SpamRank – Fully Automatic Link Spam Detection*
Work in progress

András A. Benczúr[1,2] Károly Csalogány[1,2] Tamás Sarlós[1,2] Máté Uher[1]

[1] Computer and Automation Research Institute, Hungarian Academy of Sciences (MTA SZTAKI)
11 Lagymanyosi u., H–1111 Budapest, Hungary
[2] Eötvös University, Budapest
{benczur, cskaresz, stamas, umate}@ilab.sztaki.hu
www.ilab.sztaki.hu/websearch

Abstract

Spammers intend to increase the PageRank of certain spam pages by creating a large number of links pointing to them. We propose a novel method based on the concept of personalized PageRank that detects pages with an undeserved high PageRank value without the need of any kind of white or blacklists or other means of human intervention. We assume that spammed pages have a biased distribution of pages that contribute to the undeserved high PageRank value. We define SpamRank by penalizing pages that originate a suspicious PageRank share and personalizing PageRank on the penalties. Our method is tested on a 31 M page crawl of the `.de` domain with a manually classified 1000-page stratified random sample with bias towards large PageRank values.

1 Introduction

Identifying and preventing spam was cited as one of the top challenges in web search engines in a 2002 paper [24]. Amit Singhal, principal scientist of Google Inc. estimated that the search engine spam industry had a revenue potential of $4.5 billion in year 2004 if they had been able to completely fool all search engines on all commercially viable queries [36]. Due to the large and ever increasing financial gains resulting from high search engine ratings, it is no wonder that a significant amount of human and machine resources are devoted to artificially inflating the rankings of certain web pages.

In this paper we concentrate on identifying pages backlinked by a large amount of other pages in order to mislead search engines to rank their target higher. Our main goal is to compute for each Web page a SpamRank value that measures the amount of the undeserved PageRank [32] of a page. Note that by the nature of our methods we make no distinction between fair or malicious intent and our algorithm will likely rank pages with a large amount of low quality backlinks as spam.

In order to understand the nature of link spam, we first consider the characterization of an "honest" link by Chakrabati et al. [11]:

> "hyperlink structure contains an enormous amount of latent human annotation that can be extremely valuable for automatically inferring notions of authority."

*Support from NKFP-2/0017/2002 project Data Riddle and various ETIK, OTKA and AKP grants

Note that their notion of an authority plays similar role as a page with high PageRank value. In this sense the existence of a hyperlink should affect ranking only if it expresses human annotation. As examples for different uses of hyperlinks we refer to the article of Davison [13] that shows how to determine intra-site links that may serve both navigational and certain spam purposes. In [22] more examples are given, among others spamming guest books with links or mirroring with the sole purpose of the additional links to spam targets.

We believe that identifying email spam or web content spam by *human inspection* is relative easy and automated methods cannot, in any case, perform as good as human judgement. Györgyi and Garcia-Molina [22] list a few methods that confuse users including term hiding (background color text); cloaking (different content for browsers and search engine robots) and redirection; some of these techniques can still be found by inspecting the HTML code within the page source. Detecting redirection may already require certain expertise: we found quite a number of doorway spam pages which used obfuscated JavaScript code to redirect to their target.

Web link spam, in contrast, appears to be much harder to catch. As an example of a page that we ranked high for spam, 2way.handyfritz.de looks like an "innocent" site for mobile logos and tones while a large number of its backlinks are in fact spam messages in guestbooks. As a side effect we penalize pages often cited in blogs and lists, for example to www.golem.de/0501/35735.html. Certain forms of a link spam are however visible to Web users as well: the `thema-*.de` clique contains no useful content, only long list of links to itself and to various eBay.de auctions and pop-up ads fill its pages. The misuse of the Scout24.de affiliate program is also quite popular among the German spammers.

We also see frequent examples of non-spam with undeserved PageRank. For example for affiliate pages, link spamming is not always the purpose, but it is always a side-effect as they get paid to redirect traffic to certain sites. Since we make no analysis of content, we cannot adopt a more restrictive notion of spam favored by the search engine optimization community. Depending on the actual query where a page receives high PageRank, one may refer to techniques to attain the high rank as *boosting* if the page content is otherwise relevant to the query, see e.g. [22]. Our method, in this sense, could be called de-boosting with the purpose of assisting users to find pages where the maintainer has a lower budget on Search Engine Optimization (SEO).

Preferred notions of Web link or content spam could be the following. In [22] a content is called spam if it is completely irrelevant for the given search terms. However when implementing a search engine one aims to use ranking methods that select the most relevant highest quality content irrespective to the amount of search engine optimization for similar, but possibly lower quality content. In another definition [34] search engine spam consists of features that maintainers would not add to their sites if search engines didn't exist. Link spam however could act even without the presence of a search engine by misleading users to visit certain pages, in particular for the purpose of misusing affiliate programs.

An example of a site with quality content that we penalize high is city-map.de. Such sites are distributed to so many pages across a large number of domain names that attain undeserved PageRank for these pages. While methods such as HostRank [17] and the site boundary detection of Davison [13] also handle these sites, we also notice their undeserved PageRank value by giving penalties for these sites. The natural idea of combining methods is beyond the scope of the current report.

