Propagating Both Trust and Distrust with Target Differentiation for Combating Web Spam*

Xianchao Zhang
School of Software
Dalian Univ. of Technology
Dalian 116620, China
xczhang@dlut.edu.cn

You Wang
School of Software
Dalian Univ. of Technology
Dalian 116620, China
youwang@mail.dlut.edu.cn

Nan Mou
School of Software
Dalian Univ. of Technology
Dalian 116620, China
nanmou0624@gmail.com

Wenxin Liang†
School of Software
Dalian Univ. of Technology
Dalian 116620, China
wxliang@dlut.edu.cn

Abstract

Propagating trust/distrust from a set of seed (good/bad) pages to the entire Web has been widely used to combat Web spam. It has been mentioned that a combined use of good and bad seeds can lead to better results. However, little work has been known to realize this insight successfully. A serious issue of existing algorithms is that trust/distrust is propagated in non-differential ways. However, it seems to be impossible to implement differential propagation if only trust or distrust is propagated. In this paper, we view that each Web page has both a trustworthy side and an untrustworthy side, and assign two scores to each Web page: T-Rank, scoring the trustworthiness, and D-Rank, scoring the untrustworthiness. We then propose an integrated framework which propagates both trust and distrust. In the framework, the propagation of T-Rank/D-Rank is penalized by the target's current D-Rank/T-Rank. In this way, propagating both trust and distrust with target differentiation is implemented. The proposed Trust-Distrust Rank (TDR) algorithm not only makes full use of both good seeds and bad seeds, but also overcomes the disadvantages of both existing trust propagation and distrust propagation algorithms. Experimental results show that TDR outperforms other typical anti-spam algorithms under various criteria.

Introduction

Web spam pages use various techniques to achieve higher-than-deserved rankings in search engines' results (Gyöngyi, Garcia-Molina, and Pedersen 2004). Spam has been identified as one of the most important challenge faced by search engines (Henzinger, Motwani, and Silverstein 2002). Many techniques for combating Web spam have been proposed so far, and among them link-based semi-automatic techniques that propagate the judgments of human experts from a set of seed pages are the most promising. These techniques can be classified into two categories: trust propagation and distrust propagation. Trust propagation techniques (e.g. TrustRank(Gyöngyi, Garcia-Molina, and Pedersen 2004)) propagate trust from a seed set of good pages recursively through links to the entire Web. Trust propagation techniques are usually used to complete the task of spam demotion (to demote spam pages in the search ranking results). Distrust propagation techniques such as Anti-Trust Rank (Krishnan and Raj 2006) propagate distrust from a seed set of bad (spam) pages recursively through inverse-links to the entire Web. Distrust propagation techniques focus on the task of spam detection (identifying pages with high distrust values as spam pages).

The above two kinds of propagation techniques use either a good seed set or a bad seed set judged by human experts. Both the good seed set and the bad seed set are of small sizes, thus the information encoded by them is invaluable on identifying Web spam. Using only one side of them loses useful information of the other side, which is a regrettable waste. It has been mentioned a combined use of both good and bad seeds can lead to better results (Zhao, Jiang, and Zhang 2008), however, little work has been known to realize this insight successfully. Wu et al. (Wu, Goel, and Davison 2006a) simply made a linear combination of TrustRank score and Anti-Trust score for each page. This kind of linear combination is hard to interpret and it only performs a little better than TrustRank or Anti-Trust Rank on one or two criteria. An anti-spam technique should be evaluated by its synthetical performance on several criteria, thus linear combination achieves limited improvements over TrustRank and Anti-Trust Rank.

The principle of trust propagation algorithms is to propagate trust to trustworthy Web pages through links since hyperlinks are regarded as conveying trust between Web pages. However, existing trust propagation algorithms propagate trust in non-differential ways, i.e., a page propagates its trust uniformly to its neighbors, without considering whether each neighbor should be trusted or distrusted. This kind of blindfold trust propagation is inconsistent with the original intention of trust propagation, thus can not be expected to gain very good effects. This issue is more serious in today's Web graph since there are more and more good-to-spam links with the development of Web 2.0. Non-differential propagation of distrust presents the same problem. However, it seems impossible to implement differential propagation if only trust or distrust is propagated.

