Link-Based Similarity Search to Fight Web Spam*

András A. Benczúr Károly Csalogány Tamás Sarlós
Informatics Laboratory
Computer and Automation Research Institute
Hungarian Academy of Sciences
11 Lagymanyosi u, H-1111 Budapest
and
Eötvös University, Budapest
{benczur, cskaresz, stamas}@ilab.sztaki.hu
www.ilab.sztaki.hu/websearch

ABSTRACT

We investigate the usability of similarity search in fighting Web spam based on the assumption that an unknown spam page is more similar to certain known spam pages than to honest pages.

In order to be successful, search engine spam never appears in isolation: we observe link farms and alliances for the sole purpose of search engine ranking manipulation. The artificial nature and strong inside connectedness however gave rise to successful algorithms to identify search engine spam. One example is trust and distrust propagation, an idea originating in recommender systems and P2P networks, that yields spam classificators by spreading information along hyperlinks from white and blacklists. While most previous results use PageRank variants for propagation, we form classifiers by investigating similarity top lists of an unknown page along various measures such as co-citation, companion, nearest neighbors in low dimensional projections and SimRank. We test our method over two data sets previously used to measure spam filtering algorithms.

1. INTRODUCTION

With the advent of search engines web spamming appeared as early as 1996 [7]. Identifying and preventing spam was cited as one of the top challenges in web search engines in a 2002 paper [16]. The birth of the highly successful PageRank algorithm [29] was indeed partially motivated by the easy spammability of the simple in-degree count; its variants [19; 11; 15; 3; 39; 2, and many others] proved successful in fighting search engine spam.

Spam and various means of search engine optimization seriously deteriorate search engine ranking results; as a response, building black and whitelist belongs to the daily routine of search engine operation. Our goal is to extend this invaluable source of human annotation either to automatically demote pages similar to certain known spam pages or to suggest additional pages for the operator to be included in the blacklist.

Recently several results has appeared that apply rank propagation to extend initial trust or distrust judgements over a small set of seed pages or sites to the entire web. These methods are either based on propagating trust forward or distrust backwards along the hyperlinks based on the idea that honest pages predominantly point to honest ones, or, stated the other way, spam pages are backlinked only by spam pages. We argue that compared to unidirectional propagation methods, the initial labels are better utilized if we apply similarity search techniques, which involve a bidirectional backward and forward step.

In this paper we concentrate on spreading trust and distrust information from a seed set with the help of hyperlink based similarity measures. Our main goal is to identify features based on similarities to known honest and spam pages that can be used to classify unknown pages. We demonstrate the usability of co-citation, Companion [8], SimRank [18] and variants [10] as well as the singular value decomposition of the adjacency matrix in supervised spam learning.

Hyperlink based similarity to spam versus honest pages is comparable to trust and distrust propagation while giving a natural combination of backward and forward propagation. Given a link farm alliance [13] with one known target labeled as spam, similarity based features will automatically label other targets as spam as well.

As the main result of our paper, we show that over our data set of the `.de` domain as well as the `.ch` domain data in courtesy of the `search.ch` engine [33] similarity based single features perform better than trust or distrust propagation based single features at higher recall values. Ironically, the easiest-to-manipulate co-citation performs best; as an alternate somewhat more robust against manipulations but performing similarly well we suggest Companion [8]. Our results are complementary to the recent results of [2] based on link structure and of [27] based on content analysis. We leave classification based on the combination of features as future work.

2. RELATED RESULTS

Next we survey related results both for hyperlink based spam detection and similarity search. Recently very large number of results appeared to fight spam; we list just the most relevant ones and point to references therein.

2.1 PageRank based trust and distrust propagation

When using trust and distrust information, we may propagate trust forward to pages pointed by trusted ones or distrust backward to pages that point to spam. In previous results we see all variants: TrustRank [15, 39] propagates trust forward, BadRank [31,

*Support from the NKFP 2005 project MOLINGV and by the Inter-University Center for Telecommunications and Informatics.

AIRWEB'06, August 10, 2006, Seattle, Washington, USA.

9] distrust backward; [38] uses a combination. We describe these important predecessors of our work next.

As the first trust propagation method against link spam, Gyöngyi et al. [15] show that spam sites can be pushed down in PageRank ordering if we personalize 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. Notice that TrustRank requires a very carefully selected seed set that we cannot provide in our experiment. Wu et al. [39] describes an improvement of TrustRank by reducing the bias induced by the seed set. Gyöngyi et al. [12] recognize link spam by comparing the TrustRank and PageRank values.

Trust and distrust propagation in trust networks originates in Guha et al. [11] for trust networks; Wu et al. [38] show its applicability for Web spam classification. As noticed by [11] distrust propagation is more problematic that that of trust. Although for a different data type (trust/distrust among Epinions reviewers), they raise the question of interpreting the distrust of a distrusted party. While [38] emphasizes the difference between identifying preferences of a single user and a global notion of trust over the Web, they also require a combination of trust and distrust propagation to achieve best results.

As an earlier result, [19] EigenTrust is PageRank with weights that are trust values. Another method [25] penalizes the biconnected component of a spam page in a subgraph obtained by backward distrust propagation.