1.1 Our method: Where does your PageRank come from?

We assume that spammed pages have a biased distribution of *supporter* pages that contribute to the undeserved high PageRank value. As described in detail in Section 2, a node's PageRank is equal to the average of its personalized PageRank where personalization is over all Web pages. By the recent algorithm of [21]

Algorithm 1: Overall Structure of SpamRank Algorithm

for all Web pages i **do**
  Support$_{i,\cdot}$ $\leftarrow$ empty sparse vector of reals
Phase 1: Supporter Generation
generate nodes into vector Support$_{i,\cdot}$ that have high contribution in the PageRank of i
Phase 2: Penalization
for all Web pages i **do**
  give Penalty$_i$ based on irregular behavior of PageRank over Support$_{i,\cdot}$
Phase 3: SpamRank as Personalized PageRank (PPR) over Penalties
SpamRank $\leftarrow$ PPR(Penalty)

we are able to approximately compute all these values[1] in order to deduce a large fraction of the origin of the node's PageRank value.

PageRank distribution in your neighborhood: looks honest or spam? Our key assumption is that supporters of an honest page should not be overly dependent on one another, i.e. they should be spread across sources of different quality. Just as in the case of the entire Web, the PageRank distribution of an honest set of supporters should be power law. Particular examples that raise suspicion when a page receives its PageRank only from very low ranked pages (and then from a very large number of them); such a page has little quality support that makes the fairness of the large number of low-quality supporters questionable. Another example is a set of supporters, all with PageRank values falling into a narrow interval. In this case the large number of similar objects raise the suspicion that they appear by certain means of a cooperation.

The two key observations in detecting link farms, colluding pages or other means of PageRank boosting in the neighborhood of a page are the following:

- Portions of the Web are self-similar; an honest set of supporter pages arise by independent actions of individuals and organizations that build a structure with properties similar to the entire Web. In particular, the PageRank of the supporters follows a power law distribution just as the case for the entire Web.
- Link spammers have a limited budget; when boosting the PageRank of a target page, "unimportant" structures are not replicated.

A perfect form of a link spam is certainly a full replica of the entire Web that automatic link based methods are unable to distinguish from the original, honest copy. Our method hence targets at finding the missing statistical features of dishonest page sets. In our experiment the power law distribution acts as this feature; we remark that (i) other features may perform well and (ii) our algorithm can be fooled by targeting a link farm towards the particular statistical property we employ, hence in a practical application a large number of features should be combined that should probably include non-link based methods as well.

Our algorithm to define SpamRank, a value that measures the amount of undeserved PageRank score of a Web page, consist of three main steps; the overall structure is given in Algorithm 1. First we select the supporters of each given page by a Monte Carlo personalized PageRank simulation as described in Section 2. Then in the second phase we penalize pages that originate a suspicious PageRank share, i.e. their

[1] Approximate personalized PageRank values are stored space-efficiently in a sparse matrix

personalized PageRank is distributed with bias towards suspicious targets. This step is performed target by target; we measure the similarity of the PageRank histogram of sources to an ideal power law model suggested by [5, 27]. Then in the third step we simply personalize PageRank on the penalties. This last step concentrates suspicious activities to their targets; in another way to state, we determine SpamRank by a back-and-forth iteration between targets and sources.

1.2 Related work

With the advent of search engines web spamming appeared as early as 1996 [12, 2]. The first generation of search engines relied mostly on the classic vector space model of information retrieval. Thus web spam pioneers manipulated the content of web pages by stuffing it with keywords repeated several times.

Following Google's success all major search engines quickly incorporated link analysis algorithms such as HITS [26] and PageRank [32] into their ranking schemes. The birth of the highly successful PageRank algorithm [32] was indeed partially motivated by the easy spammability of the simple in-degree count. However Bianchini et al. [7] proved that for any link farm and any target set of pages such that each target page is pointed to by at least one link farm page the sum of PageRank over the target set's nodes is at least large as a linear function of the number of pages in the link farm.

Bharat and Henzinger [6] improved HITS to reduce its sensitivity to mutually reinforcing relationships between hosts. Generally, [31, 8, 35] discuss the (negative) effects of dense subgraphs (known as tightly-knit communities, TKCs) on HITS and other related algorithms.

Section 7 of [29] and the references therein give an overview of the theoretical results underlying the TKC effect that indicate a very weak TKC-type spam resistance of HITS and a somewhat better but still unsatisfying one of PageRank. The results show that HITS is unstable, its hub and authority ranking values can change by an arbitrary large amount if the input graph is perturbed. On the other hand, PageRank values are stable, but the ranking order induced by them is still unstable.

Davison [13] applied a decision tree trained on a broad set of features to distinguish navigational and link-spam (dubbed as nepotistic) links from the good ones. To the best of our knowledge Davison's work is the first openly published research paper *explicitly* devoted to the identification of link spam. More recently Amitay et al. [3] extracted features based on the linkage patterns of web sites. Clustering of the feature space produced a decent amount clusters whose members appeared to belong to the same spam ring.

Recently Fetterly et al. [18] demonstrated that a sizable portion of machine generated spam pages can be identified through statistical analysis. Outliers in the distribution of various web page properties – including host names and IP addresses, in- and out-degrees, page content and rate of change – are shown to be mostly caused by web spam.

Eiron et al. [17] gives evidence that HostRank – PageRank calculated over the host graph – is more resilient against link spam. HostRank's top list contained far fewer questionable URLs than PageRank's because of the relatively reduced weight given to link farm sites. This finding is in good agreement with the before mentioned linear spammability of PageRank [7].

