the problem is that this would require  $O(N^2)$  memory for the marks if done in parallel or  $O(N^2)$  time to repeat a BFS from each one of the  $N$  nodes if done sequentially. A possible solution could be to compute the supporters only for a subset of “suspicious” nodes; the problem with this approach is that we do not know *a priori* which nodes are spammers. An efficient solution will be presented in Section 6.

## 4 Datasets

The two link-based algorithms, presented originally in [4] and here briefly described in Section 5 and 6, are tested over the UK2002 collection. The combined classifier presented in Section 8 is trained using the WESPAM-UK2006 collection. In the following of this section, we give some details about the 2 collections

## 4.1 UK2002

This collection is a set of 18.5 million pages from the .uk domain, downloaded in 2002. The pages were located at 98,452 different hosts. Given the large size of this collection, we decided to classify entire hosts instead of individual pages. While this introduces errors, as some hosts are a mixture of non-spam and spam content, it allows us to have much broader coverage. To provide class labels for the classifier, we manually inspected a sample of 5,750 hosts (5.9%) that covered 31.7% of the Web pages in the collection.

For every host, we inspected a few pages manually and looked at the list of URLs collected from that host by the Web crawler. Whenever we found a link farm inside the host, we classified the entire host as spam. In practice, in only very few cases we observed mixtures of spam and non-spam content in the same host.

The manual classification stage was done by one of the authors of this paper and took roughly three work days. The sampling was biased towards hosts with high Page-Rank, thus following the same approach taken by other researchers in link-spam detection [5, 16]. We also marked all hosts ending in “.ac.uk” (academic domains) as normal. When tagging, we discarded the hosts that were no longer available (about 7%) and classified the remaining 5,344 hosts into 3 categories: spam, normal and suspicious. Table 1 shows the number of hosts and pages in each class.

Class	Hosts		Pages	
Spam	840	16%	329 K	6%
Normal	4,333	81%	5,429 K	92%
... Suspicious	171	3%	118 K	2%
Total	5,344 (5.8%)		5,877 K (31.7%)	

**Table 1.** Relative sizes of the classes in the manually-classified sample. The last row gives the fraction of classified hosts and pages over the entire collection.

## 4.2 WEBSpAM-UK2006

This publicly available Web spam collection [6] is based on a crawl of the .uk domain done in May 2006, including 77.9 million pages and over 3 billion links in about 11,400 hosts.

This reference collection is tagged at the host level by a group of volunteers. The assessors labeled hosts as “normal”, “borderline”, “spam” or “can not classify”, and were paired so that each sampled host was labeled by two persons independently. We obtained 6,552 evaluations. The distribution of the labels assigned by the judges is shown in Table 2. The most common label was “normal”, followed by “spam”, followed by “borderline”.

For the ground truth, we used only hosts for which the assessors agreed, plus the hosts in the collection marked as non-spam because they belong to special domains such as `police.uk` or `gov.uk`.



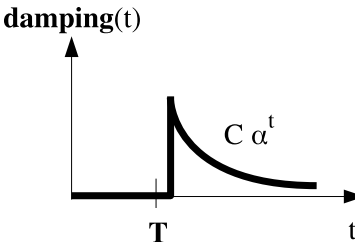
Label	Frequency	Percentage
Normal	4,046	61.75%
Borderline	709	10.82%
Spam	1,447	22.08%
Can not classify	350	5.34%

**Table 2.** Distribution of the number of pages reviewed by each judge.

## 5 Truncated PageRank

In [4] we described Truncated PageRank, a link-based ranking function that decreases the importance of neighbors that are topologically “close” to the target node. In [22] it is shown that spam pages should be very sensitive to changes in the damping factor of the PageRank calculation; in our case with Truncated PageRank we modify not only the damping factor but the whole damping function.