2.2 Similarity search, HITS and spam

Several link-based algorithms were designed to evaluate node-to-node similarities in networks; we refer to [23] for an exhaustive list of the available methods ranging from co-citation to more complex measures such as max-flow/min-cut-based similarities of [24] in the vicinity graph of the query. Closest to our notions of link based similarity is co-citation already used in [11] as an elementary step of trust propagation.

Dean and Henzinger [8] describe the Companion algorithm that is reported to outperform co-citation in finding related pages. Their algorithm computes the authority scores by the HITS algorithm [20] in the vicinity of the query page.

HITS itself is known to be vulnerable to spam and in particular to the so-called tightly knit community (TKC) effect. Vulnerability to spam, however, makes HITS a good candidate to actually detect spam when run in the neighborhood of known spam pages that we explore in our paper. An overview of the theoretical results underlying the TKC effect is given in Section 7 of [22] and the references therein that indicate a very weak TKC-type spam resistance of HITS and a somewhat better but still unsatisfying one of PageRank.

Another example of HITS and spam is the result of Wu and Davison [37]. Unlike our approach of exploiting the spam sensibility of HITS in prediction, they make HITS spam resistant by identifying 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 in-degree scheme on the modified graph yields significantly better precision for top ten page hit lists than the Bharat-Henzinger [5] HITS variant.

Additionally we mention the first example that gives anecdotal evidence for the usability of similarities in hyperlink structure to identify spam. Amitay et al. [1] extracted features based on the linkage patterns of web sites and trained a decision tree and a Bayesian classifier to classify each site to one of the 8 predefined functional categories. A cosine metric based clustering of the feature space produced a decent amount clusters whose members appeared to belong to the same spam ring. As it was not the original goal of their research, no results were published on classifying sites as spam or non-spam.

Finally we remark that identifying spam pages is somewhat analogous to classifying web documents into multiple topics. Several results [32, and the references therein] demonstrate that classification accuracy can be significantly increased by taking into account the class labels assigned to neighboring nodes. In accordance with our experiments, Qi and Davison [32] found that most of the improvement comes from the neighborhood defined by co-citation.

2.3 Spam data sets and methodology

Before describing our measurements, we elaborate on the hardness of comparing results of different authors and data sets. We show preliminary results indicating the difficulty of correctly labeling spam by human evaluators as well as compare the different availability of data sets.

While we believe that identifying email spam and certain types of web content spam by human inspection is relative easy and automated methods cannot, in any case, perform as good as human judgement. Search engine spam, however, is much harder to identify. Györgyi and Garcia-Molina [14] 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. A few examples of the `.de` domain are given in our previous result [3].

In contrast to the hardness of manual spam classification, apart from our previous result [3] we have no knowledge of investigations for the reliability of the manual labels. In our experiment [3] over the `.de` domain we report a very poor pairwise $\kappa = 0.45$ [6] over the 100 pairs of URLs with judgements by two different evaluators. 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. Gyöngyi et al. [15] mention "using an author as an evaluator raises the issue of bias in the results" and emphasize the expertise needed for search engine operators that, in our work, have no access. They also describe the hardness of the task as "manual evaluations took weeks: checking a site involves looking at many of its pages and also the linked sites to determine if there is an intention to deceive search engines."

Results on Web spam are in general based on data that needs careful analysis to replicate and compare. The `.uk` crawl [2] that we plan to use in future work is not yet publicly available. Some of the data such as the MSN crawl [27] is proprietary. Gyöngyi et al. [15] use an AltaVista crawl together with a proprietary tool for contracting pages within the same site to a single node prior to PageRank computation that we only mimic over the `.de` domain. While the Stanford WebBase [39] contains pages that are outdated, manual classification is possible with care through the Wayback Machine [39]. This is also true for our 2004 `.de` crawl [36] even though we use a 2005 manual classification [3].

Various top-level or otherwise selected domains may have different spamming behavior; Ntoulas et al. [27] give an invaluable comparison that show major differences among national domains and languages of the page. For the `.de` domain their findings agree with our 16.5% [3] while for the `.uk` domain together with Becchetti et al. [2] they report approximately 6%; the latter measurement also reports 16% of sites as spam over `.uk`.

We also mention the importance of giving more weight to pages

of high rank. Similar to our method, [15] uses a stratified random sample based on PageRank buckets for evaluation. Notice that a uniform sample would consist of mostly very low rank pages that would give little information about top ranked pages most important in search engine applications.

3. THE SIMILARITY BASED SPAM DETECTION ALGORITHMS

In our experiments we use the four similarity measures co-citation, SimRank [18], Companion [8] and singular vectors and we suggest the applicability of further SimRank variants. In this section we briefly introduce notation and the efficient algorithms [10, 35] that we use. We give special importance to algorithms with modest hardware requirements; our experiments ran on a commodity PC.