Gyöngyi et al. [23] show that spam sites can be further pushed down in HostRank ordering if we personalize HostRank on a few trusted hub sites. Their method is semi automatic, the trusted 180 seed pages were carefully hand picked from 1250 good hub pages distilled automatically using Inverse PageRank[2]. From the same authors, [22] gives a detailed taxonomy of current web spamming techniques.

Zhang et al. [39] argue that the PageRank of colluding nodes (i.e. pages within the same dense, cliquish

[2] Although [23] makes no citation on Inverse PageRank, computing PageRank on the transposed web graph for finding hubs has been independently introduced previously in [14, 20].

subgraph) is highly correlated with $1/\epsilon$, where ϵ denotes the teleportation probability. Their method increases the probability of jump from nodes with large correlation coefficients. The resulting Adaptive Epsilon scheme is shown to be resistant against artificial, hand-made manipulations implanted by the authors to a real world web graph crawled by the Stanford WebBase project. Baeza-Yates et al. [4] improve Zhang et al.'s analysis and experimentally study the effect of various collusion topologies.

Wu and Davison [38] identify a seed set of link farm pages based on the observation that the in- and out-neighborhood of link farm pages tend to overlap. Then the seed set of bad pages is iteratively extended to other pages which link to many bad pages; finally the links between bad pages are dropped. Experiments show that a simple weighted indegree scheme on the modified graph yields significantly better precision for top ten page hit lists than the Bharat-Henzinger HITS variant. Moreover link farm sites with not too high original HostRank suffer a drastic loss when HostRank is calculated over the pruned graph.

As for a broader outlook, email spam is thoroughly discussed in the proceedings of the recently set up conference [16]. The sensitivity of e-commerce collaborative filtering algorithms to spam attacks is analyzed empirically in [28]. Dwork et al. [15] present spam resistant algorithms for rank aggregation in meta search engines.

2 Preliminaries

In this section we briefly introduce notation, and recall definitions and basic facts about PageRank. We also describe the Monte Carlo simulation for Personalized PageRank of [21].

Let us consider the web as a graph: let pages form a vertex set and hyperlinks define directed edges between them. Let A denote the stochastic matrix corresponding to random walk on this graph, i.e.

$$A_{ij} = \begin{cases} 1/\text{outdeg}(i) & \text{if page } i \text{ points to } j, \\ 0 & \text{otherwise.} \end{cases}$$

The *PageRank* vector $p = (p_1, \ldots, p_N)$ is defined as the solution of the following equation [9]

$$p_i = (1-\epsilon) \cdot \sum_{j=1}^{N} p_j A_{ji} + \epsilon \cdot r_i \,,$$

where $r = (r_1, \ldots, r_N)$ is the teleportation vector and ϵ is the teleportation probability. If r is uniform, i.e. $r_i = 1/N$ for all i, then p is the PageRank. For non-uniform r the solution p is called personalized PageRank; we denote it by $\text{PPR}(r)$. It is easy to see that $\text{PPR}(r)$ is linear in r [25]; in particular

$$p_i = \frac{1}{N} \sum_{v} \text{PPR}_i(\chi_v) \tag{1}$$

where χ_v is the teleportation vector consisting of all 0 except for node v where $\chi_v(v) = 1$.

By equation (1) we may say that the PageRank of page i arises as the contribution of personalization over certain pages v where $\text{PPR}_i(\chi_v)$ is high. We say that page v *supports* i to the above extent.

As noticed independently by [20, 25], the (personalized) PageRank of a vertex is equal to the probability of a random walk terminating at the given vertex where the length is from a geometric distribution: we terminate in step t with probability $\epsilon \cdot (1-\epsilon)^t$. To justify, notice that PageRank can be rewritten as a power series

$$\text{PPR}(r) = \epsilon \cdot \sum_{t=0}^{\infty} (1-\epsilon)^t r \cdot A^t. \tag{2}$$

The term $r \cdot A^t$ corresponds to a random walk of length t and $\epsilon \cdot (1-\epsilon)^t$ is the probability of termination.

The fact that PageRank can be computed as the probability over certain random walks gives rise to the following algorithm [21] that we use in our procedure. We generate large enough number of random walks starting at vertex j and add up probabilities $\epsilon(1-\epsilon)^t$ for their endpoints i; based on the counts we get $\text{Support}_{j,i}$ as an unbiased estimator of $\text{PPR}(\chi_j)_i$. Experiments in [21] suggest that a thousand simulations suffice in order to distinguish high and low ranked pages. The overall idea is summarized in Algorithm 2; we use the implementation described in [21] that differs from the above simplified description in two key aspects. First, for efficiency random walks are generated edge by edge; in one iteration each random walk is augmented by an edge. For such an iteration the set of random walks is sorted by their endvertex; then the iteration can be performed in a single edge scan of the Web graph. Finally we need a last sort over the set of random walks by endvertex; then for a single endvertex i all vertices are collected that support vertex i.

