R-SpamRank: A Spam Detection Algorithm Based on Link Analysis

Chenmin Liang[1], Liyun Ru[2], Xiaoyan Zhu[1]

1. *State Key Laboratory of Intelligent Technology and Systems (LITS), Department of Computer Science and Technology Department, Tsinghua University, Beijing, 100084, China*
2. *Sogou.com, Sohu Company*

Abstract

Spam web pages intend to achieve higher-than-deserved ranking by various techniques. While human experts could easily identify spam web pages, the manual evaluating process of a large number of pages is still time consuming and cost consuming. To assist manual evaluation, we propose an algorithm to assign spam values to web pages and semi-automatically select potential spam web pages. We first manually select a small set of spam pages as seeds. Then, based on the link structure of the web, the initial R-SpamRank values assigned to the seed pages propagate through links and distribute among the whole web page set. After sorting the pages according to their R-SpamRank values, the pages with high values are selected. Our experiments and analyses show that the algorithm is highly successful in identifying spam pages, which gains a precision of 99.1% in the top 10,000 web pages with the highest R-SpamRank values.

Keywords: PageRank, spam, link farm

1. Introduction

The World Wide Web (WWW) is hypertext. Besides "flat" texts, it provides considerable auxiliary information on top of the texts of the web pages, such as link structures and link texts. The PageRank algorithm was proposed in 1998 by Lawrence Page and Sergey Brin, and it took the advantage of the link structure of the Web to produce a global "importance" ranking of every web page [1]. The PageRank algorithm has been applied to the Google search engine, which has gained great success over the past years. As one of the most important algorithms in the modern search engines, PageRank algorithm has been widely used in many search engine systems.

As search engines get more popularity among users, most frequently, they are the entryways to the Web. The web pages that rank higher in search results would attract more attention, and hopefully would gain more financial profits. Some people try to raise the rankings of their pages by misleading the search engines. We call this behavior as "search engine spam" [2][3]. Spamming techniques are divided into two categories: content-based spamming and link-based spamming. Early search engine spamming techniques are mostly content-based spamming, which aims at traditional textual information retrieval algorithms such as TF-IDF. With the popularity of link-based ranking algorithm such as PageRank, new spamming techniques, which are link-based, became important. In this paper, we will focus our discussion on link-based spamming.

Identifying and preventing spam was cited as one of the top challenges in web search engines [4]. It leads to decreases in the quality of search results, and increases the number of useless pages in indexes [5]. While human experts could identify spams, it is too costly to manually evaluate a large number of pages. In this paper, we propose a semi-automatically method to identify spam web pages.

In section 2, we review the PageRank algorithm and related algorithms to evaluate good and bad pages; In section 3, the R-SpamRank algorithm is introduced; In section 4, our experiments are presented and further analyzed; In section 5, we conclude the paper and give our future work.

2. Related Work

We will introduce briefly the PageRank algorithm and other link-based rank algorithms in this section.

2.1. PageRank

PageRank could be thought of as a model of user behavior. It assumes that there is a "random surfer". Starting from a randomly given web page, people usually keeps clicking on the forward links, never hitting "back" but eventually get bored and start inputting another random web page. PageRank computes the probability that the random surfer visits a page. The possibility for a web page to be clicked is determined by several factors: 1. the original importance of the webpage, this determines the possibility of a web page to be started; 2. the total number of web pages that link to it, and the importance and the forward link number of each of these web pages. The PageRank algorithm could be presented as follow [6]:

$$r(p) = \alpha \times \sum_{q:(q,p)\in \varepsilon} \frac{r(q)}{\omega(q)} + (1-\alpha) \times \frac{1}{N} \qquad (1)$$

where *r(p)* is the PageRank value for a web page p; *w(q)* is the number of forward links on the page q; *N* is the total number of web pages in the Web; α is the damping factor; *(q,p)*∈ε means that web page q points to web page p.

A page can have a high PageRank if there are many pages that point to it, or if there are some pages with high PageRank pointing to it. This seems very reasonable and practical. However, it is vulnerable to some link-based spamming techniques.

2.2. Other Rank Algorithms

Many algorithms have been proposed to help detect and filter spam pages based on link analysis.