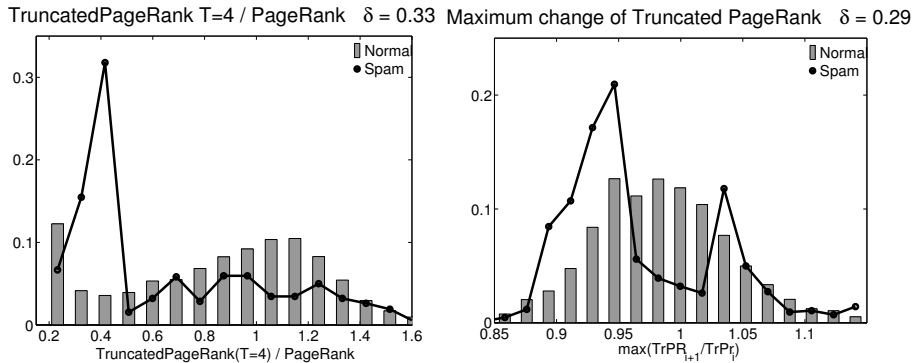
Intuitively, a way of demoting spam pages is to consider a damping function that **removes the direct contribution of the first levels of links**, such as:

$$\text{damping}(t) = \begin{cases} 0 & t \leq T \\ C\alpha^t & t > T \end{cases}$$


Where  $C$  is a normalization constant and  $\alpha$  is the damping factor used for PageRank. This function penalizes pages that obtain a large share of their PageRank from the first few levels of links; we call the corresponding functional ranking the **Truncated PageRank** of a page. The calculation of Truncated PageRank is described in detail in [4]. There is a very fast method for calculating Truncated PageRank. Given a PageRank computation, we can store “snapshots” of the PageRank values at different iterations and then take the difference and normalize those values at the end of the PageRank computation. Essentially, this means that the Truncated PageRank can be calculated for free during the PageRank iterations.

Note that as the number of indirect neighbors also depends on the number of direct neighbors, reducing the contribution of the first level of links by this method does not mean that we are calculating something completely different from PageRank. In fact, for most pages, both measures are strongly correlated, as shown in [4].

In practice, we observe that for the spam hosts in our collection, the Truncated PageRank is smaller than the PageRank, as shown in Figure 5 (left). There is a sharp peak for the spam pages in low values, meaning that many spam pages lose a large part of their PageRank when Truncated PageRank is used. We also found that studying the ratio of Truncated PageRank at distance  $i$  versus Truncated PageRank at distance  $i - 1$  also helps in identifying Web spam, as shown in Figure 5 (right). A classifier using Truncated PageRank, as well as PageRank and degree-based attributes (60 features in total) can identify 76.9% to 78.0% of the spam hosts with 1.6% to 2.5% of false positives.



**Fig. 5.** Left: histogram of the ratio between TruncatedPageRank at distance 4 and PageRank in the home page. Right: maximum ratio change of the TruncatedPageRank from distance  $i$  to distance  $i-1$ .

## 6 Estimation of supporters

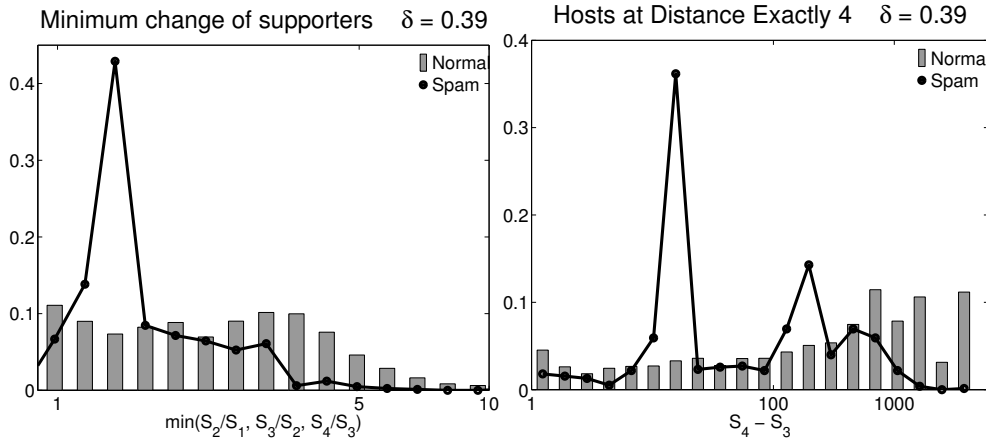
Following [5], we call  $x$  a **supporter** of page  $y$  at distance  $d$ , if the shortest path from  $x$  to  $y$  formed by links in  $E$  has length  $d$ . The set of supporters of a page are all the other pages that contribute to its link-based ranking.

