A Survey of Major Techniques for Combating Link Spamming

Yi Li\textsuperscript{a,*}, Jonathan J. H. Zhu\textsuperscript{b}, Xiaoming Li\textsuperscript{c}
\textsuperscript{a}CNDS Lab, Peking University, Beijing 100871, China
\textsuperscript{b}Web Mining Lab, Dept. of Media and Communication, City University of Hong Kong, Hong Kong, China
\textsuperscript{c}CNDS Lab, Peking University, Beijing 100871, China

Abstract

To access exponentially growing information on today’s Web, search engines serve people as the main portal of the Web. Web spamming becomes a major cause of search engine quality deterioration. One of the most dreadful spam techniques is based on manipulating hyperlinks on the Web according to the Web ranking algorithm. This paper presents an overview of present status of link spamming technique, and a summary of several combating techniques. We also present a classification of those techniques.

Keywords: Search Engine; Spamming; Link Spamming

1 Introduction

Today’s World Wide Web (WWW) has evolved into a huge information platform. And search engines have already gained tremendous success. Users rely on search engines to pinpoint information on the Web. And for Web site owners, they rely on search engines to spread information. [16] points out that higher ranking in search engine results is very likely to produce profit to Web site owners, so in order to raise the ranking of their Web sites, some Web site owners start to use certain spamming methods which deceive search engine ranking system.

In this paper, we specifically focus on recent researches aiming at combating link spamming techniques, a most notorious type of spamming. We hereby refer to spamming to actions intended to mislead search engines into ranking some pages higher than they deserve [12], and link spamming refers to spamming methods by misleading the connectivity-based ranking algorithms in search engines[11]. Generally speaking, link spamming techniques mislead search results by deceiving search engine’s ranking strategy, PageRank and HITS are the two most targeted search engine ranking algorithms. This paper gives an overview of such combating techniques, and presents a classification of these methods.

\textsuperscript{*}This research is funded by NSFC Grant 60933004, NSFC Grant 60672171 and HKCERG CityU1456/06H.

\textsuperscript{*}Corresponding author.

Email address: liyi@net.pku.edu.cn (Yi Li).

1548–7741/ Copyright © 2010 Binary Information Press
February 2010
The rest of this paper is organized as follows. Section 2 gives fundamental knowledge for studying anti-spamming techniques. Section 3 introduces the principles and details of several major anti-spamming techniques. Section 4 presents a classification of recent anti-spamming techniques and discusses future works.

2 Preliminaries

Usually we model the Web as a graph $G = (V, E)$, where $V$ denotes the set of all the pages on the Web, $E$ denotes the set of all the hyperlinks on the Web. Here we let $p, q \in V$ denote pages on the Web as default. $(p, q) \in E$ if there exist a hyperlink from $p$ to $q$. $\omega(p)$ denotes out-degree of node $p$, $\iota(p)$ denotes its in-degree. Matrix representations are also important for Web graph. Transition matrix $G$ is the most frequently used one:

$$G(p, q) = \begin{cases} 0 & \text{if } (p, q) \notin E, \\ \frac{1}{\omega(q)} & \text{if } (p, q) \in E. \end{cases}$$

Before fighting back, we need to study our enemy. [12] presents a comprehensive taxonomy of spamming techniques, in which spamming technique are classified. Their study leads to similar taxonomy of countermeasures. They also summarized most used link spamming techniques, and conclude that link spamming is the most popular spamming technique. [11] presents a detailed study of how web pages can be interconnected in order to optimize rankings (according to PageRank) from the point of view of spammers.

The major target algorithm spammers attack: PageRank algorithm is first presented by [15] for an objective ranking of all the pages on the Web by importance. The implication behind PageRank is that a Web page is important if it is pointed by many pages, or by some important pages. Once spammers know this, they either try to generate huge number of trivial pages to point to their target pages, or try to collect links from important pages. Let $r(u)$ and $\mathbf{r}$ denote the PageRank score of page $u$ and the overall PageRank vector respectively, and $d(u)$ and $\mathbf{d}$ denote the random jump probability of page $u$ and the personalized vector, $\alpha$ denotes the damping factor. The original PageRank is this: $r(u) = \alpha \sum_{v|v\rightarrow u} \frac{r(v)}{\omega(v)} + (1 - \alpha)d(u)$, or in the matrix form: $\mathbf{r} = \alpha \mathbf{G} \cdot \mathbf{r} + (1 - \alpha)\mathbf{d}$. In [15], “random surfer model” is used as an intuitive basis for interpreting PageRank algorithm.