Gyongyi et al. describe an algorithm, TrustRank, to combat web spam [7]. Unlike PageRank that is based on backward links of web pages, TrustRank considers the forward links instead. The TrustRank assumes that a good web page usually points to good web pages, and seldom links to spam web pages. They first select a small set of known good pages as the seed pages. Then they follow an approach similar to PageRank; the trust score is propagated via forward links to other web pages. Finally, the pages with high trust scores are selected as good pages. Their algorithm is not appropriate to the web island pages [8]. In contrast to their algorithm, the following BadRank algorithm and our R-SpamRank algorithm aim to detect spam web pages instead of good web pages.

BadRank is an algorithm used by Google to help detect spam web pages [9]. It uses a principle based on "linking to bad neighborhoods", that is, a page will get high BadRank value if it points to some pages with high BadRank values. While PageRank uses the backward links of a web page, BadRank gathers information on the forward links of a web page, thus BadRank could be regarded as a reversion of PageRank. The formula of BadRank is given as:

$$BR(A) = E(A)(1-d) + d\sum_{i=1}^{n} \frac{BR(Ti)}{C(Ti)} \qquad (2)$$

where *BR(A)* is the BadRank value of page A; T_i is a page that page A points to, with *BR*(T_i) as its BadRank value; *C*(T_i) is the total number of the backward links of page T_i; *d* is a damping factor; *E(A)* is the original BadRank value for page A, which is determined by the spam filter. Since no algorithms of how to calculate *E(A)* and how to combine BadRank values with other ranking methods such as PageRank are given, we cannot tell the effectiveness of this approach [10]. Our algorithm has similar philosophy to BadRank, but we give the *E(A)* value and adjust some parameters of its formula. Besides, as we have a large data set with a large list of black web pages, we could test and improve the algorithm according to our experiment results.

Benczúr et.al. proposed an algorithm, SpamRank, which rank pages with a large amount of low quality backward links as spams. It is a three-stage, scalable Monte Carlo algorithm for computing a personalized PageRank vector biased toward link spam pages. Their method does not have a whitelist, blacklist or other means of human intervention. As no traditional precision/recall results are given in the paper, we could not compare the effectiveness of our algorithm with theirs.

3. R-SpamRank Algorithm

Our algorithm aims to detect spam web pages. In our algorithm, the web page gains the spam rank value through forward links, which are the links of reverse direction used in traditional link-based algorithm. Therefore, we call our algorithm R-SpamRank which means reverse spam rank.

Our algorithm uses a blacklist containing spam web pages as seeds. The blacklist is manually collected in our experimental system. We assigned an initial R-SpamRank value for each page in the blacklist, and these values would expand in the iterative computation to the web pages linking to them. The formula of the algorithm is shown in equation (3).

$$RSR(A) = (1-\lambda)I(A) + \lambda\sum_{i=1}^{n}\frac{RSR(T_i)}{C(T_i)} \qquad (3)$$

$$I(A) = \begin{cases} 1 & if \quad A \ in \ blacklist \\ 0 & otherwise \end{cases} \qquad (4)$$

where *RSR(A)* is the R-SpamRank value of page A; λ is a damping factor, which is usually set to 0.85; I(A) is the initial value for page A, it is set to 1 if page A in the original blacklist, otherwise 0; n is the number of forward links of page A, and T_i is the i_{th} forward link page of page A; C(T_i) is the number of in links of Page T_i; *RSR*(T_i) is the R-SpamRank value of page T_i.

Our algorithm is similar to BadRank, which is discussed in section 2.2. For a web page A, if it is in the initial blacklist, I(A) is set to 1; otherwise it is set to 0. This is a simple and effective way to assign the initial value for a page. In contrast, the sum of all E(A) in BadRank has to equal the total number of web pages. As the number of web pages increases all the time, this does not seem to be applicable. At the same time, as we could not know the implementation detail of E(A) calculating in BadRank, we could not know its efficiency.

We could describe the Web as a graph, with the pages represented by nodes and the links between pages to be edges which connect nodes. An adjacent matrix M could be used to present the graph, i.e., M[i,j] is set to 1 if there is a link from page i to j, otherwise it is set to 0. This adjacent matrix could help to present many link-based algorithms.

The equivalent matrix equation form of equation (3) is:

$$\mathbf{t} = (1-\lambda)\mathbf{d} + \lambda\cdot\mathbf{T}\cdot\mathbf{t}^{*} \qquad (5)$$

where T is the adjacent matrix represented the web structure, in which T(*i,j*) is set to 1/C(T_j) ,where C(T_j)is the number of backward links of Page T_j, otherwise $T_{(i,j)}$ is set to 0; λ is the damping factor with a value of 0.85; vector d is a static score distribution vector of arbitrary, non-negative entries. Vector d can be used to assign a non-zero static score to each page in the initial blacklist; Vector t* is the vector of R-SpamRank values resulted from the last iteration; Vector t is the vector of the R-SpamRank values of the current iteration.