A natural way of fighting link spam is to count the supporters. The naive approach is to repeat a reverse breadth-first search from each node of the graph, up to a certain depth, and mark nodes as they are visited [17]. Unfortunately, this is infeasible unless a subset of “suspicious” node is known a priori. A method for estimating the number of supporters of each node in the graph is described in [4] which improves [19].

The general algorithm (described in detail in [4]) involves the propagation of a bit mask. We start by assigning a random vector of bits to each page. We then perform an iterative computation: on each iteration of the algorithm, if page  $y$  has a link to page  $x$ , then the bit vector of page  $x$  is updated as  $x \leftarrow x \text{ OR } y$ . After  $d$  iterations, the bit vector associated to any page  $x$  provides information about the number of supporters of  $x$  at distance  $\leq d$ . Intuitively, if a page has a larger number of supporters than another, more 1s will appear in the final configuration of its bit vector.

The algorithm is described in detail in [4]. In order to have a good estimation,  $d$  passes have to be repeated  $O(\log N)$  times with different initial values, because the range of the possible values for the number of supporters is very large. We have observed that counting supporters from distances  $d$  from 1 to 4 give good results in practice. We measured how the number of supporters change at different distances, by measuring, for instance, the ratio between the number of supporters at distance 4 and the number of supporters at distance 3. The histogram for the minimum and maximum change is shown in Figure 6 (left).

This algorithm can be extended very easily to consider the number of different **hosts** contributing to the ranking of a given host. To do so, in the initialization the bit masks of all the pages in the same host have to be made equal. In Figure 6 (right), we plot the number of supporters at distance 4 considering different hosts contributing towards the ranking of the home pages of the marked hosts. We observed anomalies in this distribution for the case of the spam pages, and these anomalies are more evident by counting different hosts than by counting different pages.



**Fig. 6.** Left: histogram of the minimum change in the size of the neighborhood in the first few levels. Right: number of different hosts at distance 4

Metrics	Detection rate	False positives
Degree (D)	73-74%	2-3%
D + PageRank (P)	74-77%	2-3%
D + P + TrustRank	77%	2-3%
D + P + Trunc. PageRank	77-78%	2%
D + P + Est. Supporters	78-79%	1-2%
All attributes	80-81%	1-3%

**Table 3.** Summary of the performance of the different metrics, the ranges in the error rate correspond to a simple classifier with a few rules, and to a more complex (but more precise) classifier.

Considering distance 4, the estimation of supporters based on pages (62 attributes) yields a classifier with 78.9% to 77.9% of detection rate and 1.4% to 2.5% of false positives. If we base the estimation on hosts (67 attributes, slightly more because in-degree is not the number of neighbors at distance one in this case) allows us to build a classifier for detecting 76.5% to 77.4% of the spam with an error rate from 1.3% to 2.4%.

The detection rate is two to three percentage points lower if distance 2 is considered, with roughly the same false positives ratio.

## 7 Link-based Classifier

We test different automatic classifiers, trained used different combinations of features. We used as the base classifier the implementation of C4.5 (decision trees) given in Weka [21]. We consider several link-based groups of metrics, roughly divided in:

- Degree-based measures;
- PageRank;
- TrustRank;
- Truncated PageRank;
- Estimation of Supporters.