The above two kinds of issues call for an integrated framework of propagating both trust and distrust. Trust propaga-

*The work was partially supported by NSFC grants (60873180, 60503003) of China, SRF for ROCS, SEM, and Doctoral Fund of Ministry of Education of China.

†Corresponding author.

tion algorithms assign a trust score and distrust propagation algorithms assign a distrust score to each Web page. The coexistence of this two kinds of algorithms suggests that each Web page has both a trustworthy side and an untrustworthy side. In this paper, we assign two scores to each Web page: T-Rank, scoring the trustworthiness, and D-Rank, scoring the untrustworthiness of the page. We then propose an integrated framework which propagates both trust and distrust. In each iteration of the framework, the propagation of T-Rank/D-Rank is penalized by the target's D-Rank/T-Rank in the previous iteration, thus a page propagates more trust/distust to a trustworthy/untrustworthy neighbor than to an untrustworthy/trustworthy neighbor. In this way, propagating both trust and distrust with target differentiation is implemented. The proposed Trust-Distrust Rank (TDR) algorithm not only makes full use of both good seeds and bad seeds, but also overcomes the disadvantages of both existing trust propagation and distrust propagation algorithms. Experimental results show that TDR outperforms other typical anti-spam algorithms under various criteria.

The remainder of this paper is organized as follows. In Section 2, related work is provided. Section 3 describes some preliminaries and motivation of the algorithm. The TDR algorithm is illustrated in Section 4. Sections 5 provides experimental results. Finally we conclude our discussion in Section 6.

Related Work

Link-based semi-automatic anti-spam algorithms propagate either trust through links from a set of good seed pages or distrust through inverse-links from a set of bad seed pages to the entire Web. Gyöngyi et al. firstly proposed the TrustRank algorithm (Gyöngyi, Garcia-Molina, and Pedersen 2004), which first selects a certain number of seeds by experts' manual evaluation and then propagates trust through links from them. Later improvements over TrustRank add topical information (Wu, Goel, and Davison 2006b) into the algorithm or use variable links (Chen, Yu, and Cheng 2008). The Anti-Trust Rank algorithm (Krishnan and Raj 2006) propagates distrust via inverse-links from a seed set of spam pages to identify spam pages. Wu et al. (Wu and Chellapilla 2007) proposed the Parent Penalty algorithm to identify link farm spam pages by negative value propagation. Some similar distrust propagation algorithms were presented in (Metaxas 2009).

Zhang et al. (Zhang et al. 2009) mentioned that using bidirectional link information is helpful and proposed a HITS-style trust propagation algorithm named CPV. CPV also assigns each Web page two scores, AVRank and HVrank, to measure the page's authority and hubness. However, it only propagates trust, which is different from ours. Wu et al. (Wu, Goel, and Davison 2006a) simply made a linear combination of the TrustRank and Anti-Trust Rank scores for ranking. In this framework both good and bad seeds are used. However, trust and distrust are propagated separately, thus its effect is limited.

Some link based anti-spam algorithms adopt unsupervised methods without seeds. There are also some anti-spam algorithms that involve content information, language models and user' feedback. However, all of these are more time consuming and beyond the scope of this paper since we only talk about link based trust/distrust propagation algorithms, thus they are not listed here due to space limitation.

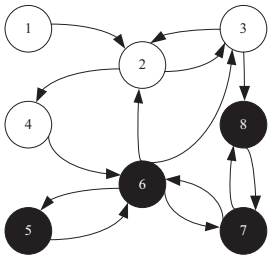

Figure 1: An small Web graph with of good and bad pages

Preliminaries and Motivation

The Web can be modeled as a directed graph. PageRank is the most popular ranking algorithm on the Web graph which simulates user's surfing with a random walk model. TrustRank is a biased version of PageRank, it propagates trust from a human judged seed set recursively to the entire Web. After convergence, good pages are expected to get higher trust scores, while spam pages get lower trust scores. The formula of TrustRank is:

$$\boldsymbol{t} = \alpha \cdot \mathbf{M}^T \cdot \boldsymbol{t} + (1-\alpha) \cdot \boldsymbol{s}, \tag{1}$$

where $\mathbf{M}$ is the line-normalized Web graph matrix; $\boldsymbol{t}$ is the TrustRank score vector; α is a decay factor; $\boldsymbol{s}$ is the normalized trust score vector for the good seed set $\mathcal{S}^+$, i.e. $\boldsymbol{s}(p) = 0$ if $p \notin \mathcal{S}^+$ and $\boldsymbol{s}(p) = 1/|\mathcal{S}^+|$ if $p \in \mathcal{S}^+$.

