ICIC Express Letters
Volume **11**, Number **2**, February **2017**
 ISSN 1881-803X
pp. **269–275**

A WEB SPAM DETECTION ALGORITHM BASED ON PAGERANK

Mei Yu[1,2,3], Ying Meng[1,3], Jian Yu[1,2,*], Xu Zhang[1,2]
Xiaodong Wu[4] and Yichen Tian[1,2]

[1]Tianjin Key Laboratory of Cognitive Computing and Application
[2]School of Computer Science and Technology
[3]School of Computer Software
Tianjin University
No. 92, Weijin Road, Nankai District, Tianjin 300072, P. R. China
{ yumei; yingpu }@tju.edu.cn; *Corresponding author: yujian@tju.edu.cn

[4]Thermal Power Plant of Dushanzi Petrochemical Company
Dushanzi 833600, P. R. China

Received July 2016; accepted October 2016

Abstract. *With the rapid development of society, page sorting has a great meaning to search engines and users. PageRank algorithm is based on link information to sort all the pages, not taking the content of the pages into consideration. For this reason, web spam pages having the higher-than-deserved rankings by link spam cannot be detected. According to the analysis of the deficiencies mentioned above, an improved algorithm called Sim-PageRank is proposed. Sim-PageRank algorithm not only uses PageRank to sort the pages, but also calculates the content similarity between the pages and the spam pages, so it is better to meet authority requirements of web spam detection. In the comparison experiment with PageRank, the recall calculated by the new algorithm is 31%, which is 6.68% higher than that of traditional way only based on PageRank algorithm.*
Keywords: Web spam, PageRank algorithm, Similarity analysis

1. **Introduction.** With the advent of the Internet era, there will be thousands of new pages all over the world every day. Based on the fact that most users will be only interested in the top ranked results search engines return, many spammers attempt to mischievously influence the page ranking produced by search engines. At the beginning, spammers focused on enriching the content of spam pages with specific words that would match query terms. With the rise of link-based ranking techniques, such as PageRank, spammers started to construct spam farms, collections of interlinked spam pages. This latter form of spamming is referred to link spamming as opposed to the former term spamming [1]. Since the PageRank algorithm does not incorporate any knowledge about the content of a site, it is not very uncommon that some spam page receives high PageRank score by link spam. To overcome this limitation, this paper proposes an improved algorithm – Sim-PageRank algorithm. Compared with PageRank, the innovation of Sim-PageRank lies in taking both the link relation and content similarity between pages into account. Especially, content similarity between pages and spam pages is calculated. Through analyzing content similarity, the spam pages having a better rankings by link spam can still be detected because of the higher content similarity. The focus of this paper is to combine the link relation with content similarity to judge whether some page is a spam page or not. On the one hand, the PageRank is used to detect the link relation; on the other hand, the content similarity is viewed as the judgement standard. Experiments prove that Sim-PageRank algorithm is more convincing and accurate than traditional PageRank algorithm.

The remainder of our paper is structured as follows. In Section 2, the related work of web spam detection methods based on PageRank algorithm is introduced. In Section 3,

the Sim-PageRank algorithm combining the content similarity with PageRank algorithm is described in detail. Section 4 proves the validity of Sim-PageRank algorithm through the experiment. Section 5 summarizes the algorithm proposed in this paper and discusses its future prospect.

2. **Related Work.** As PageRank is a popular web page ranking algorithm, lots of people are committed to the improvement of the PageRank algorithm and have made quite a few achievements. [2] first introduced the notion of PageRank. [3] used PageRank algorithm to rank web pages and brought order to the web. [4] proposed topic-sensitive PageRank by computing a set of PageRank vectors to capture more accurately the notion of importance with respect to a particular topic. [5] put forward a new method to quickly compute personalized PageRank algorithm by exploiting graph structures. [6] broke a decade-old performance barrier by proposing an edge-weighted personalized PageRank. On the premise of Web situation changing at any time, in order to update PageRank values quickly, [7] put forward the dynamic figure PageRank calculation.

Although the existing improved PageRank methods have achieved certain effects, they only use the link structure of the Web to capture the relative "importance" of Web pages, entirely independent of the content of pages. In order to deal with the defect, the content similarity between pages can be taken into consideration [8,9]. Inspired by the thoughts mentioned above, Sim-Pagerank algorithm, which takes the content similarity and link relations between pages into account, is proposed to detect Chinese web spam [10].

