Link Spam Detection Based on Mass Estimation

October 31, 2005 (Revised: June 8, 2006) ◇ Technical Report

Zoltán Gyöngyi* zoltan@cs.stanford.edu
Computer Science Department
Stanford University, Stanford, CA 94305

Pavel Berkhin pberkhin@yahoo-inc.com
Yahoo! Inc.
701 First Avenue, Sunnyvale, CA 94089

Hector Garcia-Molina hector@cs.stanford.edu
Computer Science Department
Stanford University, Stanford, CA 94305

Jan Pedersen jpederse@yahoo-inc.com
Yahoo! Inc.
701 First Avenue, Sunnyvale, CA 94089

Abstract

Link spamming intends to mislead search engines and trigger an artificially high link-based ranking of specific target web pages. This paper introduces the concept of spam mass, a measure of the impact of link spamming on a page's ranking. We discuss how to estimate spam mass and how the estimates can help identifying pages that benefit significantly from link spamming. In our experiments on the host-level Yahoo! web graph we use spam mass estimates to successfully identify tens of thousands of instances of heavy-weight link spamming.

1 Introduction

In an era of search-based web access, many attempt to mischievously influence the page rankings produced by search engines. This phenomenon, called *web spamming*, represents a major problem to search engines [13, 10] and has negative economic and social impact on the whole web community. Initially, spammers focused on enriching the contents of spam pages with specific words that would match query terms. With the advent of link-based ranking techniques, such as PageRank [12], spammers started to construct *spam farms*, collections of interlinked spam pages. This latter form of spamming is referred to as *link spamming* as opposed to the former *term spamming*.

The plummeting cost of web publishing has rendered a boom in link spamming. The size of many spam farms has increased dramatically, and many farms span tens, hundreds, or even thousands

*Work performed during a summer internship at Yahoo! Inc.

of different domain names, rendering naïve countermeasures ineffective. Skilled spammers, whose activity remains largely undetected by search engines, often manage to obtain very high rankings for their spam pages.

This paper proposes a novel method for identifying the largest and most sophisticated spam farms, by turning the spammers' ingenuity against themselves. Our focus is on spamming attempts that target PageRank. We introduce the concept of *spam mass*, a measure of how much PageRank a page accumulates through being linked to by spam pages. The target pages of spam farms, whose PageRank is boosted by many spam pages, are expected to have a large spam mass. At the same time, popular reputable pages, which have high PageRank because other reputable pages point to them, have a small spam mass.

We estimate the spam mass of all web pages by computing and combining two PageRank scores: the regular PageRank of each page and a biased one, in which a large group of known reputable pages receives more weight. Mass estimates can then be used to identify pages that are significant beneficiaries of link spamming with high probability.

The strength of our approach is that we can identify any major case of link spamming, not only farms with regular interconnection structures or cliques, which represent the main focus of previous research (see Section 5). The proposed method also complements our previous work on TrustRank [9] in that it detects spam as opposed to "detecting" reputable pages (see Sections 3.4 and 5).

This paper is organized as follows. We start with some background material on PageRank and link spamming. The first part of Section 3 introduces the concept of spam mass through a transition from simple examples to formal definitions. Then, the second part of Section 3 presents an efficient way of estimating spam mass and a practical spam detection algorithm based on mass estimation. In Section 4 we discuss our experiments on the Yahoo! search engine index and offer evidence that spam mass estimation is helpful in identifying heavy-weight link spam. Finally, Section 5 places our results into the larger picture of link spam detection research and PageRank analysis.

2 Preliminaries

2.1 Web Graph Model

Information on the web can be viewed at different levels of granularity. For instance, one could think of the web of individual HTML pages, the web of hosts, or the web of sites. Our discussion will abstract from the actual level of granularity, and see the web as an interconnected structure of *nodes*, where nodes may be pages, hosts, or sites, respectively.

We adopt the usual graph model of the web, and use $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to denote the web graph that consists of a set $\mathcal{V}$ of nodes (vertices) and a set $\mathcal{E}$ of directed *links* (edges) that connect nodes. We disregard the exact definition of a link, although it will usually represent one or a group of hyperlinks between corresponding pages, hosts, or sites. We use unweighted links and disallow self-links.

Each node has some incoming links, or *inlinks*, and some outgoing links, or *outlinks*. The number of outlinks of a node x is its *outdegree*, whereas the number of inlinks is its *indegree*. The nodes pointed to by a node x are the *out-neighbors* of x. Similarly, the nodes pointing to x are its *in-neighbors*.

2.2 Linear PageRank

A popular discussion and research topic, PageRank as introduced in [12] is more of a concept that allows for different mathematical formulations, rather than a single clear-cut algorithm. From among the available approaches (for an overview, see [3], [4], and [5]), we adopt the linear system formulation of PageRank, which we introduce next.

At a high level, the PageRank scores assigned to web nodes correspond to the stationary probability distribution of a random walk on the web graph: Assume a hypothetical web surfer moving from node to node by following links, *ad infinitum*. Then, nodes will have PageRank scores proportional to the time the surfer spends at each.

A significant technical problem with PageRank is that on the web as it is, such a hypothetical surfer would often get stuck for some time in nodes without outlinks. Consider the *transition matrix* $\mathbf{T}$ corresponding to the web graph $\mathcal{G}$, defined as

$$T_{xy} = \begin{cases} 1/\text{out}(x), \ \text{if } (x, y) \in \mathcal{E}, \\ 0, \ \text{otherwise}, \end{cases}$$

where $\text{out}(x)$ is the outdegree of node x. Note that $\mathbf{T}$ is *substochastic*: the rows corresponding to nodes with outlinks sum up to 1, but the rows corresponding to dangling nodes are all 0. To allow for a true probabilistic interpretation, $\mathbf{T}$ has to be transformed into a *stochastic transition matrix* $\mathbf{T}'$, commonly done as follows. Consider a vector $\mathbf{v}$ of positive elements with the norm $\|\mathbf{v}\| = \|\mathbf{v}\|_1 = 1$, specifying a probability distribution. Then,

$$\mathbf{T}' = \mathbf{T} + \mathbf{d}\mathbf{v}^T ,$$

where $\mathbf{d}$ is a dangling node indicator vector:

$$d_x = \begin{cases} 1, \ \text{if } \text{out}(x) = 0 , \\ 0, \ \text{otherwise.} \end{cases}$$

This transformation corresponds to adding "virtual" links from dangling nodes to (all) other nodes on the web, which are then followed according to the probability distribution $\mathbf{v}$.

Even for $\mathbf{T}'$ it is not immediately clear whether a stationary probability distribution, and therefore unique PageRank scores exist. To guarantee a unique stationary distribution, the Markov chain corresponding to our random walk has to be *ergodic*, that is, our surfer should be able to navigate from any node to any other node. This property is satisfied if we introduce a *random jump* (also known as *teleportation*)): at each step, the surfer follows one of the links from $\mathbf{T}'$ with probability c or jumps to some random node (selected based on the probability distribution $\mathbf{v}$) with probability $(1 - c)$. The corresponding *augmented transition matrix* $\mathbf{T}''$ is defined as

$$\mathbf{T}'' = c\mathbf{T}' + (1 - c)\mathbb{1}_n \mathbf{v}^T .$$

PageRank can now be defined rigorously as the stationary distribution $\mathbf{p}$ of the random walk on $\mathbf{T}''$. In fact, $\mathbf{p}$ is the dominant eigenvector (corresponding to the eigenvalue $\lambda = 1$) of the system

$$\mathbf{p} = \mathbf{T}''^T \mathbf{p} = [c\mathbf{T}^T + c\mathbf{v}\mathbf{d}^T + (1 - c)\mathbf{v}\mathbb{1}_n^T]\mathbf{p} , \qquad (1)$$

which can be solved by using, for instance, the power iterations algorithm.

It turns out, however, that we can reach a solution following a simpler path. We make the following two observations:

1. $\mathbb{1}_n^T \mathbf{p} = \|\mathbf{p}\|$;
2. $\mathbf{d}^T \mathbf{p} = \|\mathbf{p}\| - \|\mathbf{T}^T \mathbf{p}\|$.