Anti-Trust Rank, broadly based on the same principle as TrustRank, propagates distrust via inverse-links from a seed set of bad (spam) pages. After convergence, spam pages are expected to get higher distrust scores, while good pages get lower distrust scores. The formula of Anti-Trust Rank is:

$$\boldsymbol{a} = \alpha' \cdot \mathbf{N}^T \cdot \boldsymbol{a} + (1-\alpha') \cdot \boldsymbol{s}', \tag{2}$$

where $\mathbf{N}$ is the line-normalized inverse Web graph matrix; $\boldsymbol{a}$ denotes the Anti-Trust Rank score vector; α' denotes a backward decay factor; $\boldsymbol{s}'$ is the normalized distrust score vector for the bad seed set $\mathcal{S}^-$, i.e. $\boldsymbol{s}'(p) = 0$ if $p \notin \mathcal{S}^-$ and $\boldsymbol{s}'(p) = 1/|\mathcal{S}^-|$ if $p \in \mathcal{S}^-$.

TrustRank, Anti-Trust and their variations have two main issues: (1) only good seeds or bad seeds are used; (2) trust/distrust is propagated without target differentiation. The first issue causes lose of valuable information for better result. The second blindfold trust/distrust propagation issue causes hardness to distinguish bad from good, e.g., a page having one good neighbor and one bad(spam) neighbor propagates half of its trust/distrust score to both of the neighbors, thus the two neighbors have the same trust/distrust scores and no one knows which is bad based on these scores.

These issues often cause dissatisfying results. We use an example to illuminate such results. Fig. 1 represents a small eight-page Web graph where good pages are shown as white, and bad(spam) pages as black. Bad page 5, 6, 7 and 8 make up a link farm (the most popular way of link-based

spam). Using $\{1, 2\}$ as good seeds, setting $\alpha = 0.85$, iterating TrustRank to convergence, we get the following vector of the pages' TrustRank scores:

$$\boldsymbol{t} = [0.08, 0.23, 0.14, 0.10, 0.04, 0.17, 0.13, 0.11].$$

The result is frustrating, since bad pages 6, 7 and 8 get very high TrustRank scores.

For the same example in Fig. 1, using $\{5, 7\}$ as bad seeds, setting $\alpha' = 0.85$, iterating Anti-Trust Rank algorithm until it converges, we get the following Anti-Trust Rank scores:

$$\boldsymbol{a} = [0.03, 0.10, 0.07, 0.08, 0.16, 0.28, 0.19, 0.09].$$

The result is also disappointing, since good page 2 gets higher Anti-Trust Rank score than bad page 8.

The TDR Algorithm

We aim to design an integrated framework that (1) takes advantages from both good and bad seeds, (2) implements target differential propagation such that in each iteration each page propagates less trust and more distrust to a spam neighbor than to a good neighbor. Note that till now there has been two kinds of semi-automatic spam combating algorithms: trust propagation algorithms assign a trustworthy score to each Web page, while distrust propagation algorithms assign an untrustworthy score to each Web page. Thus we can say all Web pages have trustworthy sides and untrustworthy sides but with different extents. This is realistic since each page contains some valuable contents, and at the same time, has a possibility of playing some spam tricks or being manipulated by some spam pages. This phenomenon is just like human beings: an honest man is not expected to always tell the truth, while a dishonest man does not always tell lies. Therefore, we assign each page a *T-Rank* to represent the page's trustworthiness, and a *D-Rank* to represent the page's spamicity or possibility of being manipulated by spam pages.

Intuitively, a page pointed by many reputable pages (with high T-Rank scores but low D-Rank scores) is reputable and should be assigned with a high T-Rank and a low D-Rank. Similarly, a page pointing to many spam pages (with low T-Rank scores but high D-Rank scores) should be assigned with a low T-Rank but a high D-Rank. Both TrustRank and Anti-Trust Rank have partially realized this intuition and the only exception is that these algorithms use only one side of trust or distrust information. However, we need to implement a differential propagation in which (1) a page propagates more trust to a trustworthy neighbor but less trust to an untrustworthy neighbor, (2) a page propagates more distrust to an untrustworthy neighbor but less distrust to an trustworthy neighbor. We design a penalty mechanism to realize differential trust/distrust propagation.