Similarity based spam prediction is less straightforward than trust and distrust propagation that directly ranks a page as honest or spam. Before describing the algorithms we hence describe our evaluation method. For a given unknown host u, our algorithm computes the similarity top list of u and makes a prediction based on the known spam and honest hosts in this list. For each similarity measure we extract four different features from the size k similarity top list of u. Let the top list contain h honest and s spam pages of the evaluation sample; in general $h + s < k$. Let the sum of the similarities of these pages be s^* and h^*, respectively. We define our features as follows.

- Spam Ratio (SR): fraction of the number of spam within labeled spam and honest pages, $s/(s+h)$.
- Spam over Non-spam (SON): number of spam divided by number of honest pages in the top list, s/h.
- Spam Value Ratio (SVR): sum of the similarity values of spam pages divided by the total similarity value of labeled spam and honest pages under the appropriate similarity function, $s^*/(s^*+h^*)$.
- Spam Value over Non-spam Value (SVONV): similarity value sum for spam divided by same for honest, s^*/h^*.

Given the above values, we may impose a threshold and predict the unknown input page spam if the measure is above the prescribed threshold. For different thresholds we obtain predictions of different quality; by decreasing its value we increase recall and likely but not necessarily decrease precision. For threshold 0 we predict all pages as spam with recall 1 and precision equal to the spam fraction in the data.

3.1 SimRank

Let us consider the web as a graph over hosts by contracting all individual pages that share a common fully qualified host name into the same vertex as in [15]. Let there be N vertices and let hyperlinks define directed edges E between them. Given node v we denote its in- and out-degree by $d^+(v)$ and $d^-(v)$, respectively.

The *PageRank* vector $p = (p_1, \ldots, p_N)$ is defined as the solution of the following equation [29]

$$p_u = (1-c) \cdot \sum_{(v,u)\in E} p_v/d^+(v) + c \cdot r_u \,, \qquad (1)$$

where $r = (r_1, \ldots, r_N)$ is the teleportation distribution and c is the teleportation probability with a typical value of $c \approx 0.15$. We get the PageRank if we set all r_i to $1/N$; for general r we get PageRank personalized on r.

Jeh and Widom [18] define SimRank by the following equation very similar to the PageRank power iteration: initially $\mathrm{Sim}^{(0)}(u_1, u_2) = 1$ if $u_1 = u_2$ and 0 otherwise and then

$$\mathrm{Sim}^{(i)}(u_1,u_2) = \begin{cases} (1-c)\cdot \frac{\sum \mathrm{Sim}^{(i-1)}(v_1,v_2)}{d^-(u_1)\cdot d^-(u_2)} & \text{if } u_1 \neq u_2 \\ 1 & \text{if } u_1 = u_2, \end{cases} \qquad (2)$$

where the summation is for all pairs $(v_1, u_1) \in E, (v_2, u_2) \in E$. SimRank is the multi-step generalization of co-citation in the same way as PageRank generalizes in-degree.

Given a Web page we predict spam by co-citation and SimRank based on the similarity top-list over the entire host graph. In the case of co-citation, the list includes all pages that have a common back-link; ranking is based on the number of such pages. We compute co-citation by keeping the host graph in internal memory.

SimRank power iterations as in (2) are infeasible since they require quadratic space; we use the algorithm of [35] instead, with additive error $\epsilon = 0.001$ and 10 iterations. We use all non-zeroes as the top list with size k. Since in internal steps the algorithm rounds down to multiples of the error parameter, the choice of ϵ determines the value of k.

Fogaras and Rácz [10] describe two variants PSimRank and XJaccard by modifying similarity propagation in the above equation (2); they give randomized approximation algorithms and measure PSimRank as better predictor of topical similarity than SimRank. Additionally, the algorithm we use for SimRank can very easily be modified to take self-similarities into account by relaxing the condition $\mathrm{Sim}(u_1, u_2) = 1$ and making a page more similar to itself if similar pages point to it. This modified measure may serve well for penalizing nepotism; we plan to test variants of self-similarity and PSimRank in future work.

3.2 Companion and SVD

The Singular Value Decomposition (SVD) of a rank ρ matrix $A \in \mathbb{R}^{m\times n}$ is given by $A = U\Sigma V^T$ with $U \in \mathbb{R}^{m\times\rho}$, $\Sigma \in \mathbb{R}^{\rho\times\rho}$ and $V \in \mathbb{R}^{n\times\rho}$ where Σ is a positive diagonal matrix with the singular values in the diagonal. By the Eckart-Young theorem the best rank-t approximation of A with respect to both the Frobenius and spectral norms is $A_t = U_t\Sigma_t V_t^T$, where $U_t \in \mathbb{R}^{m\times t}$ and $V_t \in \mathbb{R}^{n\times t}$ contain the first t columns of U and V and the diagonal $\Sigma_t \in \mathbb{R}^{t\times t}$ contains first t entries of Σ.

We use SVD for nearest neighbor search after a low dimensional projection of the adjacency matrix of the host graph. We represent host u by row u of $V_t\Sigma_t$ and measure similarity as the Eucledian distance in this t dimensional space. Besides computational advantages, the low dimensional projection also serves noise reduction in a similar way as Latent Semantic Indexing [30] applies to the word–document matrix. We perform brute force nearest neighbor search in the t dimensional space defined by U_t and consider the first 1000 nearest vertices as top list. Given that we use very low values of t, we could replace brute force search by more elaborate data structures [34] or approximation [17]; in our case however the sample was small enough to use the simplest implementation. In the experiments we use the SVDPACK [4] Lanczos implementation for computing the first 10 singular vectors.