Algorithm 2: Phase 1 outline for finding supporters by Monte Carlo simulation. Actual implementation uses external sort [21].

for all Web pages i **do**
  for $\ell = 1, \ldots, 1000$ **do**
    $t \leftarrow$ random value from geometric distribution with parameter ϵ
    $j \leftarrow$ endvertex of a random walk of length t starting at i
    $\text{Support}_{j,i} \leftarrow \text{Support}_{j,i} + \epsilon(1-\epsilon)^t$

---

3 Algorithm

We define SpamRank, a measure of undeserved PageRank share of Web page, through a three-phase algorithm (Algorithm 1). The algorithm first identifies candidate sources of undeserved PageRank scores. In Phase 1 we select the supporters of each page by the Monte Carlo simulation of [21]. Then in Phase 2 pages receive penalties based on how many potential targets are affected and how strong is the influence on their PageRank values. Finally in Phase 3 we define *SpamRank* as PageRank personalized on the vector of penalties, in a similar way as, by folklore information, Google's BadRank [1] is computed (on the Web graph with reverse edge direction) personalized on identified spam.

In Phase 1 (Algorithm 2) we compute the approximate personalized PageRank vector of all pages j. We use the Monte Carlo approximation of [21]; this algorithm under practically useful parameter settings computes a set of roughly 1,000 nodes i together with a weight $\text{Support}_{i,j}$. This weight can be interpreted as the probability that a random PageRank walk starting at j will end in i.

Before proceeding with the penalty computation in Phase 2, we invert our data (by an external sort) and for each page i we consider the list of pages j such that i is ranked high when personalized on j; the strength is given by $\text{Support}_{i,j}$ as above. Notice that $\text{Support}_{i,j}$ arises from a Monte Carlo simulation and hence its value is 0 for all j where the actual personalized PageRank value is negligible.

For a fixed page i, penalties are defined by considering the PageRank histogram of all i with $\text{Support}_{i,j} > 0$ for pages that receive enough supporters. Pages with less than n_0 supporters (in our experiment $n_0 = 1000$) are ignored; supporter pages that spread their personalized PageRank to targets with less than n_0 incoming paths are of little spamming power anyway.

In the heart of our algorithm we find the method of identifying irregularities in the PageRank distribution of a page's supporters. Given such a measure $\rho \leq 1$ where $\rho = 1$ means perfect regularity, we proceed by

Algorithm 3: Phase 2 Penalty Calculation for Web pages, two variants.

Initialize vector Penalty by all 0
for all Web pages i with at least n_0 supporters j with nonzero $\text{Support}_{i,j}$ **do**
  $\rho \leftarrow$ regularity of the supporters of i
  if $\rho < \rho_0$ **then**
    for all Web pages j with $\text{Support}_{i,j} > 0$ **do**

$$\text{Penalty}_j \leftarrow \text{Penalty}_j + \begin{cases} (\rho_0 - \rho) & \{\text{Variant I}\} \\ (\rho_0 - \rho) \cdot \text{Support}_{i,j} & \{\text{Variant II}\} \end{cases}$$

    {we use $\rho_0 = 0.85$}
for all Web pages i **do**
  if $\text{Penalty}_i > 1$ **then**
    $\text{Penalty}_i \leftarrow 1$

penalizing all the supporter pages proportional to $(\rho_0 - \rho)$ if the measure is below a threshold ρ_0. In our experiments the variant where penalties are also proportional to the strength of the support, $\text{Support}_{i,j}$, proves slightly more effective. Also we put an upper limit of 1 for the penalties of a single page; penalties are otherwise accumulate for pages that participate in several irregular supporter sets.

3.1 Global properties of the Web graph

The fully automatic detection of irregular sets of supporters forms crucial part of our algorithm, hence we briefly describe the intuition behind our method. Key ingredients are

- Rich get richer evolving models: The in-degree and the PageRank of a broad enough set of pages should follow power law distribution.
- Self-similarity: A large-enough supporter set should behave similar to the entire Web.

We build on widely known models of the evolution of the Web [5, 27] that describe global properties such as the degree distribution or the appearance of communities. These models indicate that the overall hyperlink structure arises by copying links to pages depending on their existing popularity, an assumption agreeing with common sense. For example in the most powerful model [27] pages within similar topics copy their links that result in "rich gets richer" and we see a power law degree distribution.

The distribution of PageRank behaves very similar to that of the indegree as noticed among others in [33]. In all Web crawls considered by experiments PageRank has a power law distribution. Clearly PageRank and in-degree should be related as each page has its ϵ/N teleportation share of PageRank and propagates this value through out-links. Figures about PageRank and in-degree correlation vary; some claim a value close to 0 but typically a moderate value close to 0.5 is reported; as an example, the authors of [30] corrected their initial low correlation value to 0.34 in personal communication.

When looking at individual pages, models could in theory completely lose all their predictive power. In practice however strong self-similarity of various portions of the Web is observed [5] that may indicate that the PageRank in a neighborhood should have the same statistical properties as in the entire Web.

Stating in an opposite way, we argue that the neighborhood of a spam page will look different from an honest one. The neighborhood of a link spam will consist of a large number of artificially generated links. These links likely come from similar objects; the same fine granularity obtained by the rich gets richer principle is harder to be locally replicated.

3.2 Phase 2: Penalty generation

We are ready to fill in the last detail (Algorithm 4) of our SpamRank algorithm. Firstly for a page i we may consider the histogram of either the PageRank of all of its supporter pages j with $\text{Support}_{i,j} > 0$ or the product $\text{Support}_{i,j} \cdot \text{PageRank}_j$ for all pages. While the latter variant appears more sophisticated, in our experiments we find Variant A perform slightly better.