PageRank can be interpreted as a Markov process, too: $(\mathbf{I} - \alpha \mathbf{G}) \cdot \mathbf{r} = (1 - \alpha)\mathbf{d}$. In addition, [9] interprets PageRank as a power series (random surfer): $r(u) = \sum_{l=0}^{\infty} \alpha^l (1 - \alpha) \sum_{v} a_l(v, u)d(v)$, with $a_0(v, u) = \begin{cases} 1 & \text{if } v = u \\ 0 & \text{otherwise} \end{cases}$ and $a_{l+1}(v, u) = \sum_{w|w\rightarrow u} a_l(v, w) \frac{\omega(v, w)}{\omega(w)}$, where $a_l(v, u)$ is the probability a walk of length $l$ starting at $v$ stops at $u$.

3 Different Approaches

3.1 TrustRank, Anti-Trust Rank and Spam Mass

A semi-automatic technique, TrustRank, is proposed by [13]. It is developed to separate reputable, good pages from spam. Its main idea relies on the approximate isolation hypothesis: good pages
seldom point to bad pages. It uses a small set of human assessed, well connected “good” pages as seed pages to spread “goodness” scores through links to the whole graph. Thus, it is called semi-automatic. Then “bad” pages would get lower scores than “good” ones. This paper presented schemes of selecting seed sets. After that, TrustRank algorithm is introduced for determining the likelihood that pages are reputable. Algorithm 1 shows the TrustRank algorithm. This is not so different from an algorithm computes the original PageRank. However, seed selection phase become extremely important. The paper points out some criteria on how to select seeds. One of them is to choose pages with high PageRank score, because these pages are the most suspectable ones. And the other one is that seeds should have high connectivity, in other words, they should be able to spread out the “goodness” scores. Thus inverse PageRank is used in the paper to pick out highly connected seeds pages. This work also gives several standard evaluation criteria for anti-spamming techniques for the first time. Pairwise orderliness, means that good pages should be ordered before spam pages; the other one is precision and recall, which assesses how many anti-spamming techniques for the first time. Pairwise orderliness, means that good pages should be able to spread out the “goodness” scores. Thus inverse PageRank is used in the paper to pick out highly connected seeds pages. This work also gives several standard evaluation criteria for anti-spamming techniques.

Algorithm 1: TrustRank

1. \textbf{Input:} $T$, $N$, $L$, $\alpha_B$, $M_B$
2. \textbf{Output:} TrustRank scores, $t^*$
3. $s \leftarrow \text{SelectSeeds}(...)$;
4. $\sigma \leftarrow \text{Rank}(1, \ldots, N, s)$;
5. $d \leftarrow 0_N$;
6. for $i \leftarrow 1 \text{ to } L$ do if $O(\sigma(i)) = 1$ then $d(\sigma(i)) \leftarrow 1$;
7. $t^* \leftarrow d$;
8. for $i = 1 \text{ to } M_B$ do $t^* = \alpha_B \cdot T \cdot t^* + (1 - \alpha_B)d$

Following the previous work, [14] proposed ”Anti-Trust Rank” algorithm. It reverses the intuition of [13]: if a page points to bad pages, then it is very likely that is page is bad, too. Anti-Trust Rank algorithm propagates “badness” scores in the inverse direction of links. Compare to TrustRank, Anti-Trust Rank picks out “bad” pages rather than good pages.