HITS [20] authority scores are the coordinates of the first (right) singular vector of the adjacency matrix of the vicinity subgraph. The idea of using more than just the first singular vector appears in several results. The instability of a single authority (or hub) vector and stability of the t dimensional projection U_t is described by [26].

The Companion algorithm [8] builds the 2-step alternating neighborhood of the given vertex; then performs the HITS authority computation and returns the top authorities. We use a simplified

version that excludes steps such as edge weighting, large degree handling and link order considerations. For a query node v we build the vicinity graph by selecting nodes of length two alternating forward-backward of backward-forward paths starting at v. We randomly truncate large neighborhoods to a maximum of 2000 nodes in the first step and to 10 in the second step, as in [8]. We rank by the authority score and use all nodes of the vicinity graph as the top list. HITS is computed by simple power iteration.

4. EXPERIMENTS

4.1 Data sets

We use two data sets, the 31.2 M page crawl of the `.de` domain provided us by Torsten Suel and Yen-Yu Chen and the 20 M page crawl mostly from the Switzerland domain as courtesy of the `search.ch` engine [33]. We apply the evaluation methodologies of [3] for the `.de` and the data of [37] for the Switzerland domain that we review next.

The crawl carried out by the Polybot crawler [36] in April 2004 gives a German graph denser than the usual web graphs with 962 M edges implying an average out-degree of 30.82. Unlike in our previous result on the same data [3] we use the host graph not just because it speeds up experimentation but also because intra-site links that would give trivial similarity within the same host disappear and host level detection forms a more interesting task. When forming the host graph, we are left with a modest 808 K node and 24 M edge graph.

For the `.de` data we manually evaluated a stratified random sample as proposed first in [15]. 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. As this step was made for the prior experiment, we computed PageRank over the page-level graph instead of the host graph; stratification in the sample selection however has no further effect on our experiments. From each bucket we chose 50 URLs uniformly at random, resulting in a 1000 page sample heavily biased toward pages with high PageRank. The sample was manually classified as described in [3] with judgements reflecting the state as of April 2004. Figure 1 shows[1] the distribution of categories among the *hosts* that slightly differ from the page level distribution [3]. Our prior findings of 16.5% spam among `.de` pages [3] agrees with [27] and our increased 20.9% spam on the host level with the similar findings of [2] for the `.uk` domain.

The `search.ch` data is a 2004 crawl of approximately 20 M pages mostly from the `.ch` domain. We used the domain graph with 300 K nodes and 24 M edges reflecting the connectivity of the two highest levels of the domain hierarchy within this dataset [37].

The 19605 domains appearing in the URL list extracted by Wu et al. [38] from the Switzerland specific ODP [28] topics formed our trusted set. As spam set we used a labeled list of 728 domains provided by `search.ch` [33]. One particular property of this blacklist is that 627 domains share 144 different IP addresses, the remaining 101 could not be resolved in June 2006. Note that the Swiss evaluation sample contains 3.6% spam only.

4.2 Evaluation by cross-validation

We evaluate our methods together with trust and distrust propagation baselines by three-fold cross-validation. We observe very large variance between various random cross-validation splits, a phenomenon that we show for a single feature instance in Fig. 2

[1] Unless stated otherwise all figures in this section refer to the `.de` dataset.

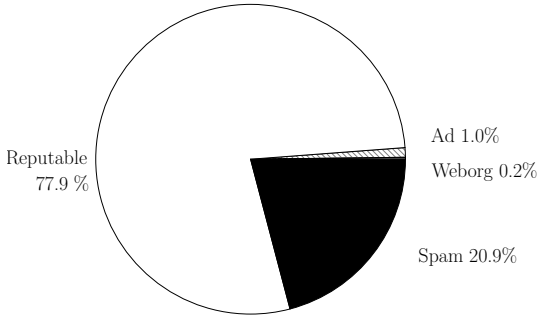

Figure 1: Distribution of categories among the hosts in the evaluation sample

but holds for all features. The explanation likely lies in the small size of our sample: given a query page, the success of spam classification heavily depends on whether those possibly very few pages that contain relevant information are used for test or training.

Given the large variance, we show or measurements by averaging cross-validation results with five independent random splits that altogether correspond to 15 measurements, three for each split. Since in the 15 measurements we will have precision values for different values of recall and we may have several precision values for a given recall, we need special care to average our measurement.

We average our measurements by extrapolating precision for a given recall from measured values. For a single measurement we obtain precision–recall value pairs by imposing different thresholds. By decreasing the threshold we increase recall; increment is however in discrete steps that changes whenever the threshold reaches new hosts. If we reach a single host that is spam, both precision and recall increases by certain amount; we may then linearly extrapolate between the recall at the boundaries. If the new host is honest, we obtain a new, smaller precision value for the previous recall; we average all these values for a single experiment before averaging between measurements. Given ties, we classify more than a single host that makes recall increase and precision change according to the fraction α of spam among the new hosts. For a given intermediate recall we may then interpolate by adding a (possible fractional) number of pages with a fraction α of spam and computing precision. This method reduces to linear interpolation for a single new spam page with $\alpha = 1$ but nonlinear otherwise.