Table 3 presents an evaluation of all these classifiers in term of

$$\text{Detection rate} = \frac{\# \text{ of spam sites classified as spam}}{\# \text{ of spam sites}}$$

$$\text{False positives} = \frac{\text{\# of normal sites classified as spam}}{\text{\# of normal sites}}.$$

For a complete list of the features used see the Appendix of [3].

## 8 Combined Classifier

We used as the base classifier the implementation of C4.5 (decision trees) given in Weka [21]. Using both link and content features, the resulting tree used 45 unique features, of which 18 are content features.

In WEBSpAM-UK2006, the non-spam examples outnumber the spam examples to such an extent that the classifier accuracy improves by misclassifying a disproportionate number of spam examples. At the same time, intuitively, the penalty for misclassifying spam as normal is not equal to the penalty for misclassifying normal examples as spam. To minimize the misclassification error, and compensate for the imbalance in class representation in the data, we used a cost-sensitive decision tree. We imposed a cost of zero for correctly classifying the instances, and set the cost of misclassifying a spam host as normal to be  $R$  times more costly than misclassifying a normal host as spam. Table 4 shows the results for different values of  $R$ . The value of  $R$  becomes a parameter that can be tuned to balance the true positive rate and the false positive rate. In our case, we wish to maximize the F-measure. Incidentally note that  $R = 1$  is equivalent to having no cost matrix, and is the baseline classifier.

**Table 4.** Cost-sensitive decision tree

Cost ratio ( $R$ )	1	10	20	30	50
True positive rate	64.0%	68.0%	75.6%	80.1%	87.0%
False positive rate	5.6%	6.8%	8.5%	10.7%	15.4%
F-Measure	0.632	0.633	<b>0.646</b>	0.642	0.594

We then try to improve the results of the baseline classifier using bagging. Bagging is a technique that creates an ensemble of classifiers by sampling with replacement from the training set to create  $N$  classifiers whose training sets contain the same number of examples as the original training set, but may contain duplicates. The labels of the test set are determined by a majority vote of the classifier ensemble. In general, any classifier can be used as a base classifier, and in our case we used the cost-sensitive decision trees described above. Bagging improved our results by reducing the false-positive rate, as shown in Table 5. The decision tree created by bagging was roughly the same size as the tree created without bagging, and used 49 unique features, of which 21 were content features.

**Table 5.** Bagging with a cost-sensitive decision tree

Cost ratio ( $R$ )	1	10	20	30	50
True positive rate	65.8%	66.7%	71.1%	78.7%	84.1%
False positive rate	2.8%	3.4%	4.5%	5.7%	8.6%
F-Measure	0.712	0.703	0.704	<b>0.723</b>	0.692

The results of classification reported in Tables 4 and 5 use both link and content features. Table 6 shows the contribution of each type of feature to the classification. The content features serve to reduce the false-positive rate, without diminishing the true positive result, and thus improve the overall performance of the classifier. The classifier that serves as the foundation for future experiments paper uses bagging with a cost-sensitive decision tree, where  $R = 30$ .

**Table 6.** Comparing link and content features

	Both	Link-only	Content-only
True positive rate	78.7%	79.4%	64.9%
False positive rate	5.7%	9.0%	3.7%
F-Measure	<b>0.723</b>	0.659	0.683