The equation (3) and (5) should be computed iteratively to gain a final stable value for each web page. To illustrate the process, we give a simple example for our algorithm. Consider the link structure of figure 1. Assume that we have an initial R-SpamRank vector as: [0,1,1,0,0,0]. That is, only web page 2 and 3 are given as seed spam web pages at first.

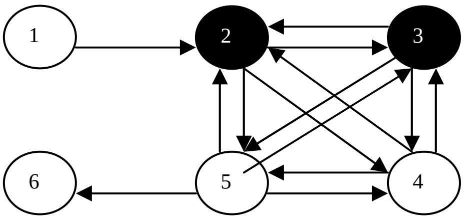

Figure 1: A simple web page graph

○ normal

● spam

The adjacent matrix for the graph in Figure 1 could be represented as:

$$T = \begin{bmatrix} 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{4} & 0 & \frac{1}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{4} & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\ 0 & \frac{1}{4} & \frac{1}{3} & \frac{1}{3} & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The vector d is represented as $\mathbf{d} = [0,1,1,0,0,0]^T$, as page 2 and 3 are in our initial blacklist.

At the beginning of the computing, we assign t=d, and then we use equation (5) to iteratively compute the vector t. After the first iteration, the value of the output vector t is:

$$\mathbf{t} = [0.21, 0.43, 0.36, 0.50, 0.50, 0]^T$$

We could see that the R-SpamRank values in page 2 and page 3 have been propagated to page 4 and page 5. After about 26 iterations, the elements in vector t converge to stable values. The value of the output vector t is:

$$\mathbf{t} = [0.09, 0.42, 0.40, 0.28, 0.28, 0]^T$$

The R-SpamRank values for page 2,3,4, and 5 are high, as they form a link farm [10]. The R-SpamRank value for page 6 is equal to 0, as it does not have any forward link page.

4. Experiments

The web page data set we used contains 5,000,000 web pages, which is offered by the search engine of Sogou.com. After iteratively computing equation (3) for about 50 times, the R-SpamRank value of each web page becomes stable. We sorted all pages according to their R-SpamRank values and selected the top 10,000 web pages with the highest R-SpamRank values.

To gain insights, we carried out manual analyses of the results. According to the level of spamming, we divided the spam pages into two classes in our analyses:

Pure spam pages. These pages use the regular spamming techniques, including term spamming, link spamming, and redirection.

Semi spam pages. These pages have little information that is useful, which is not qualified to its site names.

We assigned the web pages to four categories according to the page types they belong to: web pages that could not be open are assigned the value “0”; web pages that are good are assigned the value “1”; semi spam pages are assigned the value “2”; and the pure spam pages are assigned the value “3”.

4.1 Experiment analysis 1: analysis of selected pages

We first select the first 100 and the last 100 web pages from the top 10,000 web pages with the highest R-SpamRank values. We manually review each of the web pages, and classify it to one of four categories mentioned above. In order to evaluate the distribution of the spam pages, we also record the domain name of each web page. Our analysis results are shown in Table 1 and Table 2. As recorded in Table 1, 99% of the selected web pages are spam pages. This high percentage illustrates that the web pages with high R-SpamRank values have high possibilities of being spam pages. In evaluating the distribution of the spam pages, we found out that the selected web pages mainly belong to two domains. As the pages under the same domain form a link farm, this result demonstrates the capability of our algorithm in detecting link farms.

Table 1. Page Type Results (Number of total analyzed pages = 200)

Page Type	Number of pages	Percentage
Spam pages	198	99%
Good pages	0	0
Could-not open pages	2	1%

(Percentage = Number of pages/Number of total analyzed pages)

Table 2. Domain Distribution (Number of total analyzed pages = 200)

Domain Name	Number of pages	Percentage
.bobook.com	99	49.5%
.infocn.ht200.com	79	39.5%
others	22	11%

(Percentage = Number of pages/Number of total analyzed pages)

4.2 Experiment analysis 2: analysis after domain cleaning

In our first experiment, we detect several link farm domains in the selected web pages. For further study, we delete the web pages under the detected domain link farms (including *.zj.com; *.bobook.com; *.infocn.ht200.com), and do further investigation into remain web pages. The total number of remain web pages is 178. Our analysis results are recorded in Table 3 and Table 4. After domain cleaning, the remained web pages still have a spam percentage as high as 87.1%, which illustrates the capacity of our algorithm in detecting spam web pages. Among the remained web pages, two domains could still be detected as link farms, namely the “.mblogger.cn” and the “.home4u.china.com”.