Differential Trust/Distrust Propagation: Like TrustRank and Anti-Trust Rank, the T-Rank/D-Rank score of a page is split equally by the number of the page's outlinks/inlinks, and then is propagated to the page's outlink-neighbors/inlink-neighbors. Unlike TrustRank and Anti-Trust Rank, the propagation of T-Rank/D-Rank is penalized by the receiver's current D-Rank/T-Rank.

There are many ways to implement penalty, our mechanism is designed to be consistent with TrustRank and Anti-Trust Rank such that the successes of previous algorithms are inherited. In our framework, the two scores $\mathbf{t}(p)$ (T-Rank) and $\mathbf{d}(p)$ (D-Rank) of a page p are formalized as follows:

$$\mathbf{t}(p) = \alpha \sum_{q:q\to p} \frac{\beta \mathbf{t}(p)}{\beta \mathbf{t}(p) + (1-\beta)\mathbf{d}(p)} \cdot \frac{\mathbf{t}(q)}{outdegree(q)} + (1-\alpha)\mathbf{s}(p), \quad (3)$$

$$\mathbf{d}(p) = \alpha' \sum_{q:p\to q} \frac{(1-\beta)\mathbf{d}(p)}{(1-\beta)\mathbf{d}(p) + \beta \mathbf{t}(p)} \cdot \frac{\mathbf{d}(q)}{indegree(q)} + (1-\alpha')\mathbf{s}'(p), \quad (4)$$

where $s(p)$ and $s'(p)$ are the same vectors of the good seed set and bad seed set as those of TrustRank and Anti-Trust Rank, respectively; $(1-\beta)\mathbf{d}(p)$ in Eq. (3) and $\beta\mathbf{t}(p)$ in Eq. (4) are used to penalize the propagation of trust and the propagation of distrust to p, respectively; $\beta (0 \le \beta \le 1)$ is the penalty factor which represents the impacts of T-Rank and D-Rank on each other's propagation. We can set a high β value is we want to penalize more on distrust propagation and a low β value want to penalize more on trust propagation. For a page p, if $\mathbf{t}(p) = \mathbf{d}(p) = 0$, we let:

$$\frac{\beta \mathbf{t}(p)}{\beta \mathbf{t}(p) + (1-\beta)\mathbf{d}(p)} = \frac{(1-\beta)\mathbf{d}(p)}{\beta \mathbf{t}(p) + (1-\beta)\mathbf{d}(p)} = 1.$$

We call the proposed algorithm Trust-Distrust Rank (TDR), which is described in Alg. 1.

Algorithm 1: The TDR algorithm

Input: Web graph $\mathcal{G}$; good seeds trust vector $\mathbf{s}$; bad seeds distrust vector $\mathbf{s}'$; decay factor α; penalty factor β

Output: T-Rank scores $\mathbf{t}$; D-Rank scores $\mathbf{d}$

begin
- $\mathbf{t}^0 \leftarrow \mathbf{s}$
- $\mathbf{d}^0 \leftarrow \mathbf{s}'$
- **repeat**
  - Iteratively compute $\mathbf{t}$ and $\mathbf{d}$ according to Formula (3) and (4)
- **until** *Convergence*;
- **return** $\mathbf{t}$,$\mathbf{d}$

Theorem 1 *The TDR algorithm is convergent.*

The proof of the convergence of the algorithm is omitted due to space limitation.

TDR algorithm outputs two kinds of scores: T-Rank and D-Rank. T-Rank scores measure the trustworthiness of the pages and can be used for spam demotion like TrustRank scores. D-Rank scores measure the spammicities of the pages and can be used for spam detection like Anti-Trust Rank scores. Someone may wonder why we do not combine these scores into a single trust score. Firstly, these two scores have different usages. Secondly, a page with high T-Rank score is always with low D-Rank score at the same time and vice verse. Thus it is make no sense to combine these two scores.