Hence, the PageRank equation (1) can be rewritten as the *linear* system

$$(\mathbf{I} - c\mathbf{T}^T)\,\mathbf{p} = k\mathbf{v}\,, \tag{2}$$

where $k = k(\mathbf{p}) = \|\mathbf{p}\| - c\,\|\mathbf{T}^T\,\mathbf{p}\|$ is a scalar. Notice that any particular value of k will result only in a rescaling of $\mathbf{p}$ and does not change the relative ordering of nodes. In fact, we can pick any value for k, solve the linear system and then normalize the solution to $\mathbf{p}/\|\mathbf{p}\|$ to obtain the same result as for (1). In this paper, we simply set $k = 1 - c$, so that equation (2) becomes

$$(\mathbf{I} - c\mathbf{T}^T)\mathbf{p} = (1 - c)\mathbf{v}\,. \tag{3}$$

We adopt the notation $\mathbf{p} = \mathrm{PR}(\mathbf{v})$ to indicate that $\mathbf{p}$ is the (unique) vector of PageRank scores satisfying (3) for a given $\mathbf{v}$. In general, we will allow for non-uniform random jump distributions. We even allow $\mathbf{v}$ to be unnormalized, that is $0 < \|\mathbf{v}\| \leq 1$, and leave the PageRank vector unnormalized as well. The linear system (3) can be solved, for instance, by using the Jacobi method, shown as Algorithm 1.

input : transition matrix $\mathbf{T}$, random jump vector $\mathbf{v}$, damping factor c, error bound ϵ
output: PageRank score vector $\mathbf{p}$

$i \leftarrow 0$
$\mathbf{p}^{[0]} \leftarrow \mathbf{v}$
repeat
  $i \leftarrow i + 1$
  $\mathbf{p}^{[i]} \leftarrow c\mathbf{T}^T\mathbf{p}^{[i-1]} + (1 - c)\mathbf{v}$
until $\|\mathbf{p}^{[i]} - \mathbf{p}^{[i-1]}\| < \epsilon$
$\mathbf{p} \leftarrow \mathbf{p}^{[i]}$

Algorithm 1: Linear PageRank.

A major advantage of the adopted formulation is that the PageRank scores are linear in $\mathbf{v}$: for $\mathbf{p} = \mathrm{PR}(\mathbf{v})$ and $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2$ we have $\mathbf{p} = \mathbf{p}_1 + \mathbf{p}_2$ where $\mathbf{p}_1 = \mathrm{PR}(\mathbf{v}_1)$ and $\mathbf{p}_2 = \mathrm{PR}(\mathbf{v}_2)$. Another advantage is that linear systems can be solved using various numerical algorithms, such as the Jacobi or Gauss-Seidel methods, which are regularly faster than the algorithms available for solving eigensystems (for instance, power iterations). Further details on linear PageRank are provided in [3].

2.3 Link Spamming

In this paper we focus on link spamming that targets the PageRank algorithm. PageRank is fairly robust to spamming: a significant increase in score requires a *large number* of links from low-PageRank nodes and/or some *hard-to-obtain* links from popular nodes, such as The New York Times site `www.nytimes.com`. Spammers usually try to blend these two strategies, though the former is more prevalent.

In order to better understand the *modus operandi* of link spamming, we introduce the model of a *link spam farm*, a group of interconnected nodes involved in link spamming. A spam farm has a single *target node*, whose ranking the spammer intends to boost by creating the whole structure.

A farm also contains *boosting nodes*, controlled by the spammer and connected so that they would influence the PageRank of the target. Boosting nodes are owned either by the author of the target, or by some other spammer (financially or otherwise) interested in collaborating with him/her. Commonly, boosting nodes have little value by themselves; they only exist to improve the ranking of the target. Their PageRank tends to be small, so serious spammers employ a large number of boosting nodes (occasionally, thousands of them) to trigger high target ranking.

In addition to the links within the farm, spammers may gather some external links from reputable nodes. While the author of a reputable node y is not voluntarily involved in spamming (according to our model, if he/she were, the page would be part of the farm), "stray" links may exist for a number of reason:

- Node y is a blog or message board or guestbook and the spammer manages to post a comment that includes a spam link, which then slips under the editorial radar.
- The spammer creates a *honey pot*, a spam page that offers valuable information, but behind the scenes is still part of the farm. Unassuming users might then point to the honey pot, without realizing that their link is harvested for spamming purposes.
- The spammer purchases domain names that recently expired but had previously been reputable and popular. This way he/she can profit of the old links that are still out there.

Actual link spam structures may contain several target pages, and can be thought of as *alliances* of simple spam farms [8].

In this paper, we focus on identifying target nodes x that benefit mainly from boosting: spam nodes linking to x increase x's PageRank more than reputable nodes do. Subsequent sections discuss our proposed approach and the supporting experimental results.

3 Spam Mass

3.1 Naïve Approach

In order to start formalizing our problem, let us conceptually partition the web into a set of reputable nodes $\mathcal{V}^+$ and a set of spam nodes $\mathcal{V}^-$, with $\mathcal{V}^+ \cup \mathcal{V}^- = \mathcal{V}$ and $\mathcal{V}^+ \cap \mathcal{V}^- = \emptyset$.[†] Given this partitioning, we wish to detect web nodes x that gain most of their PageRank through spam nodes in $\mathcal{V}^-$ that link to them. We will conclude that such nodes x are spam farm target nodes.

A very simple approach would be that, given a node x, we look only at its immediate in-neighbors. For the moment, let us assume that it is known whether in-neighbors of x are reputable, good nodes or spam. (We will remove this unrealistic assumption in Section 3.4.) Now we wish to infer whether x is good or spam, based on the in-neighbor information.

In a first approximation, we can simply look at the number of inlinks. If the majority of x's links comes from spam nodes, x is labeled a spam target node, otherwise it is labeled good. We call this approach our first labeling scheme. It is easy to see that this scheme often mislabels spam. To illustrate, consider the web graph in Figure 1. (Our convention is to show known good nodes filled white, known spam nodes filled black, and to-be-labeled nodes hashed gray.) As x has two

[†]In practice, such perfect knowledge is clearly unavailable. Also, what constitutes spam is often a matter of subjective judgment; hence, the real web includes a voluminous gray area of nodes that some call spam while others argue against that label. Nevertheless, our simple dichotomy will be helpful in constructing the theory of the proposed spam detection method.

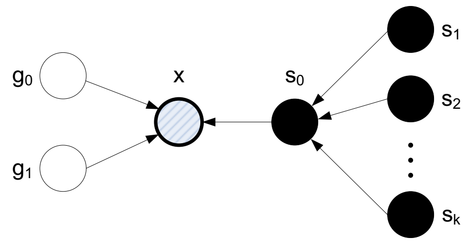

Figure 1: A scenario in which the first naïve labeling scheme fails, but the second succeeds.

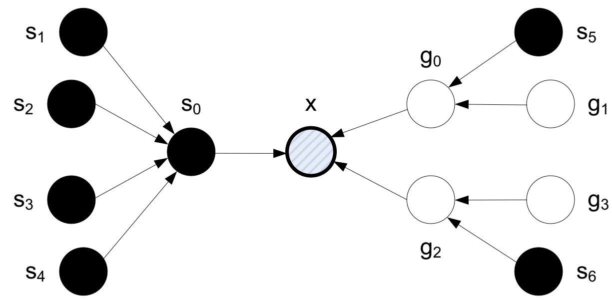

Figure 2: Another scenario in which both naïve labeling schemes fails.

links from good nodes g_0 and g_1 and a single link from spam node s_0, it will be labeled good. At the same time, solving the system of equations (1) for all nodes reveals that the PageRank of x is

$$p_x = (1 + 3c + kc^2)(1 - c)/n,$$

out of which $(c + kc^2)(1 - c)/n$ is due to spamming. (It is straightforward to verify that in the absence of spam nodes $s_0, \ldots, s_k$ the PageRank of x would decrease by this much.) For $c = 0.85$, as long as $k \geq \lceil 1/c \rceil = 2$ the largest part of x's PageRank comes from spam nodes, so it would be reasonable to label x as spam. As our first scheme fails to do so, let us come up with something better.

A natural alternative is to look not only at the number of links, but also at what amount of PageRank each link contributes. The contribution of a link amounts to the change in PageRank induced by the removal of the link. For Figure 1, links from g_0 and g_1 both contribute $c(1 - c)/n$ while the link from s_0 contributes $(c + kc^2)(1 - c)/n$. As the largest part of x's PageRank comes from a spam node, we correctly label x as spam.

However, there are cases when even our second scheme is not quite good enough. For example, consider the graph in Figure 2. The links from g_0 and g_2 contribute $(2c + 4c^2)(1 - c)/n$ to the PageRank of x, while the link from s_0 contributes $(c + 4c^2)(1 - c)/n$ only. Hence, the second scheme labels x as good. It is important, however, to realize that spam nodes s_5 and s_6 influence the PageRank scores of g_0 and g_2, respectively, and so they also have an indirect influence on the PageRank of x. Overall, the 7 spam nodes of the graph have a stronger influence on x's PageRank than the 4 reputable ones do. Our second scheme fails to recognize this because it never looks beyond the immediate in-neighbors of x.

Therefore, it is appropriate to devise a third scheme that labels node x considering all the PageRank contributions of other nodes that are directly or indirectly connected to x. The next section will show how to compute such contributions, both direct and indirect (e.g., that of s_5 to x). Then, in Section 3.3 the contributions of spam nodes will be added to determine what we call the spam mass of nodes.

3.2 PageRank Contribution

In this section we adapt some of the formalism and results introduced for *inverse P-distances* in [11].

The connection between the nodes x and y is captured by the concept of a walk. A *walk* W from x to y in a directed graph is defined as a finite sequence of nodes $x = x_0, x_1, \ldots, x_k = y$, where there is a directed edge $(x_i, x_{i+1}) \in \mathcal{E}$ between every pair of adjacent nodes x_i and x_{i+1}, $i = 0, \ldots, k-1$. The length $|W|$ of a walk W is the number $k \geq 1$ of edges. A walk with $x = y$ is called a *circuit*.