4.3 Baseline results

For baseline experiments we use the trust and distrust propagation measures of Wu et al. [38] by personalizing host based PageRank on known honest vs. spam hosts. We reproduce results of these experiments as Wu et al. [38] choose methods other than precision-recall curves for evaluation. We use the following variants described by [38]. In a single personalized PageRank iteration we may use constant splitting or logarithm splitting instead of equal splitting. We also use the maximum share variant by replacing summation by maximum in the PageRank equation. We leave the maximum parent variant of [38] for future work; we hence test 6 variants, including the original BadRank corresponding to simple

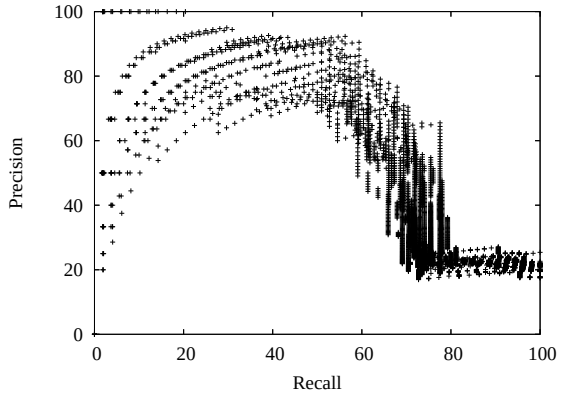

Figure 2: The outcome of five threefold cross-validation results with various random splits, for co-citation with SVR, altogether corresponding to measurement points over 15 precision-recall curves.

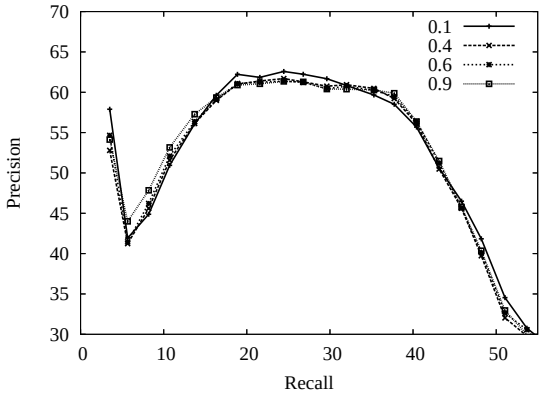

Figure 3: Precision as a function of recall for distrust propagation with logarithm splitting and maximum share. Four different values of teleportation probability c are shown.

summation equal splitting in the terminology of [38].

We experiment with different values of teleportation probability c. This value has negligible effect on the best measures as seen in Fig. 3. Other measures depend more heavily on c and reach best performance in general with low c. Since it has no effect on the comparison of methods we use a uniform $c = 0.1$ afterwards.

Our best trust and distrust propagation measurements are shown in Fig. 4, all with $c = 0.1$. Unmodified BadRank (equal split, summation) performs best at lowest recall but outperformed by equal split maximum share later. Logarithm split maximum share performs slightly worse but still outperforms the remaining three variants. Due to insufficient trust information, trust propagation performs very poor and often even below the random 20.9%, meaning that most spam manages through in cheating our TrustRank. Only the best original TrustRank (equal split, summation) is shown.

As suggested in [38], we improve results by combining trust and distrust propagation. We use linear combinations; surprisingly the best results are achieved by subtracting 0.8 times the trust score from 0.2 times the distrust score. Results for using the previous three best distrust score and the single best TrustRank is shown in Fig. 5. Hence although TrustRank performs bad alone due to insufficient trust information in our `.de` data, still its vote gives significant help to distrust propagation.

Over the `search.ch` dataset unmodified BadRank and logarithm split with simple summation performed best, their graphs are shown in Fig. 11.

4.4 Similarity based features

We use features with abbreviations SR, SON, SVR and SVONV as described in Section 3. For all four Figs. 6–9 we see bad precision at low recall, suggesting that honest pages may also collect high ranked similar spam that may be the result of artificial rank manipulations that is left for future work to verify.

Our methods perform best at relative high recall; finally converges to the random choice of 20.9% spam among hosts for very high recall. We see SR–SVR and SON–SVONV values in pairs performing very close. The first pair performs better at medium recall; the second pair performs poor in general but has a peak at higher recall where outperforms the first pair.

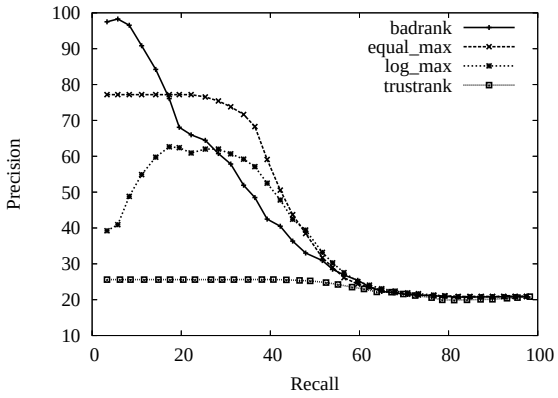

Figure 4: Precision as a function of recall for the best three distrust propagation variants and the single best TrustRank trust propagation.