The TDR algorithm regresses to the TrustRank algorithm when the penalty factor β is tuned to 1 and regresses to

the Anti-Trust Rank algorithm when β is tuned to 0, thus TDR can be seen as a combinatorial generalization of both TrustRank and Anti-TrusRank. It inherits the advantages of TrustRank and Anti-TrusRank but overcomes the disadvantages of them. We use the same example in Fig. 1 to illuminate the superiority of TDR. By setting $\alpha = \alpha' = 0.85$, $\beta = 0.5$, using $\{1, 2\}$ as good seeds, $\{5, 7\}$ as bad seeds, iterating TDR till convergence, we get the following results:

$$\boldsymbol{t} = [0.166, 0.366, 0.152, 0.148, 0.004, 0.088, 0.021, 0.055]$$

$$\boldsymbol{d} = [0.000, 0.019, 0.028, 0.065, 0.231, 0.297, 0.259, 0.102]$$

The results are excellent: all the good pages' T-Rank scores are higher than those of the bad pages, while all the bad pages' D-Rank scores are higher than those of the good pages. This indicates that TDR can overcome difficulties faced by both TrustRank and Anti-Trust Rank.

Experiments

Dataset, Baseline Algorithm and Parameter Setting

We conducted experiments on WEBSPAM-UK2007 dataset (Yahoo! 2007) and TREC Category B of ClueWeb09 dataset (Callan et al. 2009). The first dataset contains 105,896,555 pages from 114,529 hosts crawled from the .UK domain in May, 2007 and a portion of seeds manually labeled by experts. 3169 good hosts and 134 bad hosts in the strongly connected component of the host graph were used as good seeds and bad seeds, respectively. The second dataset contains 428,136,613 English Web pages. Gordon V. Cormack (Cormack, Smucker, and Clarke 2010) provides the spam labels of this dataset by content based methods with high precision. According to the guideline of the spam scores, we labeled those with percentile-score less than 70 to be spam, and the rest as non-spam. So we had 86,823,693 spam pages together with 61,323,911 non-spam pages (the remaining pages were not assigned percentile-scores). We randomly selected 5% pages from each class as seeds.

We chose PageRank (Brin and Page 1998), TrustRank (Gyöngyi, Garcia-Molina, and Pedersen 2004), CPV (Zhang et al. 2009), LCRank (linear combination of TrustRank and Anti-Trust Rank) (Wu, Goel, and Davison 2006a) as the baseline algorithms for comparison with TDR (using T-Rank) for the task of spam demotion, and chose Anti-Trust Rank (Krishnan and Raj 2006), Inverse PageRank as the baseline algorithms for comparison with TDR (using D-Rank) for the task of spam detection.

The decay factors in these algorithms were all assigned to 0.85 and the penalty factor β in TDR was assigned to 0.5. As suggested in (Wu, Goel, and Davison 2006a), LCRank $=$ $0.1 \times$ TrustRank $- 0.9 \times$ Anti-Trust Rank.

Effectiveness of TDR for spam demotion

Spam demotion aims to demote the ranking positions of spam pages as much as possible on condition that reputable pages keep their relative ranking (PageRank) positions. Two evaluate the performances of spam demotion algorithms, the set of sites (pages) is split into a number (we use 20 here) of buckets according to PageRank values, then usually three evaluation criteria are used (Gyöngyi, Garcia-Molina, and Pedersen 2004; Zhang et al. 2009).

Spam Sites (Pages) in Each Bucket: A good spam demotion algorithm should put fewer spam sites (pages) in small-indexed buckets and more in large-indexed buckets, i.e., the more spam sites (page) are demoted from small-indexed buckets to large-indexed buckets, the better.

Spam Sites (Pages) in Top-k Buckets: This criterion counts the overall spam sites (pages) from bucket 1 to bucket k. It is like the spam sites (pages) in each bucket criterion but easier to compare among different algorithms. The fewer spam sites (pages) in top-k buckets, the better.

Average Spam Sites (Pages) Demotion Distances: This criterion indicates the average demotion distance (how many buckets) of spam sites in the ranking results. The longer the distance is, the more effective the algorithm is.

Spam Sites (Pages) in Each Bucket The number of spam sites in each bucket on WEBSPAM-UK2007 dataset of the algorithms are shown in Fig. 2. It can be seen that TDR (using T-Rank) puts the fewest spam sites in nearly all buckets from 1 to 13. The only two exceptions are bucket 2 and bucket 6. TDR puts a bit more spam sites than CPV in bucket 2, which is because TDR demotes more spam sites from bucket 1 to bucket 2 than CPV. TDR puts a bit more spam sites than LCRank in bucket 6, which is because TDR demotes more spam sites from buckets 1 to 5 to bucket 6 than LCRank. It can also be seen that TDR puts far more spam sites in bucket 19 and bucket 20 than the other algorithms, this indicates that TDR has the strongest effect of demoting spam sites. Thus overall it can be concluded that TDR performs the best on WEBSPAM-UK2007 dataset in term of *Spam Sites (Pages) in Each Bucket*.