Acyclic graphs contain a finite number of walks while cyclic graph have an infinite number of walks. The (possibly infinite) set of all walks from x to y is denoted by $\mathcal{W}_{xy}$.

We define the *PageRank contribution* of x to y *over the walk* W as

$$q_y^W = c^k \pi(W)(1-c)v_x,$$

where v_x is the probability of a random jump to x, as introduced in Section 2.2, and $\pi(W)$ is the *weight* of the walk:

$$\pi(W) = \prod_{i=0}^{k-1} \frac{1}{\text{out}(x_i)} .$$

This weight can be interpreted as the probability that a Markov chain of length k starting in x reaches y through the sequence of nodes $x_1, \ldots, x_{k-1}$.

In a similar manner, we define the total PageRank contribution of x to y, $x \neq y$, over *all walks* from x to y (or simply: the PageRank contribution of x to y) as

$$q_y^x = \sum_{W \in \mathcal{W}_{xy}} q_y^W .$$

For a node's contribution to itself, we also consider an additional *virtual* circuit Z_x that has length zero and weight 1, so that

$$\begin{aligned} q_x^x = \sum_{W \in \mathcal{W}_{xx}} q_x^W &= q_x^{Z_x} + \sum_{V \in \mathcal{W}_{xx}, |V| \geq 1} q_x^V \\ &= (1-c)v_x + \sum_{V \in \mathcal{W}_{xx}, |V| \geq 1} q_x^V . \end{aligned}$$

Note that if a node x does not participate in circuits, x's contribution to itself is $q_x^x = (1-c)v_x$, which corresponds to the random jump component.

For convenience, we extend our notion of contribution even to those nodes that are unconnected. If there is no walk from node x to node y then the PageRank contribution q_y^x is zero.

The following theorem reveals the connection between the PageRank *contributions* and the PageRank *scores* of nodes. (The proofs of the theorems are provided as appendices.)

Theorem 1 *The PageRank score of a node y is the sum of the contributions of all other nodes to y:*

$$p_y = \sum_{x \in \mathcal{V}} q_y^x .$$

It is possible to compute the PageRank contribution of a node to all nodes in a convenient way, as stated next.

Theorem 2 *Under a given random jump distribution* $\mathbf{v}$*, the vector* $\mathbf{q}^x$ *of contributions of a node* x *to all nodes is the solution of the linear PageRank system for the core-based random jump vector* $\mathbf{v}^x$*:*

$$v_y^x = \begin{cases} v_x, & \text{if } x = y, \\ 0, & \text{otherwise,} \end{cases}$$

that is,

$$\mathbf{q}^x = \mathrm{PR}(\mathbf{v}^x)\,.$$

Remember that the PageRank equation system is linear in the random jump vector. Hence, we can easily determine the PageRank contribution $\mathbf{q}^{\mathcal{U}}$ of any subset of nodes $\mathcal{U} \subseteq \mathcal{V}$ by computing PageRank using the random jump vector $\mathbf{v}^{\mathcal{U}}$ defined as

$$v_y^{\mathcal{U}} = \begin{cases} v_y, & \text{if } y \in \mathcal{U}, \\ 0, & \text{otherwise.} \end{cases}$$

To verify the correctness of this last statement, note that $\mathbf{q}^x = \mathrm{PR}(\mathbf{v}^x)$ for all $x \in \mathcal{U}$ and $\mathbf{v}^{\mathcal{U}} = \sum_{x \in \mathcal{U}} \mathbf{v}^x$, therefore $\mathbf{q}^{\mathcal{U}} = \mathrm{PR}(\mathbf{v}^{\mathcal{U}}) = \sum_{x \in \mathcal{U}} \mathbf{q}^x$.

3.3 Definition of Spam Mass

Returning to the example in Figure 2, let us check whether PageRank contributions could indeed help in labeling x. We calculate and add the contributions of known good and spam nodes to the PageRank of x:

$$q_x^{\{g_0,\ldots,g_3\}} = (2c + 2c^2)(1-c)/n$$

and

$$q_x^{\{s_0,\ldots,s_6\}} = (c + 6c^2)(1-c)/n\,.$$

Then, we can decide whether x is spam based on the comparison of $q_x^{\{s_0,\ldots,s_6\}}$ to $q_x^{\{g_0,\ldots,g_3\}}$. For instance, for $c = 0.85$, $q_x^{\{s_0,\ldots,s_6\}} = 1.65 q_x^{\{g_0,\ldots,g_3\}}$. Therefore, spam nodes have more impact on the PageRank of x than good nodes do, and it might be wise to conclude that x is in fact spam. We formalize our intuition as follows.

For a given partitioning $\{\mathcal{V}^+, \mathcal{V}^-\}$ of $\mathcal{V}$ and for any node x, it is the case that $p_x = q_x^{\mathcal{V}^+} + q_x^{\mathcal{V}^-}$, that is, x's PageRank is the sum of the contributions of good nodes and that of spam nodes. (The formula includes x's contribution to itself, as we assume that we are given information about *all* nodes.)

Definition 1 *The* absolute spam mass *of* x*, denoted by* M_x*, is the PageRank contribution that* x *receives from spam nodes, that is,* $M_x = q_x^{\mathcal{V}^-}$.

Hence, the spam mass is a measure of how much direct or indirect in-neighbor spam nodes increase the PageRank of a node. Our experimental results indicate that it is suggestive to take a look at the spam mass of nodes in comparison to their total PageRank:

Definition 2 *The* relative spam mass *of* x*, denoted by* m_x*, is the* fraction *of* x*'s PageRank due to contributing spam nodes, that is,* $m_x = q_x^{\mathcal{V}^-}/p_x$.

3.4 Estimating Spam Mass

The assumption that we have accurate *a priori* knowledge of whether nodes are good (i.e., in $\mathcal{V}^+$) or spam (i.e., in $\mathcal{V}^-$) is of course unrealistic. Not only is such information currently unavailable for the actual web, but it would also be impractical to produce and would quickly get outdated. In practice, the best we can hope for is some approximation to (subset of) the good nodes (say $\tilde{\mathcal{V}}^+$) or spam nodes (say $\tilde{\mathcal{V}}^-$). Accordingly, we expect that search engines have some reliable white-list and/or black-list, comprising a subset of the nodes, compiled manually by editors and/or generated by algorithmic means.

Depending on which of these two sets is available (either or both), the spam mass of nodes can be approximated by estimating good and spam PageRank contributions.

In this paper we assume that only a subset of the good nodes $\tilde{\mathcal{V}}^+$ is provided. We call this set $\tilde{\mathcal{V}}^+$ the *good core*. A suitable good core is not very hard to construct, as discussed in Section 4.2. Note that one can expect the good core to be more stable over time than $\tilde{\mathcal{V}}^-$, as spam nodes come and go on the web. For instance, spammers frequently abandon their pages once there is some indication that search engines adopted anti-spam measures against them.

Given $\tilde{\mathcal{V}}^+$, we compute two sets of PageRank scores:

1. $\mathbf{p} = \mathrm{PR}(\mathbf{v})$, the PageRank of nodes based on the uniform random jump distribution $\mathbf{v} = (\frac{1}{n})_n$, and
2. $\mathbf{p}' = \mathrm{PR}(\mathbf{v}^{\tilde{\mathcal{V}}^+})$, a *core-based PageRank* with a random jump distribution $\mathbf{v}^{\tilde{\mathcal{V}}^+}$,

$$v_x^{\tilde{\mathcal{V}}^+} = \begin{cases} 1/n, & \text{if } x \in \tilde{\mathcal{V}}^+, \\ 0, & \text{otherwise.} \end{cases}$$

   Note that $\mathbf{p}'$ approximates the PageRank contributions that nodes receive from the good nodes on the web. The core-based PageRank is closely related to TrustRank in that both rely on a random jump distribution biased to good nodes. However, the random jump in TrustRank is biased to a small and highly selective *seed* of superior quality nodes, whereas $\tilde{\mathcal{V}}^+$

   - should be as large as possible, orders of magnitude larger than the TrustRank seed, and
   - should include as many known good nodes as possible, not only the highest quality ones.

   In Section 4.5 we discuss how major differences in core size have significant impact on the performance of mass-based spam detection.