Algorithm 4: Irregularity Calculation for Web page i, two variants.

Create a list of buckets for $k = 0, 1, \ldots$
for all Web pages j with $\text{Support}_{i,j} > 0$ **do**
$r = \begin{cases} \text{PageRank}_j & \{\text{Variant A}\} \\ \text{Support}_{i,j} \cdot \text{PageRank}_j & \{\text{Variant B}\} \end{cases}$
Add j to bucket k with $a \cdot b^{k-1} < r \leq a \cdot b^k$ {we use $a = 0.1$, $b = 0.7$}
for all nonempty buckets $k = 0, 1, \ldots$ **do**
$X_k \leftarrow k, \quad Y_k \leftarrow \log(|\text{bucket}_i|)$
$\rho \leftarrow \text{Pearson-correlation}(X, Y)$

Given the PageRank histogram as above, we use a very simple approach to test its fit to a power law distribution. We split pages into buckets by PageRank; we let bucket boundary values grow exponentially as $a \cdot b^k$. We use $a = 0.1$ and $b = 0.7$; the choice has little effect on the results. If the PageRank values follow a power law distribution such that the ℓ-th page in order has rank proportional to $c \cdot \ell^{-\alpha}$, then the theoretic size of bucket k should be

$$\begin{aligned} \int_{a \cdot b^k}^{a \cdot b^{k+1}} c \cdot \ell^{-\alpha} \, d\ell &= c'(b^{k \cdot (-\alpha-1)} - b^{(k+1) \cdot (-\alpha-1)}) \\ &= c'' b^{k \cdot (-\alpha-1)}. \end{aligned}$$

Hence logarithm of the theoretical count within bucket k is linear in k. In our algorithm we penalize proportional to the Pearson correlation coefficient between the index and the logarithm of the count within the bucket as one possible measure for testing a line fit over the data.

4 Experiments

4.1 Data set

Torsten Suel and Yen-Yu Chen kindly provided us with the web graph and the set of URLs extracted from a 31.2 M page crawl of the `.de` domain. The crawl was carried out by the Polybot crawler [37] in April 2004. This German graph is denser than the usual web graphs, it has 962 M edges, which implies an average out-degree of 30.82.

While all the algorithms mentioned in Section 3 can be implemented both in external memory and in a distributed network of workstations, we applied a trivial graph compression method to speed our experiments up by using internal memory. The compressed graph size is 1.3 GB. Computing the Monte Carlo Personalized PageRank for all the nodes took 17 hours, creating the inverted PPR database required 4 hours and resulted in a 14 GB database. Determining SpamRank's personalization vector took 20 minutes, which was followed by another 20 minutes of PageRank computation by simple power method. All programs were ran on a single Pentium 4 3.0 GHz machine with Linux OS.

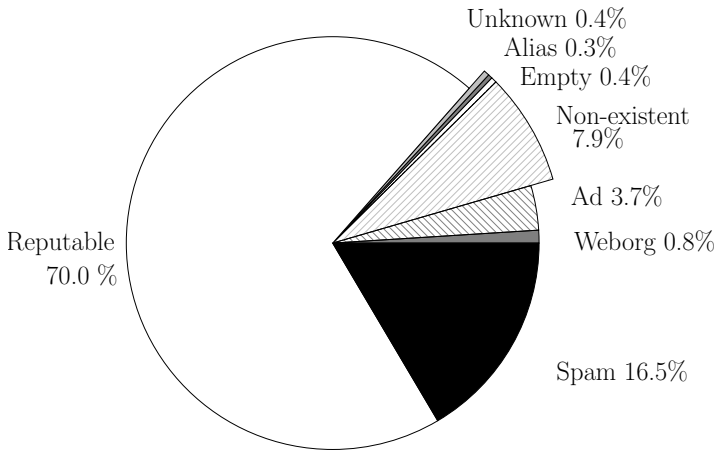

Figure 1: Distribution of categories in the evaluation sample

4.2 Results

To evaluate the signals given by SpamRank we chose a subset of the methods presented in [23]. Firstly we ordered the pages according to their PageRank value and assigned them to 20 consecutive buckets such that each bucket contained 5% of the total PageRank sum with bucket 1 containing the page with the highest PageRank. From each bucket we chose 50 URLs uniformly at random, resulting in a 1000 page sample heavily biased toward pages with high PageRank. Three of the authors received a 400 page subset of the sample, which we manually classified into one of the following categories: reputable, web organization, advertisement, spam, non-existent, empty, alias, and unknown (see [23] for detailed definition of the categories).

We observed a poor pairwise κ value [10] of 0.45 over the 100 pairs of common URLs. The majority of disagreements could be attributed to different rating of pages in affiliate programs and certain cliques. This shows that assessing link spam is nontrivial task for humans as well. By considering all remarks available concerning the judgements over the common set of URLs and using the experience gathered so far, one of the authors revised the classifications for the full 1000 page sample.

We observe relative fast changes among sample pages over time, in particular spam pages changing to non-existent. Judgements after final revision reflect the state as of April 2004.

Figure 1 shows the distribution of categories among the sample. Throwing away the non-existent, empty, unknown and alias pages gave us a usable 910 page sample. Note that the proportion of spam pages in our sample is somewhat higher than in previous studies [23, 18]. We attribute this to the denser web graph and the known notoriousness of spammers over the .de domain [19]. Because of the abundance of pages from the eBay.de domain – 0.8 M pages in the dataset – and due to the misuse of its affiliation program, we decided to treat pages from eBay.de as a separate subcategory in the next two figures.