Co-citation (Fig. 6) turns out best even at relative high recall values with Companion (Fig. 7) as the runner up. Our observations on the relative ordering of individual features and similarity functions hold unchanged over the Swiss dataset as well, hence we report figures only for the German data.

While our most successful candidate, notice the very easy spammability of the co-citation measure. As described by [21] we have to resist both false negative attacks of hiding spam as well as false positive ones that demote the competitor. Co-citation suffers the same vulnerability against spammers as in-degree: a spammer can easily create a large number of honey pot hosts that co-cite quality pages along with the spam target. By adding hosts that point to an honest h and a spam host s, we increase the chance of voting h spam and s honest.

Although SimRank (Fig. 8) performs poorest, it is the most robust measure against manipulations. In order to modify SimRank, the spammer must use a large number of pages that lead to both the spam target s and an honest page h. Depending on the Page-

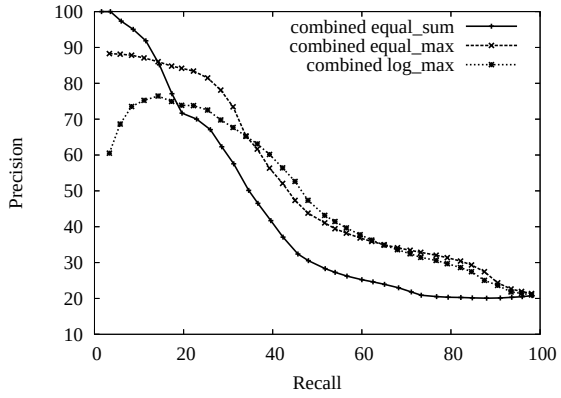

Figure 5: Precision as a function of recall for combined 0.8 times trust and 0.2 times distrust propagation.

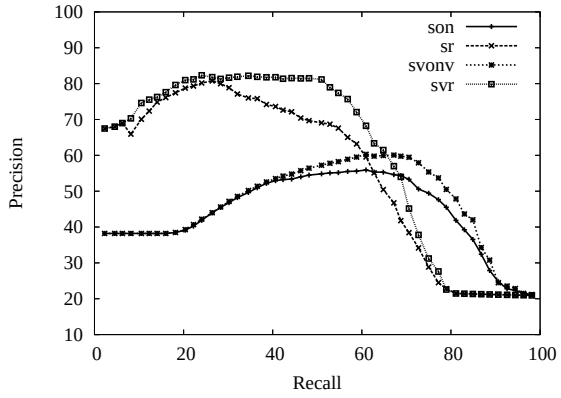

Figure 6: Precision as a function of recall for the four co-citation features.

Rank of h it is very unlikely that paths backward from h meet those from s that would mean high SimRank between h and s. Replacing SimRank with a better performing variant remains future work.

4.5 Comparison of best features

Finally in Fig. 10 we show all features that perform best for certain ranges of recall. BadRank is very effective at penalizing spam only but its recall is very low. Combined 0.2 times distrust minus 0.8 times trust propagation extends BadRank's performance, for the price of slightly decreased precision, to somewhat higher recall. Finally co-citation seems most effective for prediction with high recall. We also show Companion in Fig. 10 as the next best candidate if we disqualify co-citation due to its manipulability.

Turning to the Swiss dataset depicted in Fig. 11 we observe that distrust propagation with the unmodified BadRank algorithm or logarithm split and simple summation performs on par with Companion. As before, co-citation is the most precise measure with the exact feature depending on the level of recall. However overall accuracy is significantly higher than those observed for the `.de` domain. We attribute this to the (undisclosed) method(s) applied

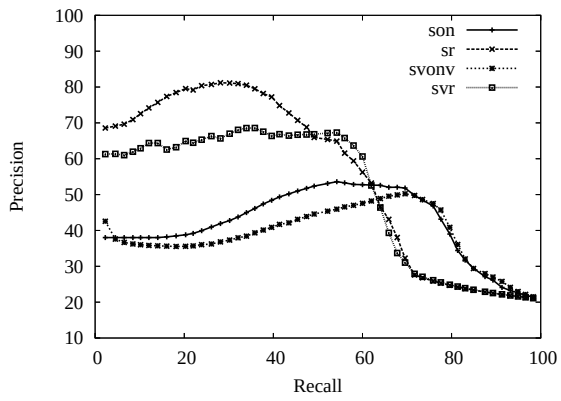

Figure 7: Precision as a function of recall for the four companion features.