In the row of “others” in Table 4, 14 good pages exist, which sums up to 34.1% of the total number of the row “others”. Besides, 9 pages could not be opened, which amount to a percentage of 22%. When evaluated strictly with the rules of pure spam web pages (term spamming, link spamming, and redirection), the number of pure spam web pages is 145, which has a percentage of 81.5% of the total analyzed pages. These pure spam web pages are mainly distributed under the domain of “.mblogger.cn” and “.home4u.china.com”.

Table 3. Page Type Result After Domain Cleaning (Number of total analyzed pages = 178)

Page Type	Number of pages	Percentage
Spam pages	155	87.1%
Good pages	14	7.9%
Could-not open pages	9	5%

(Percentage = Number of pages/Number of total analyzed pages)

Table 4. Domain Distribution After Domain Cleaning (Number of total analyzed pages = 178)

Domain Name	Number of pages	Percentage	Number of spam pages	Percentage of spam pages
.mblogger.cn	125	70.2%	125	100%
.home4u.china.com	12	6.7%	12	100%
others	41	23.1%	18	43.9%

(Percentage = Number of spam pages/Number of total analyzed pages)

4.3 Experiment analysis 3: domain distribution of crawled pages

Of the top 10,000 web pages from the R-SpamRank output results, 6867 web pages are crawled by the spider of Sogou.com. The crawled web pages are mainly distributed under several domains, which are later recognized as link farms. The results are listed in Table 5.The 6867 crawled pages come from the top 10,000 pages with the highest R-SpamRank values, in which 5 link farms could be detected. The pages in the link farms sum up to 99.1% of all the crawled pages.

Table 5. Domain Distribution of crawled pages(Number of total crawled pages = 6867)

Domain	Number of web pages	Percentage
.bobook.com	5703	83%
.infocn.ht200.com	835	12.2%
.zj.com	127	1.8%
.mblogger.cn	129	1.9%
.home4u.china.com	12	0.2%
others	61	0.9%

(Percentage = Number of web pages/Number of total crawled pages)

5. Conclusion

In this paper, we discuss our R-SpamRank algorithm and take further analyses into its experimental results. Based on the link structure of the web, our R-SpamRank algorithm aims to detect spam web pages, especially spam web pages in a link farm. To evaluate the capacity of our algorithm, we carried out experiments on a large data set containing 5,000,000 web pages. Of the top 10,000 output web pages with the highest R-SpamRank values, we carried out extensive analyses. In the crawled web pages of these top 10,000 pages, 91.1% of them are spam web pages.

Our algorithm requires a seed blacklist which contains URLs for spam web pages. Right now, the number of URLs in the blacklist is still small, and the domains of these seed URLs are limited to a small set. This limits the capacity of our algorithm to find more link farms under different domains. In our future work, we may gradually enlarge the seed blacklist, thus facilitate the detecting progress.

Acknowledgement

We appreciate the great job done by Lixia Zhen, a test member of Sogou.com search engine, who did a lot of algorithm testing for us. We also thank Ying Wang, a PhD candidate of Institute of Computing Technology, Chinese Academy of Sciences, who gave lots of good advice to our work.

References

[1] L Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. 1998.

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

[3] Z. Gyongyi and H. Garcia-Molina. Web spam taxonomy. Technical report, Stanford Digital Library Technologies Project, Mar. 2004.

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

[5] Zolt´an Gy¨ongyi, and Hector Garcia-Molina. Web Spam Taxonomy. 2005.

[6] Sergey Brin and Lawrence Page, The Anatomy of a Large-Scale Hypertextual Web Search Engine, Proc of the Seventh Int’l World Wide Web Conf. 1998.

[7] Z. Gyongyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In Proceedings of the 30th VLDB Conference, Sept. 2004.

[8] Baoning WU, Brian D.Davison, Cloaking and Redirection: A Preliminary Study. Proceedings of the 1st International Workshop on Adversarial Information Retrieval on the Web (AIRWeb). 2005.

[9] Pr0 - Google's PageRank 0, http://pr.efactory.de/e-pr0.shtml. 2002.

[10] Baoning WU, Brian D.Davison, Identifying link farm spam pages. WWW 2005