3. **Sim-PageRank Algorithm.**

3.1. **Problem description.** PageRank algorithm uses link information to assign global importance scores to all pages on the web. Usually, if page A links to page B, the link will be viewed as the forward link of page A, and the back link of page B. The forward link from page A to page B represents page A gives a vote to page B [11]. Correspondingly, PageRank is based on a mutual reinforcement between pages: the importance of a certain page influences and is being influenced by the importance of some other pages. The PageRank score $PR(A)$ of page A is shown in Equation (1) [12].

$$PR(A) = \frac{1-d}{H} + d \sum_{E \in T(A)} \frac{PR(E)}{L(E)} \tag{1}$$

In Equation (1), $PR(A)$ is the PageRank value of page A. H represents the number of all pages. d is the damping coefficient which is a constant between 0 and 1, usually 0.85. $T(A)$ is a collection of pages linking to page A. E represents a page that links to page A. $PR(E)$ is the PageRank value of page E. $L(E)$ represents the number of pages that page E points to. Hence, the PR value of some page A is composed of two parts: one part of the score comes from pages that point to A, and the other (static) part of the value is equal for all web pages.

As PageRank algorithm is only based on link structures between pages to evaluate and rank the importance of a page ignoring the content of pages, it is not surprising that spam pages receive high PageRank scores by link spam to perniciously influence the page rankings. Consequently, this paper adds content similarity between pages and spam pages to PageRank algorithm so as to detect spam pages more effectively.

3.2. **Description of Sim-PageRank algorithm.** We set the total number of the web pages is n that can form an original dataset N, mark x spam pages and store them into the collection X. Then, randomly pick m spam pages from the collection X as the seed collection M.

Step 1: Calculate the maximum similarity value of each page with all the spam pages in the seed set M to generate a similarity set S. Each web page from the original dataset N

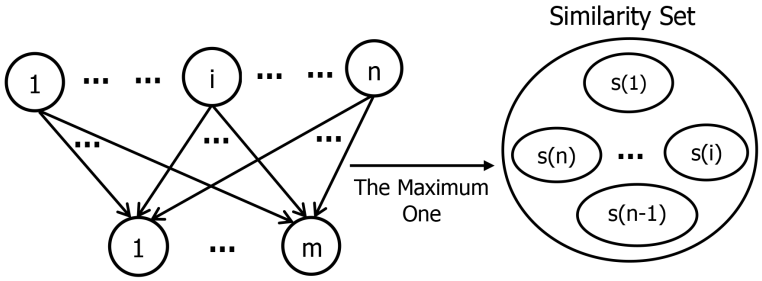

FIGURE 1. Similarity values

needs to respectively do similarity calculation with the spam pages of the seed M. After making similarity calculation, each page in the original dataset N will get m similarity values, but only the max one will be regarded as the corresponding similarity value. Therefore, there will be n similarity values and the n similarity values will form a similarity set S. Meanwhile, the threshold will be set [13]. The process of the maximum similarity selection is shown in Figure 1, and its formula is shown in Equation (2).

$$S = \{s(i)|s(i) = \max\{sim(i,j)\}\} \quad (i \in N,\ j \in M) \tag{2}$$

In Equation (2), $sim(i,j)$ is defined as the similarity value between page i and page j. Table 1 discussed the details of the maximum similarity calculation.

TABLE 1. The calculation of maximum similarity value

Algorithm 1: The maximum similarity calculation

Input:

n – the number of pages in the original dataset
m – the number of spam pages in the seed set
max – the maximum similarity value between some page and spam pages
content_vector – the matrix containing the feature vectors of pages in the original dataset
spamnode_train – the matrix containing the feature vectors of spam pages in the seed set

Output:

sim – content similarity matrix of dimension $n \times 1$

Begin

For i in 1 to n, do
$\mathbf{a} = content\_vector(i,:)$
max = 0 //set max to 0 to find the maximum similarity value
//calculate the maximum similarity value between page i and spam pages
For j in 1 to m, do
$\mathbf{b} = spamnode\_train(j,:)$
$c = sim(\mathbf{a},\mathbf{b}) = dot(\mathbf{a},\mathbf{b})/\left(norm(\mathbf{a}) * norm(\mathbf{b})\right)$
If max $\leq c$, then
do max $= c$
End If
End For
//store the maximum similarity value of page i
$sim(i,1) =$ max
End For

End

Step 2: The PageRank algorithm is used to sort all the web pages in descending order, so as to get a sorted collection U. Moreover, the higher the page ranks, the more important the page will be [14]. If we suppose the page number is proportional to the importance, the sorted collection U can be represented shown in Equation (3).

$$U = \{PR(i)|PR(n) > PR(n-1) > \ldots > PR(i) > \ldots PR(1)\} \quad (i \in N) \tag{3}$$

In Equation (3), $PR(i)$ is defined as the PageRank value of page i.

Step 3: In the sorted collection U, query the similarity values of pages in the reverse order. That is to say that once the $PR(i)$ is given, the corresponding similarity value $s(i)$ could be matched because of unique page number. Firstly, $PR(i)$ which represents the PageRank value of page i is selected. Then, find the index of page i in the original dataset N. As the index in S is the same as that in N, the corresponding similarity value of page i in similarity set S can be found.