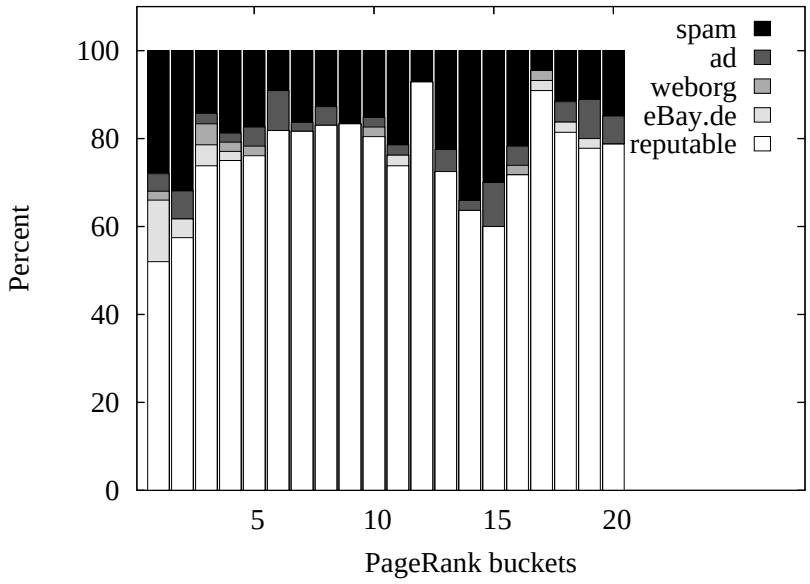

Figure 2: Distribution of categories among PageRank buckets

Figure 2 depicts the distribution of each category conditioned on the PageRank bucket. It can clearly be seen that a large amount of spam pages made it to the top two PageRank buckets where the percentage of spam is even higher than in the full sample.

Using the SpamRank ordering we again assigned each page to one of the 20 SpamRank buckets, the ith SpamRank bucket having exactly the same number of pages in it as the ith PageRank bucket. Figure 3 demonstrates that the first four SpamRank buckets contain a very large amount of spam; these buckets are low in truly reputable non-eBay content.

It is important to note that the ordering induced by SpamRank is very different from PageRank, therefore we cannot assess the properties of SpamRank using traditional precision/recall curves calculated over the original sample as it was drawn according to the PageRank distribution. We refrained from classifying a complete new sample according to the SpamRank distribution, instead we only drew a new random sample of 5×20 pages from the top 5 SpamRank buckets. As it can be seen in Figure 4 the top SpamRank buckets are rich in spam pages, though more than half of the pages in these buckets are not spam. Manual inspection of the non-spam pages revealed that they are predominantly pages from sites with dense, templatic internal structure such as forums, online retail catalogues and structured document archives. In terms of PageRank, the machine generated internal link structure of these sites behaves exactly like a large link farm, therefore we believe it is justified to mark them as pages with artificially inflated PageRank value.

Finally in Figure 5 we plotted the average difference between the PageRank and SpamRank bucket number of pages separately for spam and reputable pages (including eBay.de) in each PageRank bucket. The average demotion in SpamRank compared to PageRank is significantly higher for reputable pages. The (small) positive demotion for spam pages is explained by the fact that the top SpamRank buckets contain a number of fresh, either spammy or cliquish pages (see Figure 4) not included in the original sample.

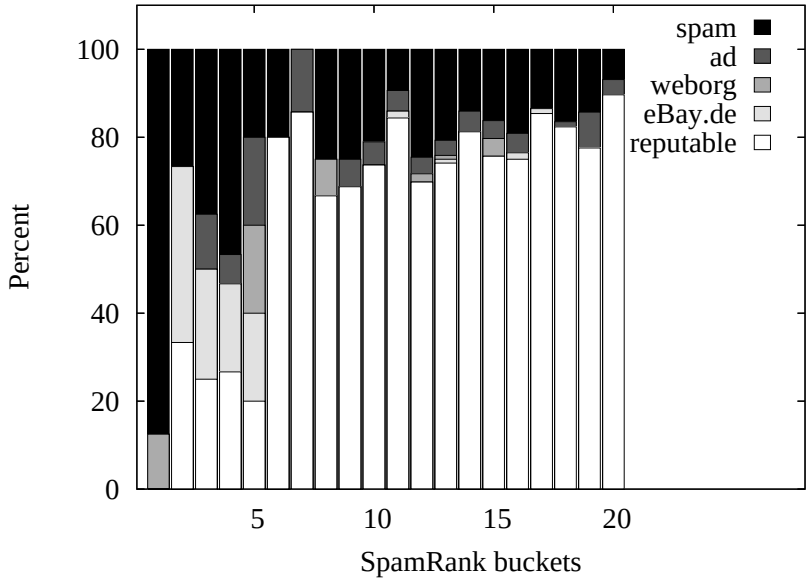

Figure 3: Distribution of categories among SpamRank buckets. Sampling is stratified using PageRank.