After TrustRank, [10] introduced Spam Mass Estimation. Spam mass is a measure of how much PageRank a page accumulates through being linked by spam pages. It computes and combines two scores, a regular one and a biased one. Complemented their TrustRank [13] in that Spam Mass detects spam as opposed to “detecting” reputable pages. Let $W$ be a walk from $x$ to $y$ in $G = (V, E)$, it is a finite sequence of nodes: $x = x_0, x_1, \ldots, x_k = y, (x_i, x_{i+1}) \in E$. And let $W_{xy}$ be the set of all walks from $x$ to $y$. Then ThePageRank contribution of $x$ to $y$ over the walk $W$ is $q^W_y = \alpha^k \pi(W)(1 - \alpha)v_x$, $\pi(W) = \prod_{i=0}^{k-1} \frac{1}{\pi(x_i)}$. And the total PageRank contribution of $x$ to $y$, $q_y = \sum_{W \in W_{xy}} q^W_y$. The PageRank score of a node $y$ is the sum of the contributions of all other nodes to $y$ is $p_y = \sum_{x \in V} q^x_y$. For a given partitioning $\{V^+, V^−\}$ of $V$ for any node $x$, it is the case that $p_x = q^x_Y + q^x_Y$. Then the absolute spam mass of $x$ is $M_x = q^x_Y$. The relative spam mass of $x$ is $m_x = q^x_Y / p_x$. However $\{V^+, V^−\}$ would be impossible, besides, we would not need the entire algorithm at all if we already have such a partition. So a practical assumption is that we have accurate a priori knowledge of whether some nodes are good $(V^+)$, or spam $(V^-)$. In this paper, they assume only a $V^+$, which is called good core. Given $V^+$, compute 2 sets of PageRank scores as
\[ p = \text{PR}(v), \text{where } v = \left(\frac{1}{n}\right)_n, p' = \text{PR}(v^+), \text{where } v^+ = \begin{cases} \frac{1}{n} & \text{if } x \in \bar{V}^+ \\ 0 & \text{otherwise} \end{cases}. \]

The good core is somewhat similar to the seeds of TrustRank, however in contrast, the good core should be as large as possible, its orders of magnitude should be larger than TrustRank, and should include as many known good nodes as possible. Given \( p_x \) and \( p'_x \), the estimated absolute spam mass of node \( x \) is \( M_x = p_x - p'_x \), and the estimated relative spam mass is \( \tilde{m}_x = \frac{p_x - p'_x}{p_x} = 1 - \frac{p'_x}{p_x} \). An alternative situation is that \( \bar{V}^- \) is provided. Or both \( \bar{V}^+ \) and \( \bar{V}^- \) are provided. Algorithm 2 illustrates the computing of spam mass. You can see Spam Mass detects spam pages, and TrustRank only depresses them.

**Algorithm 2: Spam Mass**

**Data:** good core \( \bar{V}^+, \) relative mass threshold \( \tau \), PageRank threshold \( \rho \)

**Result:** set of spam candidates \( S \)

1. \( S \leftarrow \emptyset \);
2. compute PageRank scores \( p \);
3. construct \( w \) based on \( \bar{V}^+ \) and compute \( p' \);
4. \( \tilde{m} \leftarrow \frac{p - p'}{p} \);
5. for each node \( x \) so that \( p_x \geq \rho \) do if \( m_x \geq \tau \) then \( S \leftarrow S \cup \{x\} \);

### 3.2 Parenterally

[16] presents ParentPenalty. The intuition of this work is that pages within link farms are densely connected with each other and many common pages will exist both in the incoming and the outgoing link sets for a page in a link farm. Their algorithm includes 3 steps. First, generate a seed set from the whole data set, algorithm 3 shows how seed set is chosen.

**Algorithm 3: ParentPenalty: Seed Set**

1. for \( p \) do
2. for \( i \) in \( \text{IN}(p) \) do
3. if \( d(i) \neq d(p) \) and \( d(i) \) not in \( \text{INdomain}(p) \) then add \( d(i) \) to \( \text{INdomain}(i) \);
4. for \( k \) in \( \text{IN}(p) \) do
5. if \( d(k) \neq d(p) \) and \( d(k) \) not in \( \text{OUTdomain}(p) \) then add \( d(k) \) to \( \text{OUTdomain}(i) \);
6. \( \mathcal{X} \leftarrow \text{the intersection of } \text{INdomain}(p) \text{ and } \text{OUTdomain}(p) \);
7. if size(\( \mathcal{X} \)) \( \geq T_{IO} \) then \( A[p] \leftarrow 1 \);

**Algorithm 4: ParentPenalty: Seed Set Expansion**

**Data:** \( A[N], T_{PP} \)

1. while \( A \) do change do
2. for \( p : A[p] = 0 \) do
3. \( \text{badnum} \leftarrow 0 \);
4. for \( k \in \text{OUT}(p) \) do if \( A[k] = 1 \) then \( \text{badnum} \leftarrow \text{badnum} + 1 \);
5. if \( \text{badnum} \geq T_{PP} \) then \( A[p] \leftarrow 1 \);

Here \( \text{IN}(p) \) denotes the incoming links set of page \( p \), \( \text{INdomain}(p) \), \( \text{OUTdomain}(p) \) denote the
domains of the incoming and outgoing links of page \( p \) respectively. \( d(i) \) is the domain name of link \( i \). The expansion step illustrated in algorithm 4 propagates the initial badness value to additional pages. Finally, ranking step combines the badness value together with normal link-based ranking algorithm. On how to ranking using the results that algorithm 3 and 4 give, the paper suggests that, we can use \( A \) to change the transition matrix or to down-weight or delete the pages in \( A \), or even down weight or delete the links among bad pages in \( A \).