The PageRank vectors $\mathbf{p}$ and $\mathbf{p}'$ can be used to estimate spam mass as follows:

Definition 3 *Given PageRank scores p_x and p'_x, the* estimated absolute spam mass *of node x is*

$$\tilde{M}_x = p_x - p'_x$$

and the estimated relative spam mass *of x is*

$$\tilde{m}_x = (p_x - p'_x)/p_x = 1 - p'_x/p_x.$$

As a simple example of how spam mass estimation works, consider again the graph in Figure 2 and assume that the good core is $\tilde{\mathcal{V}}^+ = \{g_0, g_1, g_3\}$. For $c = 0.85$ and $n = 12$, the PageRank score, actual absolute mass, estimated absolute mass, and corresponding relative counterparts are

	PageRank $\mathbf{p} =$	core-based PageRank $\mathbf{p}' =$	absolute mass $\mathbf{M} =$	estimated abs. mass $\tilde{\mathbf{M}} =$	relative mass $\mathbf{m} =$	estimated rel. mass $\tilde{\mathbf{m}} =$
x	9.33	2.295	6.185	7.035	0.66	0.75
g_0	2.7	1.85	0.85	0.85	0.31	0.31
g_1	1	1	0	0	0	0
g_2	2.7	0.85	0.85	1.85	0.31	0.69
g_3	1	1	0	0	0	0
s_0	4.4	0	4.4	4.4	1	1
$s_1, \ldots, s_6$	1	0	1	1	1	1

Table 1: Various features of nodes in Figure 2.

shown for each of the nodes in Table 1. Note that here, as well as in the rest of the paper, numeric PageRank scores and absolute mass values are *scaled* by $n/(1-c)$ for increased readability. Accordingly, the scaled PageRank score of a node without inlinks is 1.

For instance, the scaled PageRank score of g_0 is 2.7. Of that, $M_{g_0} = 0.85$ is contributed by spam pages, in particular by s_5. Hence, g_0's relative mass is $m_{g_0} = 0.85/2.7 = 0.31$.

The difference between actual and estimated mass can be observed in case of nodes x and g_2. Although g_2 is a good node, it is not a member of $\tilde{\mathcal{V}}^+$. Hence, both its absolute mass and relative mass are overestimated. The mass estimates for node x are also larger than the actual values.

Note that the absolute and relative mass estimates of most good nodes are small compared to the estimated mass of spam nodes. While the example in Figure 2 is overly simple, the relative separation of good and spam indicates that mass estimates could be used for spam detection purposes.

In the alternate situation that $\tilde{\mathcal{V}}^-$ is provided, the absolute spam mass can be estimated by $\hat{\mathbf{M}} = \mathrm{PR}(\mathbf{v}^{\tilde{\mathcal{V}}^-})$. Finally, when both $\tilde{\mathcal{V}}^-$ and $\tilde{\mathcal{V}}^+$ are known, the spam mass estimates could be derived, for instance, by simply computing the average $(\tilde{\mathbf{M}} + \hat{\mathbf{M}})/2$. It is also possible to invent more sophisticated combination schemes, e.g., a weighted average where the weights depend on the relative sizes of $\tilde{\mathcal{V}}^-$ and $\tilde{\mathcal{V}}^+$, with respect to the estimated sizes of $\mathcal{V}^-$ and $\mathcal{V}^+$.

3.5 Size of the Good Core

In a final step before devising a practical spam detection algorithm based on mass estimation, we need to consider a technical problem that arises for real web data.

Even with all good nodes included, one can expect for the web that the core $\tilde{\mathcal{V}}^+$ will be significantly smaller than the actual set of good nodes $\mathcal{V}^+$. That is, $|\tilde{\mathcal{V}}^+| \ll |\mathcal{V}^+|$ and thus $\|\mathbf{v}^{\tilde{\mathcal{V}}^+}\| \ll \|\mathbf{v}\|$. Note that by the definition of $\mathbf{p} = \mathrm{PR}(\mathbf{v})$ from (3), $\|\mathbf{p}\| \leq \|\mathbf{v}\|$. Similarly, $\|\mathbf{p}'\| \leq \|\mathbf{v}^{\tilde{\mathcal{V}}^+}\|$. It follows that $\|\mathbf{p}'\| \ll \|\mathbf{p}\|$, i.e., the total estimated good contribution is much smaller than the total PageRank of nodes. In this case, when estimating spam mass, we will have $\|\mathbf{p} - \mathbf{p}'\| \approx \|\mathbf{p}\|$ with only a few nodes that have absolute mass estimates differing from their PageRank scores.

A simple remedy to this problem is as follows. We can construct a (small) uniform random sample of nodes and manually label each sample node as spam or good. This way it is possible to roughly approximate the prevalence of spam nodes on the whole web. We introduce γ to denote the fraction of nodes that we estimate (based on our sample) that are good, so $\gamma n \approx |\mathcal{V}^+|$. Then, we scale the core-based random jump vector $\mathbf{v}^{\tilde{\mathcal{V}}^+}$ to $\mathbf{w}$, where

$$w_x = \begin{cases} \gamma/|\tilde{\mathcal{V}}^+|, & \text{if } x \in \tilde{\mathcal{V}}^+, \\ 0, & \text{otherwise.} \end{cases}$$

Note that $\|\mathbf{w}\| = \gamma \approx \|\mathbf{v}^{\mathcal{V}^+}\|$, so the two random jump vectors are of the same order of magnitude. Then, we can compute $\mathbf{p}'$ based on $\mathbf{w}$ and expect that $\|\mathbf{p}'\| \approx \|\mathbf{p}^{\mathcal{V}^+}\|$, so we get a reasonable estimate of the total good contribution.

Using $\mathbf{w}$ in computing the core-based PageRank leads to an interesting situation. As $\tilde{\mathcal{V}}^+$ is small, the good nodes in it will receive an unusually high random jump ($\gamma/|\tilde{\mathcal{V}}^+|$ as opposed to $1/n$). Therefore, the good PageRank contribution of these known reputable nodes will be overestimated, to the extent that occasionally for some node y, p'_y will be larger than p_y. Hence, when computing $\tilde{\mathbf{M}}$, there will be nodes with *negative spam mass*. In general, a negative mass indicates that a node is known to be good in advance (is a member of $\tilde{\mathcal{V}}^+$) or its PageRank is heavily influenced by the contribution of nodes in the good core.

3.6 Spam Detection Algorithm

Section 3.3 introduced the concept of spam mass, Section 3.4 provided an efficient way of estimating it, while Section 3.5 eliminated some technical obstacles in our way. In this section we put all pieces together and present our link spam detection algorithm based on mass estimation.

While very similar in nature, our experiments (discussed in Sections 4.4 and 4.6) indicate that relative mass estimates are more useful in spam detection than their absolute counterparts. Therefore, we build our algorithm around estimating the relative mass of nodes. Details are presented as Algorithm 2.

The first input of the algorithm is the good core $\tilde{\mathcal{V}}^+$. The second input is a threshold τ to which relative mass estimates are compared. If the estimated relative mass of a node is equal to or above this threshold then the node is labeled as a spam candidate.

The third input is a PageRank threshold ρ: we only verify the relative mass estimates of nodes with PageRank scores larger than or equal to ρ. Nodes with PageRank less than ρ are never labeled as spam candidates.

input : good core $\tilde{\mathcal{V}}^+$, relative mass threshold τ, PageRank threshold ρ
output: set of spam candidates $\mathcal{S}$

$\mathcal{S} \leftarrow \emptyset$
compute PageRank scores $\mathbf{p}$
construct $\mathbf{w}$ based on $\tilde{\mathcal{V}}^+$ and compute $\mathbf{p}'$
$\tilde{\mathbf{m}} \leftarrow (\mathbf{p} - \mathbf{p}')/\mathbf{p}$
for *each node x so that $p_x \geq \rho$* **do**
    if $m_x \geq \tau$ **then**
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{x\}$
    end
end

Algorithm 2: Mass-based spam detection.

There are at least three reasons to apply a threshold on PageRank. First, remember that we are interested in detecting nodes that profit from significant link spamming. Obviously, a node with a small PageRank is not a beneficiary of considerable boosting, so it is of no interest to us.

Second, focusing on nodes x with large PageRank also means that we have more evidence—a larger number of nodes contributing to the PageRank of x. Therefore, no single node's contribution is critical alone, the decision whether a node is spam or not is based upon data collected from multiple sources.

Finally, for nodes x with low PageRank scores, even the slightest error in approximating M_x by $\tilde{M}_x$ could yield huge differences in the corresponding relative mass estimates. The PageRank threshold helps us to avoid the complications caused by this phenomenon.

As an example of how the algorithm operates, consider once more the graph in Figure 2 with node features in Table 1. Let us assume that $\tilde{\mathcal{V}}^+ = \{g_0, g_1, g_3\}$, $\rho = 1.5$ (once again, we use *scaled* PageRank scores), $\tau = 0.5$ and $\mathbf{w} = \mathbf{v}^{\tilde{\mathcal{V}}^+}$. Then, the algorithm disregards nodes g_1, g_3 and $s_1, \ldots, s_6$ because their low PageRank of $1 < \rho = 1.5$. Again, such nodes cannot possibly benefit from significant boosting by link spamming.

Node x has PageRank $p_x = 9.33 \geq \rho = 1.5$ and a large estimated relative mass $\tilde{m}_x = 0.75 \geq \tau = 0.5$, hence it is added to the spam candidate set $\mathcal{S}$. Similarly, node s_0 is labeled spam as well. A third node, g_2 is a false positive: it has a PageRank of $p_{g_2} = 2.7$ and an estimated relative mass of $\tilde{m}_{g_2} = 0.69$, so it is labeled spam. This error is due to the fact that our good core $\tilde{\mathcal{V}}^+$ is incomplete. Finally, the other good node g_0 is correctly excluded from $\mathcal{S}$, because $\tilde{m}_{g_0} = 0.31 < \tau$.