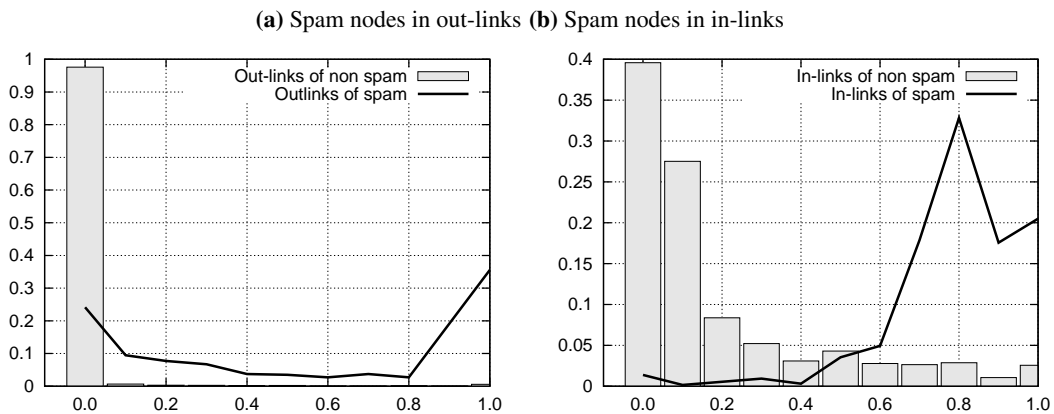
## 9 Topological dependencies of spam nodes

From our studies, we have an experimental evidence for the following two hypotheses:

Non-spam nodes tend to be linked by very few spam nodes, and usually link to non spam nodes.

Spam nodes are mainly linked by spam nodes.

Examining the out-link and the in-link graphs separately, we count the number of spam hosts contained in the adjacency list of each one of the hosts.



**Fig. 7.** Histogram of the fraction of spam hosts in the links of non-spam or spam hosts.

Let  $S_{OUT}(x)$  be the fraction of spam hosts linked by host  $x$  out of all labeled hosts linked by host  $x$ . Figure 7(a) shows the histogram of  $S_{OUT}$  for spam and non-spam hosts. We see that almost all non-spam hosts link mostly to non-spam hosts. The same is not true for spam hosts, which tend to link both spam and non-spam hosts.

Similarly, let  $S_{IN}(x)$  be the fraction of spam hosts that link to host  $x$  out of all labeled hosts that link to  $x$ . Figure 7(b) shows the histograms of  $S_{IN}$  for spam and non-spam hosts. In this case there is a clear separation between spam and non-spam hosts.

The general trend is that spam hosts are linked mostly by other spam hosts. More than 85% of the hosts have an  $S_{IN}$  value of more than 0.7. On the other hand, the opposite is true for non-spam hosts; more than 75% of the non-spam hosts have an  $S_{IN}$  value of less than 0.2.

These observations induce us to use aggregation of spam hosts to improve the accuracy of spam detection. In particular we use 3 different methods:

- Graph clustering algorithms;
- Propagation of the predicted labels;
- Stacked graphical learning.

For space reasons, we do not enter in the details of these methods and we send the reader to [7]. We want to stress that all these methods perform better than the basic classifier of Section 7. In particular, with the stacked graphical learning [8], the improvement of the F-Measure from 0.723 to 0.763 is of about 5.5%, and this actually translates to a large improvement in the accuracy of the classifier. The smoothing techniques we use improve the detection rate from 78.7% to 88.4%, while the error rate grows by less than one percentage point, from 5.7% to 6.3%.

## 10 Conclusion

Web spam detection is a challenging problem. We present a number of techniques able to detect up to 88% of spam pages. The performance of Web spam detection algorithms is still modest when compared with the error rate of modern e-mail spam detection systems. In our opinion, the current precision and recall of Web spam detection algorithms can be improved using a combination of factors already used by search engines as, for example, user interaction features (e.g. data collected via toolbar or by observing clicks in search engine results). Our future works should investigate such a possible research line.