For ClueWeb09 dataset, the number of spam pages in each bucket are shown in Fig. 3. It can be seen that TDR puts the fewest spam pages in nearly all buckets from 1 to 16. The only exception is bucket 6, which is because that TDR demotes more spam pages from buckets 1 to 5 to bucket 6. It can also be seen that TDR puts far more spam sites in buckets from 17 to 20 than the other algorithms. Thus TDR performs the best on ClueWeb09 dataset on *Spam Sites (Pages) in Each Bucket* criterion.

Spam Sites (Pages) in Top-k Buckets The *Spam Sites (Pages) in Top-k Buckets* (k ranges from 1 to 20) of the algorithms on WEBSPAM-UK2007 dataset and ClueWeb09 dataset are shown in Fig. 4 and 5, respectively. The *Spam Sites (Pages) in Top-20 Buckets* contains all the spam sites (pages) in the data sets, thus in the end all the curves reach to the same point. It can be clearly seen that when k ranges from 1 to 19, TDR puts fewer spam sites (pages) than all the other algorithms in the Top-k buckets on both data sets. Thus TDR is the best algorithm under the *Spam Sites (Pages) in Top-k Buckets* criterion, and its superiority is more obvious on large scale data set.

Average Spam sites (Pages) Demotion Distances Fig. 6 shows the average bucket level demotion distances of spam site on WEBSPAM-UK2007, and the results on ClueWeb09 are shown in Fig. 7. Only the first 12 buckets are shown because it is meaningless to discuss the demotion distances of sites (pages) in the last 8 (13 to 20) buckets.