4 Experimental Results

4.1 Data Set

To evaluate the proposed spam detection method we performed a number of experiments on actual web data. The data set that we used was based on the web index of the Yahoo! search engine as of 2004.

From the complete index of several billion web pages we extracted a list consisting of approximately 73.3 million individual web hosts.[‡]

The web graph corresponding to hosts contained slightly more than 979 million edges. These edges were obtained by collapsing all hyperlinks between any pair of pages on two different hosts into a single directed edge.

Out of the 73.3 million hosts, 25.6 million (35%) had no inlinks and 48.6 million (66.4%) had no outlinks. Reasonable explanations for the large number of hosts without outlinks are (1) the presence of URLs that never got visited by the Yahoo! spider due to the crawling policy and (2) the presence of URLs that could not be crawled because they were misspelled or the corresponding host was extinct. Some 18.9 million hosts (25.8%) were completely isolated, that is, had neither inlinks nor outlinks.

4.2 Good Core

The construction of a good core $\tilde{\mathcal{V}}^+$ represented a first step in producing spam mass estimates for the hosts. As we were aiming for a large good core, we felt that the manual selection of its members is unfeasible. Therefore, we devised a way of assembling a substantially large good core with minimal human intervention:

1. We included in $\tilde{\mathcal{V}}^+$ all hosts that appear in a small web directory which we consider being virtually void of spam. (We prefer not to disclose which directory this is in order to protect it

[‡]Web host names represent the part of the URL between the `http://` prefix and the first `/` character. Host names map to IP addresses through DNS. We did not perform alias detection, so for instance `www-cs.stanford.edu` and `cs.stanford.edu` counted as two separate hosts, even though the URLs map to the exact same content.

from infiltration attempts of spammers who might read this paper.) After cleaning the URLs (removing incorrect and broken ones), this group consisted of 16,776 hosts.

2. We included in $\tilde{\mathcal{V}}^+$ all US governmental (`.gov`) hosts (55,320 hosts after URL cleaning). Though it would have been nice to include other countries' governmental hosts, as well as various international organizations, the corresponding lists were not immediately available to us, and we could not devise a straightforward scheme for their automatic generation.

3. Using web databases (e.g., `univ.cc`) of educational institutions worldwide, we distilled a list of 3,976 schools from more than 150 countries. Based on the list, we identified 434,045 individual hosts that belong to these institutions, and included all these hosts in our good core $\tilde{\mathcal{V}}^+$.

The resulting good core consisted of 504,150 unique hosts.

4.3 Experimental Procedure

First, we computed the regular PageRank vector $\mathbf{p}$ for the host graph introduced in Section 4.1. We used an implementation of Algorithm 1 (Section 2.2).

Corroborating with earlier research reports, the produced PageRank scores follow a power-law distribution. Accordingly, most hosts have very small PageRank: slightly more than 66.7 out of the 73.3 million (91.1%) have a scaled PageRank less than 2, that is, less than the double of the minimal PageRank score. At the other end of the spectrum, only about 64,000 hosts have PageRank scores that are at least 100-times larger than the minimal. This means that the set of hosts that we focus on, that is, the set of spam targets with large PageRank, is by definition small compared to the size of the web.

Second, we computed the core-based PageRank vector $\mathbf{p}'$ using the same PageRank algorithm, but a different random jump distribution. Initially we experimented with a random jump of $1/n$ to each host in $\tilde{\mathcal{V}}^+$. However, the resulting absolute mass estimates were virtually identical to the PageRank scores for most hosts as $\|\mathbf{p}'\| \ll \|\mathbf{p}\|$.

To circumvent this problems, we decided to adopt the alternative of scaling the random jump vector to $\mathbf{w}$, as discussed in Section 3.5. In order to construct $\mathbf{w}$, we relied on the conservative estimate that at least 15% of the hosts are spam.[§] Correspondingly, we set up $\mathbf{w}$ as a uniform distribution vector over the elements of $\tilde{\mathcal{V}}^+$, with $\|\mathbf{w}\| = 0.85$.

Following the methodology introduced in Section 3.4, the vectors $\mathbf{p}$ and $\mathbf{p}'$ were used to produce the absolute and relative mass estimates of hosts ($\tilde{\mathbf{M}}$ and $\tilde{\mathbf{m}}$, respectively). We analyzed these estimates and tested the proposed spam detection algorithm. Our findings are presented next.

4.4 Relative Mass

The main results of our experiments concern the performance of Algorithm 2 presented in Section 3.6.

With relative mass values already available, only the filtering and labeling steps of the algorithm were to be performed. First, we proceeded with the PageRank filtering, using the arbitrarily selected scaled PageRank threshold $\rho = 10$. This step resulted in a set $\mathcal{T}$ of 883,328 hosts with scaled PageRank scores greater than or equal to 10. The set $\mathcal{T}$ is what we focus on in the rest of this section.

[§]In [9] we found that more than 18% of web sites are spam.

Group	1	2	3	4	5	6	7	8	9	10
Smallest $\tilde{m}$	-67.90	-4.21	-2.08	-1.50	-0.98	-0.68	-0.43	-0.27	-0.15	0.00
Largest $\tilde{m}$	-4.47	-2.11	-1.53	-1.00	-0.69	-0.44	-0.28	-0.16	-0.01	0.09
Size	44	45	43	42	43	46	45	45	46	40

Group	11	12	13	14	15	16	17	18	19	20
Smallest $\tilde{m}$	0.10	0.23	0.34	0.45	0.56	0.66	0.76	0.84	0.91	0.98
Largest $\tilde{m}$	0.22	0.33	0.43	0.55	0.65	0.75	0.83	0.90	0.97	1.00
Size	45	48	45	42	47	46	45	47	46	42

Table 2: Relative mass thresholds for sample groups.

4.4.1 Evaluation Sample

In order to evaluate the effectiveness of Algorithm 2 we constructed and evaluated a sample $\mathcal{T}'$ of $\mathcal{T}$. $\mathcal{T}'$ consisted of 892 hosts, or approximately 0.1% of $\mathcal{T}$, selected uniformly at random.

We performed a careful manual inspection of the sample hosts, searching for traces of spamming in their contents, links, and the contents of their in- and out-neighbors. As a result of the inspection, we were able to categorize the 892 hosts as follows:

- 564 hosts (63.2% of the sample) were reputable, *good* ones. The authors of the pages on these hosts refrained from using spamming techniques.
- 229 hosts (25.7%) were *spam*, that is, had some content or links added with the clear intention of manipulating search engine ranking algorithms. The unexpectedly large number of spam sample hosts indicates that the prevalence of spam is considerable among hosts with high PageRank scores. Given that earlier research results (e.g., [6], [9]) reported between 9% and 18% of spam in actual web data, it is possible that we face a growing trend in spamming.
- In case of 54 hosts (6.1%) we could not ascertain whether they were spam or not, and accordingly labeled them as *unknown*. This group consisted mainly of East Asian hosts, which represented a cultural and lingustic challenge to us. We excluded these hosts from subsequent steps of our experiments.
- 45 hosts (5%) were *non-existent*, that is, we could not access their web pages. The lack of content made it impossible to accurately determine whether these hosts were spam or not, so we excluded them from the experimental sample as well.

The first question we addressed is how good and spam hosts are distributed over the range of relative mass values. Accordingly, we sorted the sample hosts by their estimated relative mass. Then, we split the list into 20 groups, seeking a compromise between approximately equal group sizes and relevant thresholds. As shown in Table 2, the relative mass estimates of sample hosts varied between -67.90 and 1.00, and groups sizes spanned the interval 40 to 48.

Figure 3 shows the composition of each of the sample groups, after discarding non-existent and unknown hosts. The size of each group is shown on the vertical axis and is also indicated on the top of each bar. Vertically stacked bars represent the prevalence of good (white) and spam (black) sample hosts.

We decided to show separately (in gray) a specific group of *good* hosts that have high relative mass. The relative mass estimates of all these hosts were high because three very specific, isolated anomalies in our data, particularly in the good core $\tilde{\mathcal{V}}^{+}$:

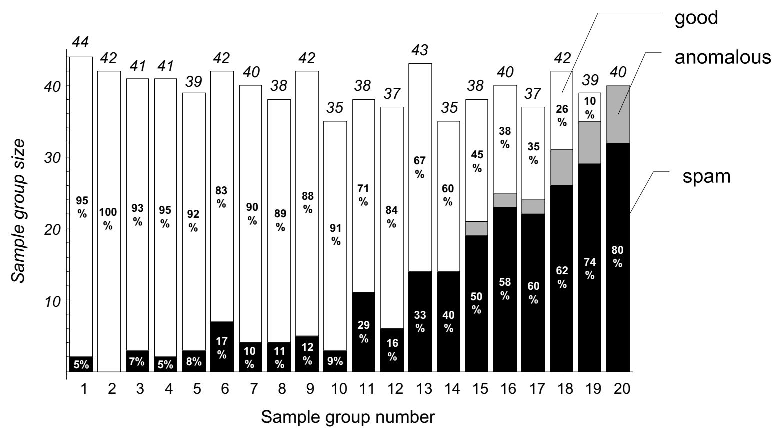

Figure 3: Sample composition.