### 3.3 Truncated PageRank and Estimation of Supporters

Based on their study of the damping function of PageRank[8], [4] presents Truncated PageRank and Estimation of Supporters algorithms. In this work, they proposes a damping function for rank propagation, and a technique for link spam detection that exploits the distribution of the number of Web page supporters with respect to distance.

Truncated PageRank is based on a particular characteristic: a spam page in a link farm might have a large number of distinct supporters at shorter than it should be distances, and highly-ranked pages have a large number of supporters after a few levels, while lowly-ranked pages do not. Functional ranking proposed in [2] is a link-based ranking algorithm to compute scoring vector \( W \) of the form: \( W = \sum_{t=0}^{\infty} \frac{\text{damping}(t)}{N} G^t \). Truncated PageRank uses a damping function that ignores the contribution of the first few levels of links. Algorithm 5 and 6 shows the Truncated PageRank.

**Algorithm 5: Truncated PageRank: Initialization**

```plaintext
Data: \( N \): number of nodes, \( 0 < \alpha < 1 \): damping factor, \( T \geq -1 \): distance of truncation
1 for \( i : 1 \ldots N \) do
2 \( R[i] \leftarrow \frac{1-\alpha}{\alpha T+1} N \);  
3 if \( T \geq 0 \) then \( \text{Score}[i] \leftarrow 0 \);  
4 else \( \text{Score}[i] \leftarrow R[i] \); // Normal PageRank
```

**Algorithm 6: Truncated PageRank**

```plaintext
1 distance \( \leftarrow 1 \);  
2 while not converged do  
3 \( \text{Aux} \leftarrow 0 \);  
4 for \( src : 1 \ldots N \) do  
5 \( \text{for all link from src to dest do} \) \( \text{Aux}[\text{dest}] \leftarrow \text{Aux}[\text{dest}] + \frac{R[\text{src}]}{\text{outdegree}(\text{src})} \);  
6 for \( i : 1 \ldots N \) do  
7 \( R[i] \leftarrow A[i] \times \alpha \);  
8 if \( distance > T \) then \( \text{Score}[i] \leftarrow \text{Score}[i] + R[i] \);  
9 distance \( \leftarrow distance + 1 \);
```

Estimation of Supporters is based on the fact that spam pages usually have huge number supporters within a few steps while spam-free pages have their supporters distribute after a few levels. However computing the distribution of supporters for all nodes is very expensive. [4] developed a probabilistic algorithm called “Bit Propagation” to estimate supporters. And algorithm 7 shows the overall algorithm. The next question is how to estimate. Let \( S(x, d) \) be
the set of supporters of pages \( x \) at distance \( d \), \( N(x, d) = |S(x, d)| \). In \( \text{INIT}(\text{node}, \text{bit}) \), the \( j \)-th bit of \( v_i \) is set to 1 with probability \( \epsilon \). A base estimation technique is like this, consider a page \( x \):
\[
P[X_i(x) = 1] = 1 - (1 - \epsilon)^{N(x)}.
\]
If \( B_\epsilon(x) = \sum_{i=1}^{k} X_i(x) \), then:
\[
\mathbb{E}[B_\epsilon(x)] = k - k(1 - \epsilon)^{N(x)}.
\]
From the equation above, we can compute \( N(x) \) as:
\[
\bar{N}(x) = \log \frac{1}{1 - \epsilon \left( 1 - \bar{B}_\epsilon(x) \right)^{\frac{1}{k}}}.
\]