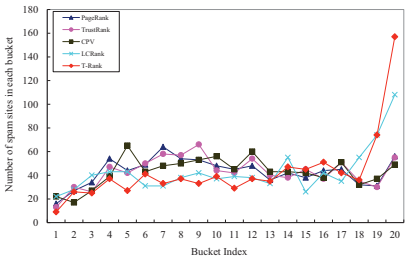

Figure 2: Spam sites in each bucket on WEBSPAM-UK2007

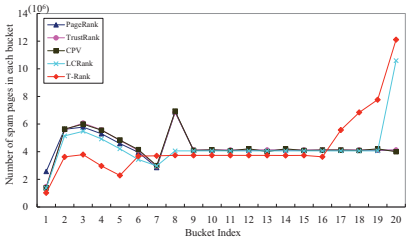

Figure 3: Spam pages in each bucket on ClueWeb09

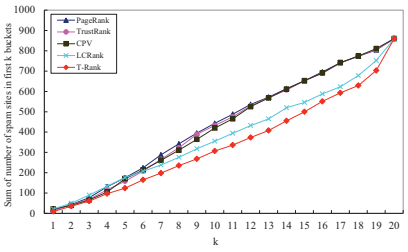

Figure 4: Spam sites in Top-k buckets on WEBSPAM-UK2007

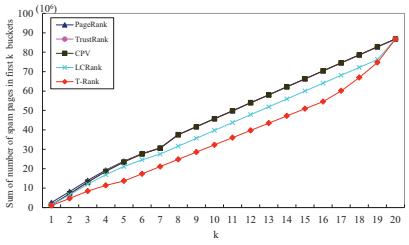

Figure 5: Spam pages in Top-k buckets on ClueWeb09

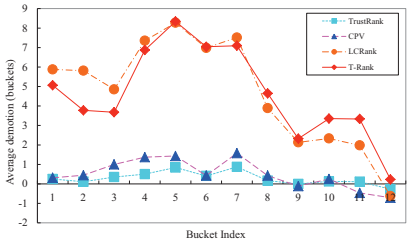

Figure 6: Average demotion distances of spam sites on WEBSPAM-UK2007

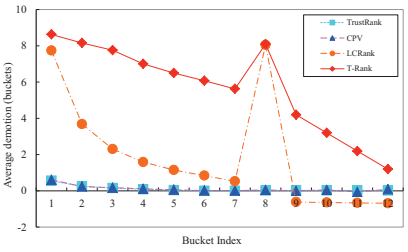

Figure 7: Average demotion distances of spam pages on ClueWeb2009

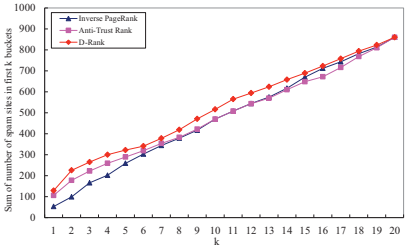

Figure 8: Spam sites in top k buckets of spam detection algorithms on WEBSPAM-UK2007

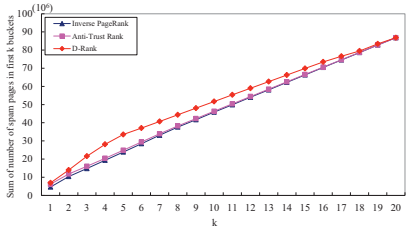

Figure 9: Spam pages in top k buckets of spam detection algorithms on ClueWeb09

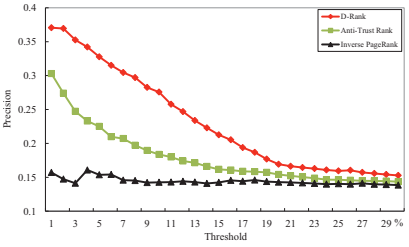

Figure 10: Precisions of different spam detection algorithms on WEBSPAM-UK2007

From Fig. 6, it can be seen that LCRank and TDR evidently outperform TrustRank and CPV on WEBSPAM-UK2007. In 1 to 4 buckets, LCRank demotes bad sites to longer distances than TDR, while in 10 to 12 buckets, TDR demotes bad sites to longer distances than LCRank. In the other buckets, TDR and LCRank alternatively perform better. Overall, the performance of LCRank is a bit better than that of TDR on this criterion. This is because LCRank weights Anti-Trust Rank 8 times higher than TrustRank. Nevertheless, using LCRank, if a good site has a large Trust score and a small Anti-Trust score, it might be mistakenly demoted to a long distance. Only those reputable sites that have nearly zero Anti-Trust scores remain on their original ranking positions, which is a drawback of LCRank since reputable sites (pages) are expected to relatively remain on their original ranking positions.

On ClueWeb09 dataset, TDR demotes spam sites to longer distances than all the other three algorithms. LCRank performs better than CPV and TrustRank in buckets 1 to 8, but it performs worse and even promotes spam pages in buckets 9 to12. As the ratio of spam in ClueWeb09 is extremely higher than that in WEBSPAM-UK2007, we can conclude that TDR performs better than LCRank on demoting spam sites if more spam pages exist. Thus TDR is more suitable for today's Web where spam pages are reported to increase much faster than ever before. Note that in bucket 8 of ClueWeb09, both TDR and LCRank make extraordinarily long demotion distances. It is easy to explain thorough investigation. In bucket 8 of PageRank on this dataset, there are a large amount of forum and blog Web sites. This sites are easily manipulated by spam sites thus they have many links to spam sites, so they have high TrustRank values (relative to the sites in larger index buckets) and high Anti-Trust Rank values at the same time. Therefore, both LCRank and T-Rank scores of such sites will be definitely low. Thus LCRank and T-Rank demote these kind of mixed (good and bad) sites to long distances.

Summarizing the synthetical performances under the above criteria on the two datasets, we can draw a conclusion that TDR (using T-Rank) is the best algorithm for the task of spam demotion.

Effectiveness of TDR for spam detection

Rank-based spam detection algorithms, e.g. Inverse Page Rank, Anti-Trust Rank and TDR (using D-Rank) try to make sites (pages) which are likely to be spam with high rank values such that search engines can take further actions to filter real spam pages (sites) out (usually other features are involved). Thus spam pages (sites) in a good spam detection algorithm should be relatively highly ranked (compared with their PageRank). Spam pages (sites) in top ranked results can be used as one criterion to evaluate this kind of rank-based spam detection algorithms. The more spam pages (sites) are put in the top-k buckets, the more effective the algorithm is. The sum of the number of spam sites in the top-k (k ranges from 1 to 20) buckets on WEBSPAM-UK2007 dataset are shown in Fig. 8 and those on ClueWeb09 dataset are shown in Fig. 9. From these two figures, it can be clearly seen that TDR outperforms the other two algorithms on both datasets.