- Five good hosts in groups 18, 19, and 20 belonged to the Chinese e-commerce site Alibaba, which encompasses a very large number of hosts, all with URLs ending in `.alibaba.com`. We believe that the reason why Alibaba hosts received high relative mass is that our good core $\tilde{\mathcal{V}}^+$ did not provide appropriate coverage of this part of the Chinese web.
- Similarly, the remaining good hosts in the last 2 groups were Brazilian blogs with URLs ending in `.blogger .com.br`. Again, this is an exceptional case of a large web community that appears to be relatively isolated from our $\tilde{\mathcal{V}}^+$.
- Finally, groups 15 through 18 contained a disproportionately large number of good hosts from the Polish web (URLs ending in `.pl`). It turns out that this is due to the incomprehensiveness of our good core: $\tilde{\mathcal{V}}^+$ only contained 12 Polish educational hosts. In comparison, $\tilde{\mathcal{V}}^+$ contained 4020 Czech (`.cz`) educational hosts, even though the Czech Republic (similar to Poland socially, politically, and in geographical location) has only one quarter of Poland's population.

4.4.2 Elimination of Anomalies

It is important to emphasize that *all* hosts in the gray group had high estimated relative mass due only to the three issues mentioned above. Accordingly, one could make adjustments to the good core in an attempt to eliminate certain anomalies and increase the prevalence of spam in groups 15–20. Some anomalies are easier to rectify, while others might require more effort.

For example, we were able to easily correct the problem with the Alibaba hosts as follows. First, we identified 12 key hosts in the `alibaba.com` domain, such as `china.alibaba.com` and `www.alibaba.com`, and added them to the good core. Then, using the new core of 504,162 hosts, we recomputed the core-biased PageRank scores and relative mass estimates of hosts. The relative mass estimates of all but the Alibaba sample hosts remained virtually the same as before (the average absolute change in relative mass was of only 0.0298 for hosts with a positive relative mass). In case of the Alibaba hosts the change was more dramatic: the relative mass estimates of the two hosts originally in group 20 decreased from 0.9989 and 0.9923 to 0.5298 and 0.3488, respectively. Similarly, the other three hosts from groups 19 and 18 ended up with relative mass estimates below 0.3. Hence, a simple expansion of the core set led to the elimination of one of the three types of anomalies.

Dealing with the other two types of anomalies was more challenging. Apparently, there are relatively few Polish educational hosts, so we had difficulty expanding our list of Polish core hosts significantly. However, an organization with more resources and better knowledge of Poland may very well be able to compile an adequately long list of Polish hosts for the good core.

The situation with Brazilian blogs was also challenging: The `blogger.com.br` domain lacks a group of easily identifiable reputable hosts that are at the center of the community. Therefore, in order to assure a proper representation of the domain in the core, one would need to analyze a fair fraction of the individual blogs and assemble a list of good ones. We believe that it would be possible for a search engine company to perform this analysis, either automatically (e.g., using statistical natural language processing) and/or by relying on white-lists provided by vigilante groups, such as Technorati.

In general, one can fully expect web search companies to gradually eliminate the few types of anomalies that may arise. They could follow a procedure similar to ours:

- First, identify good nodes with large relative mass by either sampling the results (as we did) or based on editorial or user feedback on search results;
- Next, determine the anomalies in the core that cause the large relative mass estimates of specific groups of identified good nodes;
- Finally, devise and execute correction measures for some of the anomalies, according to their relative priorities.

Figure 3 shows that even without fixing the anomalies, hosts in groups 18–20 are very likely to be spam. If the few anomalies are eliminated, high spam mass can be an almost perfect spam indicator. The key remaining issue is the selection of the threshold τ that is used in identifying link spam candidates.

4.4.3 Performance of the Detection Algorithm

We used the sample set $\mathcal{T}'$ to determine the effectiveness of our algorithm, as given by the estimated precisions $\mathrm{prec}(\tau)$, for various threshold values τ, where

$$\mathrm{prec}(\tau) = \frac{\text{number of spam sample hosts } x \text{ with } \tilde{m}_x \geq \tau}{\text{total number of sample hosts } y \text{ with } \tilde{m}_y \geq \tau}.$$

Clearly, the closer the precision is to 1 the better. We computed the precision both for the case when we accounted for the anomalous sample hosts as false positives and when we disregarded them. Figure 4 shows the two corresponding curves for relative mass thresholds between 0.98 and 0.

The horizontal axis contains the (non-uniformly distributed) threshold values that we derived from the sample group boundaries. To give a sense of the number of hosts the precision estimates apply to, the total number of hosts in $\mathcal{T}$ above each threshold is also indicated at the top of the figure. Note that because we used uniform random sampling, there is a close connection between the size of a sample group and the total number of hosts within the corresponding relative mass range: each range corresponds to roughly 45,000 hosts in $\mathcal{T}$. For instance, there are 46,635 hosts in $\mathcal{T}$ within the relative mass range 0.98 to 1, which corresponds to sample group 20.

The vertical axis stands for the (interpolated) precision. Note that precision never drops below 48%, corresponding to the estimated prevalence of spam among hosts with positive relative mass.

If we disregard the anomalous hosts, the precision of the algorithm is virtually 100% for a threshold $\tau = 0.98$. Accordingly, we expect that almost all hosts with the highest relative mass estimates are spam. The precision at $\tau = 0.91$ is still 94% with around 100,000 qualifying hosts.

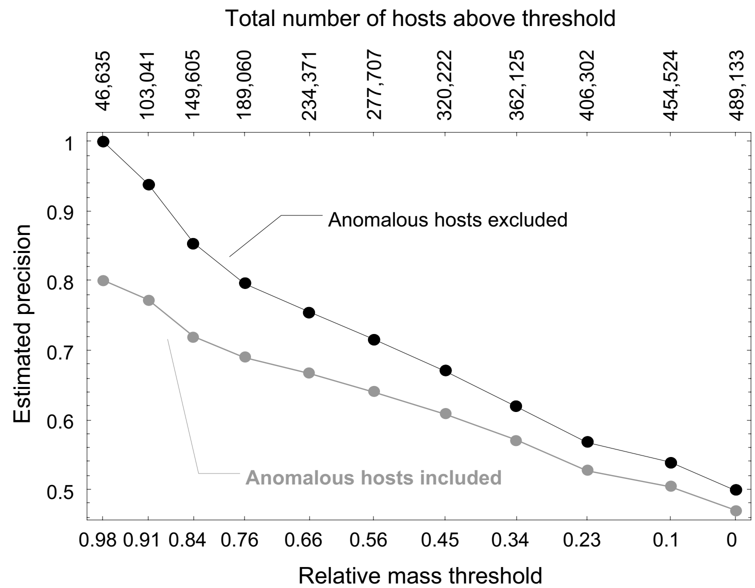

Figure 4: Precision of the mass-based spam detection algorithm for various thresholds.

Hence, we argue that our spam detection method can identify with high confidence tens of thousands of hosts that have high PageRank as a result of significant boosting by link spamming. This is a remarkably reassuring result, indicating that mass estimates could become a valuable practical tool in combating link spamming.

Beyond our basic results, we can also make a number of interesting observations about the sample composition:

1. **Isolated cliques**. Around 10% of the sample hosts with positive mass were good ones belonging to cliques only weakly connected to our good core $\tilde{\mathcal{V}}^+$. These good hosts typically were either members of some online gaming community (e.g., Warcraft fans) or belonged to a web design/hosting company. In the latter event, usually it was the case that clients linked to the web design/hosting company's site, which linked back to them, but very few or no external links pointed to either.

2. **Expired domains**. Some spam hosts had large negative absolute/relative mass values because the adopted technique of buying expired domains, already mentioned in Section 2.3. To reiterate, it is often the case that when a web domain d expires, old links from external hosts pointing to hosts in d linger on for some time. Spammers can then buy such expired domains, populate them with spam, and take advantage of the false importance conveyed by the pool of outdated links. Note that because most of the PageRank of such spam hosts is contributed by good hosts, our algorithm is not expected to detect them.

3. **Members of the good core**. The hosts from our good core received very large negative mass values because of the inherent bias introduced by the scaling of the random jump vector. Correspondingly, the first and second sample groups included 29 educational hosts and 5 governmental hosts from $\tilde{\mathcal{V}}^+$.