## References

1. Ricardo Baeza-Yates, Paolo Boldi, and Carlos Castillo. Generalizing pagerank: Damping functions for link-based ranking algorithms. In *Proceedings of ACM SIGIR*, pages 308–315, Seattle, Washington, USA, August 2006. ACM Press.
2. Ricardo Baeza-Yates, Carlos Castillo, and Vicente López. Pagerank increase under different collusion topologies. In *First International Workshop on Adversarial Information Retrieval on the Web*, 2005.
3. Luca Becchetti, Carlos Castillo, Debora Donato, Stefano Leonardi, and Ricardo Baeza-Yates. Link-based characterization and detection of Web Spam. In *Second International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, Seattle, USA, August 2006.
4. Luca Becchetti, Carlos Castillo, Debora Donato, Stefano Leonardi, and Ricardo Baeza-Yates. Using rank propagation and probabilistic counting for link-based spam detection. In *Proceedings of the Workshop on Web Mining and Web Usage Analysis (WebKDD)*, Pennsylvania, USA, August 2006. ACM Press.
5. András A. Benczúr, Károly Csalogány, Tamás Sarlós, and Máté Uher. Spamrank: fully automatic link spam detection. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, Chiba, Japan, May 2005.
6. Carlos Castillo, Debora Donato, Luca Becchetti, Paolo Boldi, Stefano Leonardi, Massimo Santini, and Sebastian Vigna. A reference collection for web spam. *SIGIR Forum*, 40(2):11–24, December 2006.
7. Carlos Castillo, Debora Donato, Aristides Gionis, Vanessa Murdock, and Fabrizio Silvestri. Know your neighbors: Web spam detection using the web topology. In *Proceedings of the 30th Annual International ACM SIGIR Conference (SIGIR)*, pages 423–430, Amsterdam, Netherlands, 2007. ACM Press.
8. William W. Cohen and Zhenzhen Kou. Stacked graphical learning: approximating learning in markov random fields using very short inhomogeneous markov chains. Technical report, 2006.

9. Brian D. Davison. Recognizing nepotistic links on the Web. In *Artificial Intelligence for Web Search*, pages 23–28, Austin, Texas, USA, July 2000. AAAI Press.
10. Brian D. Davison. Topical locality in the web. In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval*, pages 272–279, Athens, Greece, 2000. ACM Press.
11. Isabel Drost and Tobias Scheffer. Thwarting the nigritude ultramarine: learning to identify link spam. In *Proceedings of the 16th European Conference on Machine Learning (ECML)*, volume 3720 of *Lecture Notes in Artificial Intelligence*, pages 233–243, Porto, Portugal, 2005.
12. David Gibson, Ravi Kumar, and Andrew Tomkins. Discovering large dense subgraphs in massive graphs. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pages 721–732. VLDB Endowment, 2005.
13. Marco Gori and Ian Witten. The bubble of web visibility. *Commun. ACM*, 48(3):115–117, March 2005.
14. Antonio Gulli and Alessio Signorini. The indexable Web is more than 11.5 billion pages. In *Poster proceedings of the 14th international conference on World Wide Web*, pages 902–903, Chiba, Japan, 2005. ACM Press.
15. Zoltán Gyöngyi and Hector Garcia-Molina. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web*, 2005.
16. Zoltán Gyöngyi, Hector G. Molina, and Jan Pedersen. Combating web spam with trustrank. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB)*, pages 576–587, Toronto, Canada, August 2004. Morgan Kaufmann.
17. R. J. Lipton and J. F. Naughton. Estimating the size of generalized transitive closures. In *VLDB '89: Proceedings of the 15th international conference on Very large data bases*, pages 165–171, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.
18. Alexandros Ntoulas, Marc Najork, Mark Manasse, and Dennis Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the World Wide Web conference*, pages 83–92, Edinburgh, Scotland, May 2006.
19. Christopher R. Palmer, Phillip B. Gibbons, and Christos Faloutsos. ANF: a fast and scalable tool for data mining in massive graphs. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 81–90, New York, NY, USA, 2002. ACM Press.
20. Alan Perkins. The classification of search engine spam. Available online at <http://www.silverdisc.co.uk/articles/spam-classification/>, September 2001.
21. Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.
22. Hui Zhang, Ashish Goel, Ramesh Govindan, Kahn Mason, and Benjamin Van Roy. Making eigenvector-based reputation systems robust to collusion. In *Proceedings of the third Workshop on Web Graphs (WAW)*, volume 3243 of *Lecture Notes in Computer Science*, pages 92–104, Rome, Italy, October 2004. Springer.