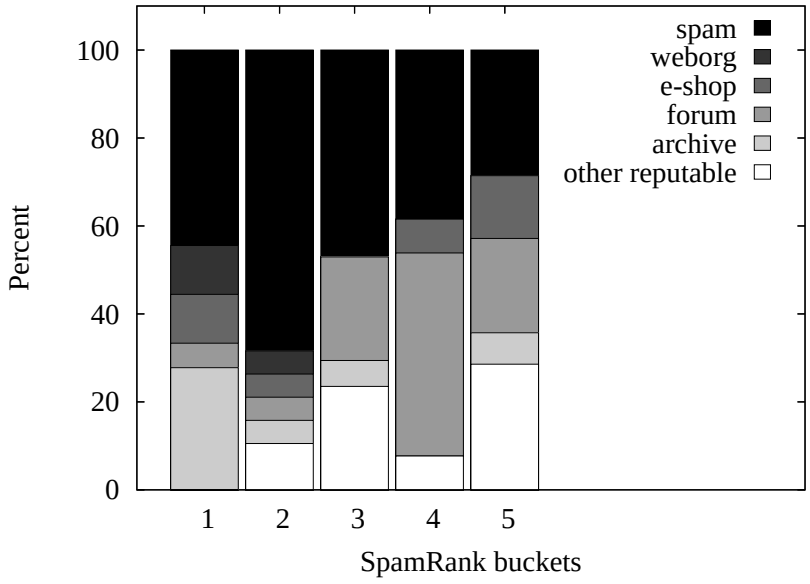

Figure 4: Distribution of categories among the top 5 SpamRank buckets. Sampling is stratified using SpamRank.

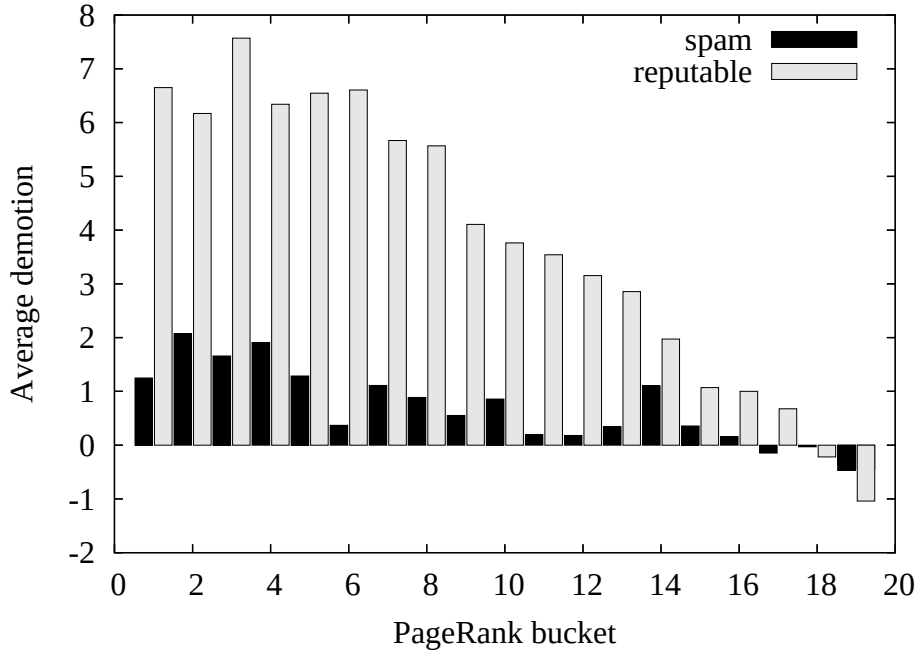

Figure 5: Demotion

5 Conclusions

We presented SpamRank, a three-stage, scalable Monte Carlo algorithm for computing a personalized PageRank vector biased toward link spam pages. Our experiments demonstrated that SpamRank is indeed capable of differentiating among spam and non-spam pages. A number of questions left to subsequent work are as follows. Explore the effects of parameters and assess variants of the algorithm (e.g. penalty dependent on the PageRank of a suspicious target page). Produce a ranking that retains the reputable part of PageRank. Incorporate SpamRank into a ranking function and measure its effect on precision for popular or financially lucrative queries. Lastly compare and evaluate SpamRank against Adaptive Epsilon [39] and Wu and Davison's method [38], the other publicly known PageRank schemes designed to be spam-resistant without human effort.

6 Acknowledgement

The authors would like to thank Torsten Suel and Yen-Yu Chen for providing the `.de` web graph, Alessandro Panconesi and Prabhakar Raghavan for inspiring our research at Bertinoro Web Bar 2004, and lastly Dániel Fogaras and Balázs Rácz for many fruitful discussions.

References

[1] BadRank as the opposite of PageRank. `http://en.pr10.info/pagerank0-badrank/`.

[2] The Word Spy - spamdexing. `http://www.wordspy.com/words/spamdexing.asp`.

[3] E. Amitay, D. Carmel, A. Darlow, R. Lempel, and A. Soffer. The Connectivity Sonar: Detecting site functionality by structural patterns. In *Proceedings of the 14th ACM Conference on Hypertext and Hypermedia (HT)*, Nottingham, United Kingdom, August 26-30 2003.

[4] R. Baeza-Yates, C. Castillo, and V. López. PageRank increase under different collusion topologies. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005.

[5] A.-L. Barabási, R. Albert, and H. Jeong. Scale-free characteristics of random networks: the topology of the word-wide web. *Physica A*, 281:69–77, 2000.