**Algorithm 7**: Estimation of Supporters

1. for \( \text{node} : 1 \ldots N \) do
2. for \( \text{bit} : 1 \ldots k \) do
3. \( \text{INIT}(\text{node}, \text{bit}) \);
4. for \( \text{distance} : 1 \ldots d \) do
5. \( \text{Aux} \leftarrow 0_k ; \)
6. for \( \text{src} : 1 \ldots N \) do
7. for all links from \( \text{src} \) to \( \text{dest} \) do
8. \( \text{Aux}[\text{dest}] \leftarrow \text{Aux}[\text{dest}] \text{ OR } V[\text{src}, \cdot] ; \)
9. for \( \text{node} : 1 \ldots N \) do
10. \( V[\text{node}, \cdot] \leftarrow \text{Aux}[\text{node}] ; \)
11. for \( \text{node} : 1 \ldots N \) do
12. \( \text{Supporters}[\text{node}] \leftarrow \text{ESTIMATE}(V[\text{node}, \cdot]) ; \)
13. return Supporters ;

### 3.4 Many Others

[6] introduces Spartan based on the hypothesis that spam pages have a biased distribution of pages that contribute to their undeserved high PageRank values. In the first phase Spartan, supporters of each page is selected using the Monte Carlo simulation. The second phase calculates a penalty score for every page according the abnormal PageRank distribution of its supporters. And finally Personalized PageRank PP is computed for ranking.

[17] studies how to divide the score of parent node to its children. While most works divide its score by the number of its outgoing links and each child gets an equal share, they studied 3 alternatives, for propagating trust there are: equal splitting, constant splitting and logarithm splitting. They also studied three methods for accumulating step: simple summation, maximum share and maximum parent. Apply the same methods to propagating distrust, \( T_D \), they calculate scores for ranking combining trust scores and distrust scores: 
\[
\text{Total}(i) = \eta \times T(i) - \beta \times T_D(i).
\]
At last, they concluded that combining trust and distrust values can demote more spam sites that the sole use of trust scores, and different choices on splitting step and accumulating step can help to demote top ranked spam sites as well as increase the range of trust propagation.

[1] presents Robust PageRank as a refinement of PageRank. This work is based on the idea that decreases the effect of the most influential nodes on the PageRank of a node to make the ranking system more robust. Robust PageRank is reliable against link spam in that the link spammers should invest more in buying new domains to increase the rank of a node in this system. Robust PageRank is computed by setting a maximum PageRank contribution.

Based on their study of damping factor of PageRank, [7] presents TotalRank. Since link farms may use the damping factor \( \alpha \) maliciously, TotalRank avoids the effect of \( \alpha \), and compute ranking scores as:
\[
\text{TotalRank} = \int_{0}^{1} \text{PR}(\alpha) \, d\alpha.
\]
Actually there are some more: [3] reported that local triangle counting can be used as feature in combating link spamming; [5] used link-based similarity to fight link spam. Here we give simple classification of the most important methods from different angles.
4 Summary and Future Works

From some aspect, we can classify these anti-spamming techniques into 3 types. The first type is machine learning techniques that try to recognize different features of spam pages. E.g., [1] developed both supervised and unsupervised learning to recognize link spamming features, like the contributing set of a node; [3] developed triangle counting algorithm to learn features; [6] recognizes abnormal distribution of PageRank of a node’s supporters.

Another type of approaches modifies the original PageRank to be more robust to link spamming techniques. E.g., [4] ignores PageRank contributions of neighbors within short distance; [6] penalize abnormal nodes by decrease their random jumping probabilities; [16] modifies the way PageRank score distribute from parent nodes to child nodes and the way contribution score from parents be aggregated; [1] decreases the contribution of nodes that excess certain value.

Also there is a particular type of anti-spamming techniques that uses the concept of trust or distrust. These approaches require a relatively small set of prior labeled set of pages, and propagate their trust or distrust. [13] propagates trust scores to suppress spam pages; while [14] propagate distrust score to pick out spam pages; [11] further extended TrustRank.

Now let us take a look at those approaches from the PageRank angle. Review the PageRank again: . There are three parameters that can be controlled, one can modify the damping factor or function of PageRank. E.g., TotalRank in [7] integrate on damping factor to avoid the influence of \( \alpha \); Truncated PageRank [4] uses a damping function to ignore nearby neighbors’ contribution. The personal vectors can also be modified. TrustRank[13], Anti-Trust Rank[14], Spam Mass[11] and SpamRank[6] are all in this group. Others modify the transition matrix of PageRank, for example, ParentPenalty[16].

A classification can help us see through many things, refer to figure 1. In this figure, approaches are distributed into blocks on a 2 dimensional space. There are still blank blocks, where we put question marks in, meaning that at least we have not tried every way yet.

Fig. 1: Classification of different approaches
References