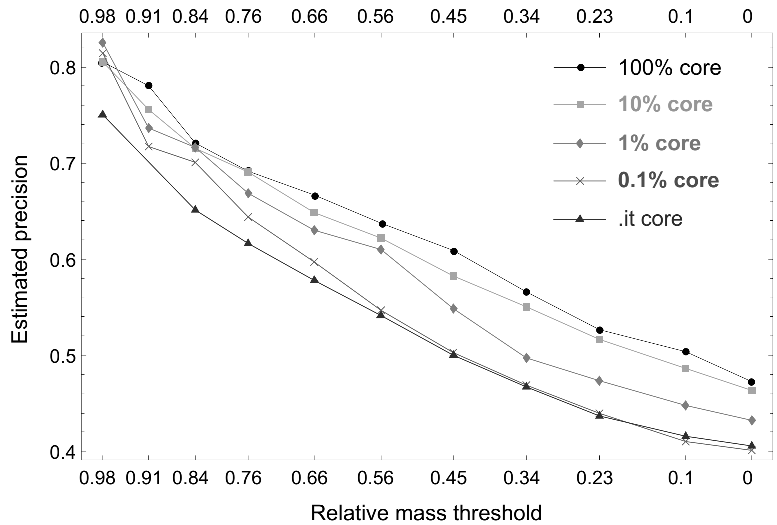

Figure 5: Precision of the mass-based spam detection algorithm for various cores.

4.5 Impact of Core Size and Coverage on Performance

It is important to understand how performance is influenced by the size and coverage of the good core. In order to investigate this issue, we generated four additional relative mass estimates for each host, in addition to the ones based on the good core $\tilde{\mathcal{V}}^+$. Three of the new estimates were derived using smaller cores: We created uniform random samples of $\tilde{\mathcal{V}}^+$ of 10% (containing 50415 hosts), 1% (5042 hosts), and 0.1% (504 hosts). These three cores are expected to have breadths of coverage similar to the original one. To understand what happens when a core covers only some parts of the entire web, for the fourth new set of estimates we arbitrarily chose to use a core containing only the 9747 Italian (`.it`) educational hosts.

In order to compare the new relative mass estimates to the original ones, we used the same evaluation sample $\mathcal{T}'$ as in Section 4.4 and Algorithm 2. The produced precision curves, shown in Figure 5, are generated in a way similar to the ones in Figure 4. The horizontal axis represents different possible relative mass thresholds between 1 and 0. The vertical axis shows the precision of the algorithm, for each threshold, as given by the fraction of spam among sample hosts with relative mass above the threshold.

First, notice that the core made up of Italian hosts only performs constantly worse than the original or any of the uniform random subsets. In particular, the precision obtained for the Italian core is less than that for the 0.1% core, which contains 19 times fewer hosts. This negative result illustrates well that the core's breadth of coverage is of paramount importance, it could matter more than the sheer size of the core.

Second, the curves corresponding to the 10%, 1%, and 0.1% cores show a gradual decline in performance. It is noteworthy that the difference between the original core's performance and that of the 10% core is not a major one, despite the change in the order of magnitude of the core size. However, the gap between $\tilde{\mathcal{V}}^+$ and the 0.1% core becomes more accentuated. Note that some smaller cores perform better at the threshold $\tau = 0.98$ than the original one. This phenomenon can be explained by the fact that the relative mass estimates for some sample spam hosts increase as the core size decreases, while the estimates for the anomalous hosts remain the same. Thus, it is not the case that the relative mass estimates for spam hosts become more accurate; on the

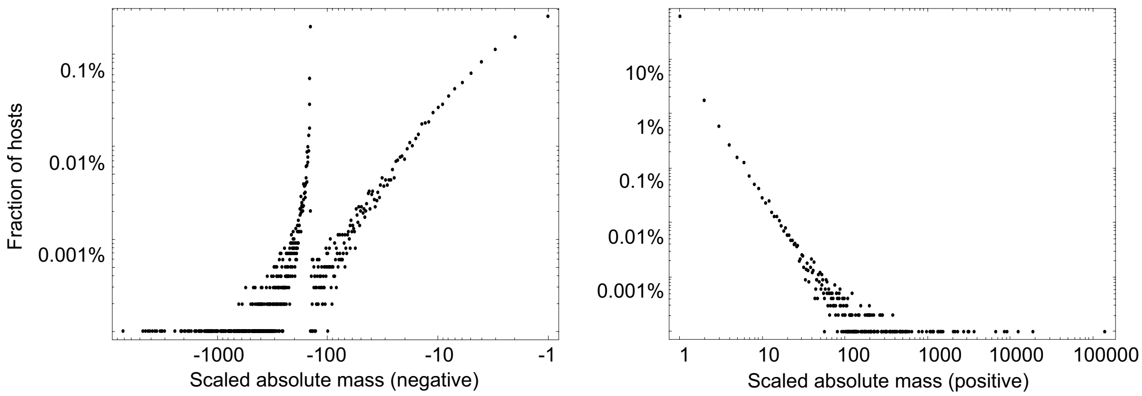

Figure 6: Distribution of estimated absolute mass values in the host-level web graph.

contrary, they become just as bad as of the anomalous good hosts.

Our experimental results on the impact of the size and coverage of the core convey a positive message: search engines interested in deploying the proposed link spam detection technique could start with relatively small cores and incrementally expand them to achieve better and better performance.

4.6 Absolute Mass

As mentioned earlier, our experiments with absolute mass were less successful than those with relative mass. Nevertheless, it is instructive to discuss some of our findings.

Mass distribution. As spam mass is a novel feature of web hosts, it is appropriate to check its value distribution. Figure 6 presents this distribution of estimated absolute mass values on a log-log scale. The horizontal axes show the range of mass values. We scaled absolute mass values by $n/(1-c)$, just as we did for PageRank scores. Hence, they fell into in the interval from -268,099 to 132,332. We were forced to split the whole range of mass estimates into two separate plots, as a single log-scale could not properly span both negative and positive values. The vertical axis shows the percentage of hosts with estimated absolute mass equal to a specific value on the horizontal axis.

We can draw two important conclusions from the figure. On one hand, positive absolute mass estimates—along with many other features of web hosts, such as indegree or PageRank—follow a power-law distribution. (For our data, the power-law exponent was -2.31.) On the other hand, the plot for negative estimated mass exhibits a combination of two superimposed curves. The right one is the "natural" distribution, corresponding to the majority of hosts. The left curve corresponds to the biased score distribution of hosts from $\tilde{\mathcal{V}}^+$ plus of those hosts that receive a large fraction of their PageRank from the good-core hosts.

Absolute mass in spam detection. A manual inspection of the absolute mass values convinced us that alone they are not appropriate for spam detection purposes. It was not a surprise to find that the host with the lowest absolute mass value was `www.adobe.com`, as its Adobe Acrobat Reader download page is commonly pointed to by various hosts. It is more intriguing, however, that `www.macromedia.com` was the host with the 3$^{\text{rd}}$ *largest* spam mass! In general, many hosts with high estimated mass were not spam, but reputable and popular. Such hosts x had an extremely large PageRank score p_x, so even a relatively small difference between p_x and p'_x rendered an absolute mass that was large with respect to the ones computed for other, less significant hosts. Hence, in the list of hosts sorted by absolute mass, good and spam hosts were intermixed without any specific mass value that could be used as an appropriate separation point.

5 Related Work

In a broad sense, our work builds on the theoretical foundations provided by analyses of PageRank (e.g., [4]). The ways in which link spamming influences PageRank are explained in [1] and [8].

A number of recent publications propose link spam detection methods. For instance, Fetterly *et al.* [6] analyze the indegree and outdegree distributions of web pages. Most web pages have in- and outdegrees that follow a power-law distribution. Occasionally, however, search engines encounter substantially more pages with the exact same in- or outdegrees than what is predicted by the distribution formula. The authors find that the vast majority of such outliers are spam pages.

Similarly, Benczúr *et al.* [2] verify for each page x whether the distribution of PageRank scores of pages pointing to x conforms a power law. They claim that a major deviation in PageRank distribution is an indicator of link spamming that benefits x.

These methods are powerful at detecting large, automatically generated link spam structures with "unnatural" link patterns. However, they fail to recognize more sophisticated forms of spam, when spammers mimic reputable web content.

Another group of work focuses on heavily interlinked groups of pages. *Collusion* is an efficient way to improve PageRank score, and it is indeed frequently used by spammers. Gibson *et al.* [7], Zhang *et al.* [15] and Wu and Davison [14] present efficient algorithms for collusion detection. However, certain reputable pages are colluding as well, so it is expected that the number of false positives returned by the proposed algorithms is large. Therefore, collusion detection is best used for penalizing all "suspicious" pages during ranking, as opposed to reliably pinpointing spam.

A common characteristic of the previously mentioned body of work is that authors focus exclusively on the link patterns between pages, that is, on *how* pages are interconnected. In contrast, this paper looks for an answer to the question "*with whom* are pages interconnected?" We investigate the PageRank of web nodes both when computed in the usual way and when determined exclusively by the links from a large pool of known good nodes. Nodes with a large discrepancy between the two scores turn out to be successfully boosted by (possibly sophisticated) link spamming.