[6] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 104–111, Melbourne, AU, 1998.

[7] M. Bianchini, M. Gori, and F. Scarselli. Inside PageRank. *ACM Transactions on Internet Technology*, 5(1), 2005.

[8] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Finding authorities and hubs from link structures on the world wide web. In *Proceedings of the 10th World Wide Web Conference (WWW)*, pages 415–429, 2001.

[9] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107–117, 1998.

[10] J. Carletta. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

[11] S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. M. Kleinberg. Mining the Web's link structure. *Computer*, 32(8):60–67, 1999.

[12] C. Chekuri, M. H. Goldwasser, P. Raghavan, and E. Upfal. Web search using automatic classification. In *Proceedings of the 6th International World Wide Web Conference (WWW)*, San Jose, US, 1997.

[13] B. D. Davison. Recognizing nepotistic links on the web. In *AAAI-2000 Workshop on Artificial Intelligence for Web Search, Austin, TX*, pages 23–28, July 30 2000.

[14] C. Ding, X. He, P. Husbands, H. Zha, and H. Simon. PageRank, HITS and a unified framework for link analysis. In *Proceedings of the 25th ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 353–354, Tampere, Finland, 2002.

[15] C. Dwork, S. R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International World Wide Web Conference (WWW)*, pages 613–622, Hong Kong, 2001.

[16] D. H. eds. *Proceedings of the 1st Conference on Email and Anti-Spam (CEAS)*. Mountain View, CA, July 2004.

[17] N. Eiron, K. S. McCurley, and J. A. Tomlin. Ranking the web frontier. In *Proceedings of the 13th International World Wide Web Conference (WWW)*, pages 309–318, New York, NY, USA, 2004. ACM Press.

[18] D. Fetterly, M. Manasse, and M. Najork. Spam, damn spam, and statistics – Using statistical analysis to locate spam web pages. In *Proceedings of the 7th International Workshop on the Web and Databases (WebDB)*, Paris, France, 2004.

[19] D. Fetterly, M. Manasse, M. Najork, and J. Wiener. A large-scale study of the evolution of web pages. In *Proceedings of the 12th International World Wide Web Conference (WWW)*, Budapest, Hungary, 2003. ACM Press.

[20] D. Fogaras. Where to start browsing the web? In *Proceedings of the 3rd International Workshop on Innovative Internet Community Systems I2CS, published as LNCS 2877*, pages 65–79, 2003.

[21] D. Fogaras and B. Rácz. Towards scaling fully personalized PageRank. In *Proceedings of the 3rd Workshop on Algorithms and Models for the Web-Graph (WAW)*, pages 105–117, Rome, Italy, October 2004. Full version to appear in Internet Mathematics.

[22] Z. Gyöngyi and H. Garcia-Molina. Web spam taxonomy. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005.

[23] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, Toronto, Canada, 2004.

[24] M. Henziger, R. Motwani, and C. Silverstein. Challenges in web search engines. *SIGIR Forum*, 36(2), 2002.

[25] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12th International World Wide Web Conference (WWW)*, pages 271–279, Budapest, Hungary, 2003. ACM Press.

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

[27] R. Kumar, P. Raghavan, S. Rajagopalan, D. Sivakumar, A. Tomkins, and E. Upfal. Stochastic models for the web graph. In *Proceedings of the 41st IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, 2000.

[28] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *Proceedings of the 13th International World Wide Web Conference (WWW)*, pages 393–402, New York, NY, USA, 2004. ACM Press.

[29] A. N. Langville and C. D. Meyer. Deeper inside PageRank. *Internet Mathematics*, 1(3):335–400, 2004.

[30] L. Laura, S. Leonardi, S. Millozzi, U. Meyer, and Y. F. Sibeyn. Algorithms and experiments for the Web graph. In *Proceedings of the European Symposium on Algorithms*, 2003.

[31] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. In *Proceedings of the 9th World Wide Web Conference (WWW)*, 2000.

[32] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford University, 1998.

[33] G. Pandurangan, P. Raghavan, and E. Upfal. Using PageRank to Characterize Web Structure. In *8th Annual International Computing and Combinatorics Conference (COCOON)*, 2002.

[34] A. Perkins. White paper: The classification of search engine spam, Sept. 2001. Online at http://www.silverdisc.co.uk/articles/spam-classification/.

[35] G. O. Roberts and J. S. Rosenthal. Downweighting tightly knit communities in world wide web rankings. *Advances and Applications in Statistics (ADAS)*, 3:199–216, 2003.

[36] A. Singhal. Challenges in running a commercial search engine. In *IBM Search and Collaboration Seminar 2004*. IBM Haifa Labs, 2004.

[37] T. Suel and V. Shkapenyuk. Design and implementation of a high-performance distributed web crawler. In *Proceedings of the IEEE International Conference on Data Engineering*, February 2002.

[38] B. Wu and B. D. Davison. Identifying link farm pages. In *Proceedings of the 14th International World Wide Web Conference (WWW)*, 2005.

[39] H. Zhang, A. Goel, R. Govindan, K. Mason, and B. V. Roy. Making eigenvector-based reputation systems robust to collusion. In *Proceedings of the 3rd Workshop on Algorithms and Models for the Web-Graph (WAW)*, Rome, Italy, October 2004. Full version to appear in Internet Mathematics.