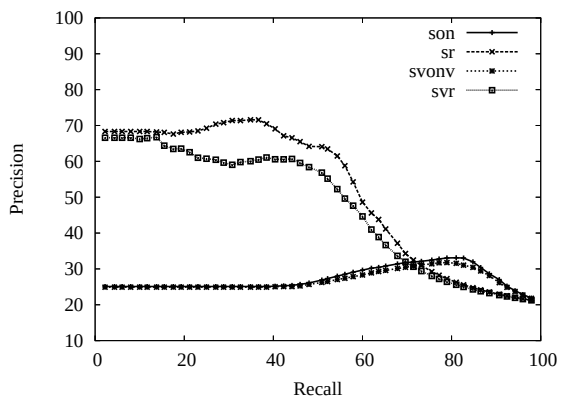

Figure 8: Precision as a function of recall for the four SimRank features.

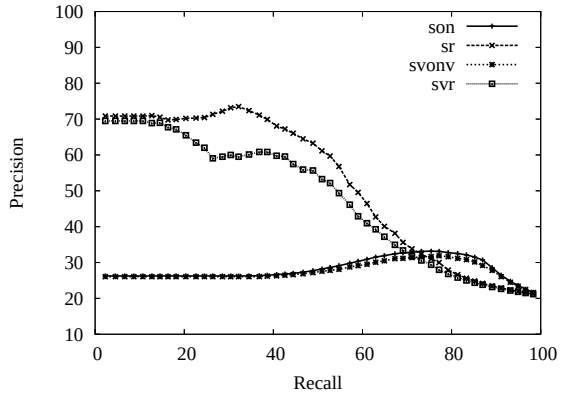

Figure 9: Precision as a function of recall for the four SVD nearest neighbor features.

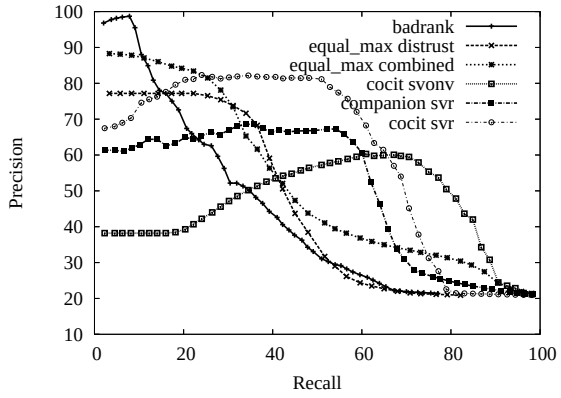

Figure 10: Precision as a function of recall for the best features.

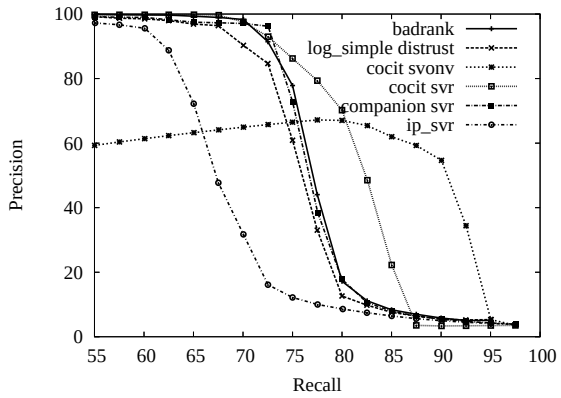

Figure 11: Precision as a function of recall for the best features on the Swiss domain graph.

by `search.ch` [33] to assemble the blacklist. For example, as already noted in Section 4.1, a large number of link farms share the same IP address. Hence a simple similarity measure based on the equality of IP addresses associated with the domains also works reasonably well. As shown on Fig. 12 accuracy decreases if we keep only a single domain for each IP address in the evaluation sample.

5. CONCLUSIONS

We presented hyperlink similarity based single feature classification measurements over a manually classified sample of the `.de` domain and the `search.ch` datasets. Our experiments demonstrated that similarity search based methods are indeed capable of learning the difference between spam and non-spam pages. In further work more SimRank variants and the combination of several features can be measured, including content based statistical features identified by Ntoulas et al. [27]; for combination decision trees as well as SVM should be used. Moreover, akin to [32] it needs to be investigated whether the accuracy of content based spam classifiers can be boosted by incorporating estimates assigned to similar nodes. In addition, the quality of the sample should be

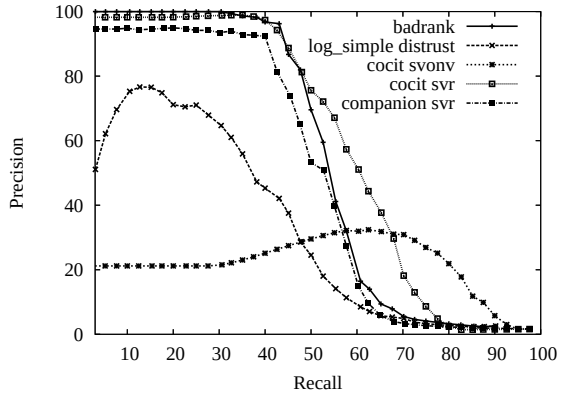

Figure 12: Precision as a function of recall for the best features on the Swiss domain graph with unique IP addresses.

improved by additional manual classification effort as well as other data sets should be involved in the measurement.

6. ACKNOWLEDGEMENT

The authors would like to thank Torsten Suel and Yen-Yu Chen for providing the `.de` web graph and Urban Müller and Baoning Wu and Brian D. Davison for the preprocessed `search.ch` dataset.