In that we combat spam using *a priori* qualitative information about some nodes, the presented approach superficially resembles TrustRank introduced in [9]. However, there are differences between the two, which make them complementary rather than overlapping. Most importantly, TrustRank helps cleansing top ranking results by identifying reputable nodes. While spam is demoted, it is not detected—this is a gap that we strive to fill in this paper. Also, the scope of TrustRank is broader, demoting all forms of web spam, whereas spam mass estimates are effective in detecting link spamming only.

6 Conclusions

In this paper we introduced a new spam detection method that can identify web nodes with PageRank scores significantly boosted through link spamming. Our approach is built on the idea of estimating the spam mass of nodes, which is a measure of the relative PageRank contribution of connected spam pages. Spam mass estimates are easy to compute using two sets of PageRank scores—a regular one and another one with the random jump biased to some known good nodes.

Hence, we argue that the spam detection arsenal of search engines could be easily augmented with our method.

We have shown the effectiveness of mass estimation-based spam detection through a set of experiments conducted on the Yahoo! web graph. With minimal effort we were able to identify several tens of thousands of link spam hosts. While the number of detected spam hosts might seem relatively small with respect to the size of the entire web, it is important to emphasize that these are the most advanced instances of spam, capable of accumulating large PageRank scores and thus making to the top of web search result lists.

We believe that another strength of our method is that it is robust even in the event that spammers learn about it. While knowledgeable spammers could attempt to collect a large number of links from good nodes, effective tampering with the proposed spam detection method would require non-obvious manipulations of the good graph. Such manipulations are virtually impossible without knowing exactly the actual set of good nodes used as input by a given implementation of the spam detection algorithm.

In comparison to other link spam detection methods, our proposed approach excels in handling irregular link structures. It also differs from our previous work on TrustRank in that we provide an algorithm for spam detection as opposed to spam demotion.

It would be interesting to see how our spam detection method can be further improved by using additional pieces of information. For instance, we conjecture that many false positives could be eliminated by complementary (textual) content analysis. This issues remains to be addressed in future work. Also, we argue that the increasing number of (link) spam detection algorithms calls for a comparative study.

A Proof of Theorem 1

The first thing to realize is that the contribution of the unconnected nodes is zero, therefore

$$\sum_{x \in \mathcal{V}} q_y^x = \sum_{z:\mathcal{W}_{zy} \neq \emptyset} q_y^z,$$

and so the equality from the theorem becomes

$$p_y = \sum_{z:\mathcal{W}_{zy} \neq \emptyset} q_y^z.$$

In order to prove that this equality holds we will make use of two lemmas. The first shows that the total PageRank contribution to y is a solution in y to the linear PageRank equation, while the second shows that the solution is unique.

Lemma 1 $\sum_{z:\mathcal{W}_{zy} \neq \emptyset} q_y^z$ *is a solution to the linear PageRank equation of* y.

Proof. The linear PageRank equation for node y has the form:

$$p_y = c \sum_{x:(x,y) \in \mathcal{E}} p_x / \text{out}(x) + (1-c) v_y \,. \tag{4}$$

Assuming that y has k in-neighbors $x_1, x_2, \ldots, x_k$, (4) can be written as

$$p_y = c \sum_{i=1}^{k} p_{x_i} / \text{out}(x_i) + (1-c) v_y \,.$$

Let us now replace p_y and p_{x_i} on both sides by the corresponding contributions:

$$\sum_{z:\mathcal{W}_{zy}\neq\emptyset} q_y^z = c\sum_{i=1}^{k}\left(\sum_{z:\mathcal{W}_{zx_i}\neq\emptyset} q_{x_i}^z\right)/\mathrm{out}(x_i) + (1-c)v_y.$$

and expand the terms q_y^z and $q_{x_i}^z$:

$$\sum_{z:\mathcal{W}_{zy}\neq\emptyset}\left(\sum_{W\in\mathcal{W}_{zy}} c^{|W|}\pi(W)(1-c)v_z\right) =$$
$$c\sum_{i=1}^{k}\left[\sum_{z:\mathcal{W}_{zx_i}\neq\emptyset}\left(\sum_{V\in\mathcal{W}_{zx_i}} c^{|V|}\pi(V)(1-c)v_z\right)\right]/\mathrm{out}(x_i) + (1-c)v_y. \quad (5)$$

Note that $x_1,\ldots,x_k$ are all the in-neighbors of y, so $\mathcal{W}_{zy}$ can be partitioned into[¶]:

$$\mathcal{W}_{zy} = \left(\bigcup_{i=1}^{k}\{V.y \,|\, V\in\mathcal{W}_{zx_i}\}\right)\cup Z_y$$

where Z_y is the zero-length circuit. Also note that for all $V\in\mathcal{W}_{zx_i}$, $|W| = |V|+1$ and $\pi(W) = \pi(V)/\mathrm{out}(x_i)$. Hence, we can rewrite the left side of (5) to produce the equation

$$\sum_{i=1}^{k}\left[\sum_{z:\mathcal{W}_{zx_i}\neq\emptyset}\left(\sum_{V\in\mathcal{W}_{zx_i}} c^{|V|+1}\pi(W)(1-c)v_z/\mathrm{out}(x_i)\right)\right] + (1-c)v_y =$$
$$c\sum_{i=1}^{k}\left[\sum_{z:\mathcal{W}_{zx_i}\neq\emptyset}\left(\sum_{V\in\mathcal{W}_{zx_i}} c^{|V|}\pi(V)(1-c)v_z\right)\right]/\mathrm{out}(x_i) + (1-c)v_y,$$

which is an identity. Hence, q_y is a solution to the PageRank equation of y. ∎

Lemma 2 *The linear PageRank equation system has a unique solution.*

Proof. As self-loops are not allowed in the matrix $\mathbf{T}$, the matrix $\mathbf{U} = (\mathbf{I} - c\mathbf{T}^T)$ will have 1's on the diagonal and values ≤ 1 in all non-diagonal positions. Therefore $\mathbf{U}$ is diagonally dominant, and hence positive definite. Accordingly, the system $\mathbf{U}\,\mathbf{p} = k\,\mathbf{v}$ has a unique solution. ∎

B Proof of Theorem 2

According to Theorem 1, for the core-based random jump vector $\mathbf{v}^x$ the PageRank score p_z of an arbitrary node z is the sum of PageRank contributions to z:

$$p_z = \sum_{y\in\mathcal{V}} q_z^y = \sum_{y\in\mathcal{V}}\sum_{W\in\mathcal{W}_{yz}} c^{|W|}\pi(W)(1-c)v_y^x.$$

[¶]Notation $V.y$: Given some walk $V = t_0, t_1, \ldots, t_m$ of length m and a node y with $(t_m, y)\in\mathcal{E}$ we can *append* y to V and construct a new walk $V.y = t_0, t_1, \ldots, t_m, y$ of length $m+1$.

In our special case we have $v^x_y = 0$ for all $y \neq x$, so p_z is in fact

$$p_z = \sum_{W \in \mathcal{W}_{xz}} c^{|W|} \pi(W)(1-c) v^x_x = q^x_z.$$

It follows that we can determine the contributions q^x_z of x to all nodes z by computing the PageRank scores corresponding to the core-based random jump vector $\mathbf{v}^x$.

References

[1] Ricardo Baeza-Yates, Carlos Castillo, and Vicente López. PageRank increase under different collusion topologies. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005.

[2] András 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 (AIRWeb)*, 2005.

[3] Pavel Berkhin. A survey on PageRank computing. *Internet Mathematics*, 2(1), 2005.

[4] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside PageRank. *ACM Transactions on Internet Technology*, 5(1), 2005.

[5] Nadav Eiron, Kevin McCurley, and John Tomlin. Ranking the web frontier. In *Proceedings of the 13th International Conference on World Wide Web*, 2004.

[6] Dennis Fetterly, Mark Manasse, and Marc Najork. Spam, damn spam, and statistics. In *Proceedings of the 7th International Workshop on the Web and Databases (WebDB)*, 2004.

[7] David Gibson, Ravi Kumar, and Andrew Tomkins. Discovering large dense subgraphs in massive graphs. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, 2005.

[8] Zoltán Gyöngyi and Hector Garcia-Molina. Link spam alliances. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, 2005.

[9] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, 2004.

[10] Monika Henzinger, Rajeev Motwani, and Craig Silverstein. Challenges in web search engines. *ACM SIGIR Forum*, 36(2), 2002.

[11] Glen Jeh and Jennifer Widom. Scaling personalized web search. In *Proceedings of the 12th International Conference on World Wide Web*, 2003.

[12] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford University, California, 1998.

[13] Amit Singhal. Challenges in running a commercial web search engine. IBM's Second Search and Collaboration Seminar, 2004.

[14] Baoning Wu and Brian Davison. Identifying link farm spam pages. In *Proceedings of the 14th International Conference on World Wide Web*, 2005.

[15] Hui Zhang, Ashish Goel, Ramesh Govindan, Kahn Mason, and Benjamin Van Roy. Making eigenvector-based reputation systems robust to collusion. In *Proceedings of the 3rd International Workshop on Algorithms and Models for the Web-Graph (WAW)*, 2004.