Rank-based spam detection algorithms are usually used by setting a threshold τ and regarding the top-$\tau\%$ ranked pages (sites) as suspected spam pages (sites). This is actually a classification process, thus we use the precision metric of classification to further investigate the performances of the algorithms. The precisions of the algorithms when the threshold τ ranges from 1 to 30 on known labeled sites of WEBSPAM-UK2007 dataset and ClueWeb09 dataset are shown in Fig. 10 and Fig. 11, respectively. From the two figures, it can be clearly seen that TDR outperforms the other two rank-based spam detection algorithms on both datasets.

Consequently, TDR is the best rank-based spam detection algorithm among the three.

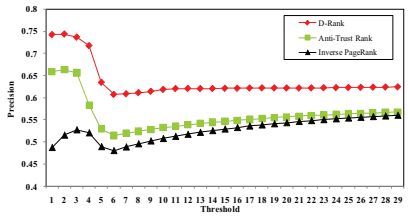

Figure 11: Precisions of different spam detection algorithms on ClueWeb09

Conclusions

In this paper we have proposed a novel anti-spam algorithm TDR which not only takes advantages of both good and bad seeds, but also implements differential trust/distrust propagation. TDR assigns each page a T-Rank score and a D-Rank score and propagates them simultaneously from the seed sets through bidirectional links to the entire Web. The propagation of T-Rank/D-Rank is penalized by the target's current D-Rank/T-Rank, thus an untrustworthy/trustworthy page receives less trust/distrust propagation than a trustworthy/untrustworthy page from the same source page. TDR is a combinatorial generalization of TrustRank and Anti-Trust Rank, but overcomes the disadvantages of both of them. Experimental results show that TDR outperforms previous anti-spam algorithms for both spam demotion and spam detection tasks. Our work can be further improved by incorporating with other improvements of TrustRank and Anti-Trust Rank such as link-variable propagation. Besides, we believe that the incorporation of content features could be more helpful than only utilizing link structure.

Acknowledgments

We are grateful to Liang Wang and Shaoping Zhu for their helpful comments and discussions.

References

Sergey Brin and Lawrence Page 1998. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, 107–117.

Callan, J.; Hoy, M.; Yoo, C.; and Zhao, L. 2009. The clueweb09 data set.

Chen, Q.; Yu, S.-N.; and Cheng, S. 2008. Link variable trustrank for fighting web spam. In *CSSE '08*, 1004–1007.

Cormack, G. V.; Smucker, M. D.; and Clarke, C. L. A. 2010. Efficient and effective spam filtering and re-ranking for large web datasets. *CoRR* abs/1004.5168.

Gyöngyi, Z.; Garcia-Molina, H.; and Pedersen, J. 2004. Combating web spam with trustrank. In *VLDB '04*, 576–587.

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

Krishnan, V., and Raj, R. 2006. Web spam detection with anti-trust rank. In *AIRWeb '06*, 37–40.

Metaxas, P. 2009. Using propagation of distrust to find untrustworthy web neighborhoods. In *ICIW '09*, 516–521.

Wu, B., and Chellapilla, K. 2007. Extracting link spam using biased random walks from spam seed sets. In *AIRWeb '07*, 37–44.

Wu, B.; Goel, V.; and Davison, B. D. 2006a. Propagating trust and distrust to demote web spam. In *MTW '06*.

Wu, B.; Goel, V.; and Davison, B. D. 2006b. Topical trustrank: using topicality to combat web spam. In *WWW '06*, 63–72.

Yahoo! 2007. Yahoo! research: Web spam collections. `http://barcelona.research.yahoo.net/webspam/datasets/` Crawled by the Laboratory of Web Algorithmics, University of Milan, `http://law.dsi.unimi.it/`.

Zhang, Y.; Jiang, Q.; Zhang, L.; and Zhu, Y. 2009. Exploiting bidirectional links: making spamming detection easier. In *CIKM '09*, 1839–1842.

Zhao, L.; Jiang, Q.; and Zhang, Y. 2008. From good to bad ones: Making spam detection easier. In *CITWORKSHOPS '08*, 129–134.