7. REFERENCES

[1] 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)*, pages 38–47, Nottingham, United Kingdom, 2003.

[2] L. Becchetti, C. Castillo, D. Donato, S. Leonardi, and R. Baeza-Yates. Link-based characterization and detection of web spam. In *Proceedings of the 2nd International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2006.

[3] A. A. Benczúr, K. Csalogány, T. Sarlós, and M. Uher. SpamRank – Fully automatic link spam detection. In *Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb)*, 2005.

[4] M. W. Berry. SVDPACK: A Fortran-77 software library for the sparse singular value decomposition. Technical report, University of Tennessee, Knoxville, TN, USA, 1992.

[5] 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.

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

[7] 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, USA, 1997.

[8] J. Dean and M. R. Henzinger. Finding related pages in the World Wide Web. In *Proceedings of the 8th World Wide Web Conference (WWW)*, pages 1467–1479, 1999.

[9] I. Drost and T. Scheffer. Thwarting the nigritude ultramarine: learning to identify link spam. In *Proceedings of the 16th European Conference on Machine Learning (ECML)*, volume 3720 of *Lecture Notes in Artificial Intelligence*, pages 233–243, Porto, Portugal, 2005.

[10] D. Fogaras and B. Rácz. Scaling link-based similarity search. In *Proceedings of the 14th World Wide Web Conference (WWW)*, pages 641–650, Chiba, Japan, 2005.

[11] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *Proceedings of the 13th International World Wide Web Conference (WWW)*, pages 403–412, 2004.

[12] Z. Gyöngyi, P. Berkhin, H. Garcia-Molina, and J. Pedersen. Link spam detection based on mass estimation. In *Proceedings of the 32nd International Conference on Very Large Data Bases (VLDB)*, Seoul, Korea, 2006.

[13] Z. Gyöngyi and H. Garcia-Molina. Link spam alliances. In *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB)*, Trondheim, Norway, 2005.

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

[15] 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)*, pages 576–587, Toronto, Canada, 2004.

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

[17] P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th ACM Symposium on Theory of Computing (STOC)*, pages 604–613, 1998.

[18] G. Jeh and J. Widom. SimRank: A measure of structural-context similarity. In *Proceedings of the 8th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 538–543, 2002.

[19] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proceedings of the 12th International World Wide Web Conference (WWW)*, pages 640–651, New York, NY, USA, 2003. ACM Press.

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

[21] 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.

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

[23] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings of the 12th Conference on Information and Knowledge Management (CIKM)*, pages 556–559, 2003.

[24] W. Lu, J. Janssen, E. Milios, and N. Japkowicz. Node similarity in networked information spaces. In *Proceedings of the Conference of the Centre for Advanced Studies on Collaborative research*, page 11, 2001.

[25] P. T. Metaxas and J. Destefano. Web spam, propaganda and trust. In *Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[26] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Link analysis, eigenvectors and stability. In *Proc. Int. Joint Conf. Artificial Intelligence, Seattle, WA*, August 2001.

[27] A. Ntoulas, M. Najork, M. Manasse, and D. Fetterly. Detecting spam web pages through content analysis. In *Proceedings of the 15th International World Wide Web Conference (WWW)*, pages 83–92, Edinburgh, Scotland, 2006.

[28] Open Directory Project (ODP). `http://www.dmoz.org`.

[29] 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.

[30] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proocedings of the ACM Conference on Principles of Database Systems (PODS)*, pages 159–168, 1998.

[31] PR10.info. BadRank as the opposite of PageRank, 2004. `http://en.pr10.info/pagerank0-badrank/` (visited June 27th, 2005).

[32] X. Qi and B. D. Davison. Knowing a web page by the company it keeps. Technical Report LU-CSE-06-011, Lehigh University, 2006.

[33] Räber Information Management GmbH. The Swiss search engine, http://www.search.ch/, 2006.

[34] H. Samet. *The design and analysis of spatial data structures*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1990.

[35] T. Sarlós, A. A. Benczúr, K. Csalogány, D. Fogaras, and B. Rácz. To randomize or not to randomize: Space optimal summaries for hyperlink analysis. In *Proceedings of the 15th World Wide Web Conference (WWW)*, 2006.

[36] T. Suel and V. Shkapenyuk. Design and implementation of a high-performance distributed web crawler. In *Proceedings of the 18th IEEE International Conference on Data Engineering (ICDE)*, pages 357–368, San Jose, California, USA, 2002.

[37] B. Wu and B. D. Davison. Identifying link farm pages. In *Proceedings of the 14th International World Wide Web Conference (WWW)*, pages 820–829, Chiba, Japan, 2005.

[38] B. Wu, V. Goel, and B. D. Davison. Propagating trust and distrust to demote web spam. In *Workshop on Models of Trust for the Web*, Edinburgh, Scotland, 2006.

[39] B. Wu, V. Goel, and B. D. Davison. Topical TrustRank: Using topicality to combat web spam. In *Proceedings of the 15th International World Wide Web Conference (WWW)*, Edinburgh, Scotland, 2006.