Step 4: Compare the similarity value with similarity threshold. If the similarity value of some page is greater than the similarity threshold, this page will be thought as the spam page and be stored into the detected spam set P [15]. Its formula is shown in Equation (4).

$$P = \{p(i)|s(i) < \alpha\} \quad (i \in N, s(i) \in S) \tag{4}$$

In Equation (4), $s(i)$ represents the similarity value of page i, and α represents the similarity threshold. $p(i)$ indicates that page i belongs to the detected spam set P.

Step 5: Repeat Step 4 and Step 5 until all the pages have been detected.

Step 6: If all the pages have been detected, the algorithm ends.

Table 2 discussed the details of Sim-PageRank algorithm.

TABLE 2. The steps of Sim-PageRank algorithm

Algorithm 2: Calculation steps of Sim-PageRank algorithm

Input:
n – the number of pages in the original dataset
PageRank – the matrix containing the PR values of all pages
content_vector – the matrix storing the content features of all pages
sim – the matrix including similarity values of pages
α – similarity threshold
t – the index // set t to 1

Output:
SpamPage – the matrix storing detected spam pages

Begin
For i in n to 1, do
a = PageRank$(i, 1)$
$b = find(content\_vector(:, 1) == a)$
If $sim(b, 1) \geq \alpha$, then
do $SpamPage(t, 1) = a$
$t = t + 1$
End If
End For
End

The overall process of Sim-PageRank is described in Figure 2.

In Figure 2, this paper supposes that each page is composed of z feature vectors, and then the original dataset can be represented as a matrix N_{data_set} of n rows and $(z + 1)$ columns. In this matrix, the first column represents the page number, and the rows represent the pages. The seed set M contains m spam pages, where a matrix M_{seed_set} of

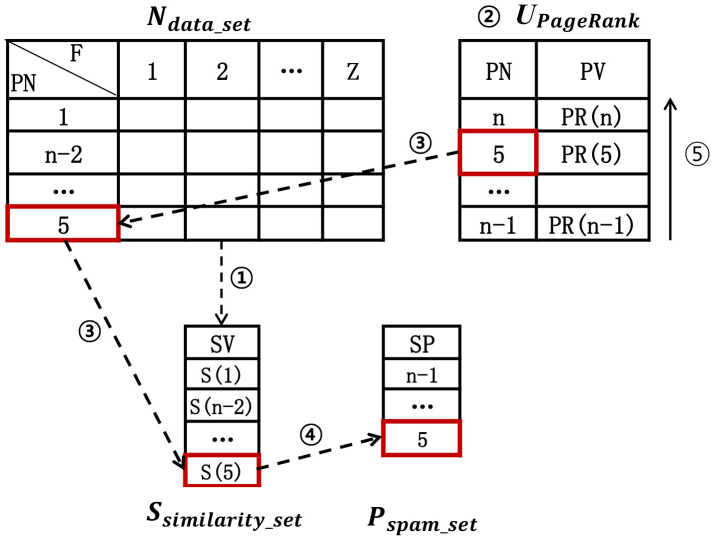

FIGURE 2. The overall process of Sim-PageRank algorithm

m rows and $(z+1)$ columns can be generated. The sorted collection U can be represented as a matrix $U_{PageRank}$ of n rows and 2 columns. In this matrix, the first column represents the page number, and the second column represents the PageRank values. The similarity set can be represented as a matrix $S_{similarity\_set}$ of n rows and 1 column. In this matrix, the column represents the similarity value. Meanwhile, once the spam pages have been detected, a spam matrix P_{spam_set} will be generated. PN represents the page number, F represents the features of pages, PV represents the PageRank values, SV represents the similarity values, and SP represents the spam pages. In addition, page 5 which is a spam page is taken as an example to display the process of Sim-PageRank. From this figure, we can see that after all the pages are sorted in descending order by use of PageRank algorithm, page 5 having a large PR value obtains a high ranking in the sorted $U_{PageRank}$, but it is still detected and stored into the P_{spam_set} because of a large content similarity with spam pages. Through the above analysis, Sim-PageRank algorithm will be better than traditional PageRank algorithm.

4. **Experimental Results.**

4.1. **Dataset.** In order to detect the spam pages, WEBSPAM-UK-2007 data set (http://chato.cl/webspam/datasets/uk2007/) has been used. We select 114529 web pages, among which, 344 are tagged spam pages. Besides, we select 50 spam pages from the tagged spam pages as the seed.

4.2. **Experiment testing and evaluation.** In this paper, the similarity threshold α will be set to 0.91, 0.93, 0.95, 0.97. The experimental results will be compared with those of PageRank. This paper takes the recall to evaluate the experiment results. Its formula is as follows in Equation (5) [16].

$$R = \frac{C_1}{C} \tag{5}$$

In Equation (5), R represents the recall, C_1 represents the total number correctly detected, and C represents the total number of the test set.

In Figure 3(a) and Figure 3(b), the horizontal axis represents different thresholds, the main vertical axis indicates the number of detected spam pages, and the vice one indicates recall. In Figure 3(a), the number and the recall do not change with the threshold. While in Figure 3(b), the higher the threshold is, the larger both the number of detected spam

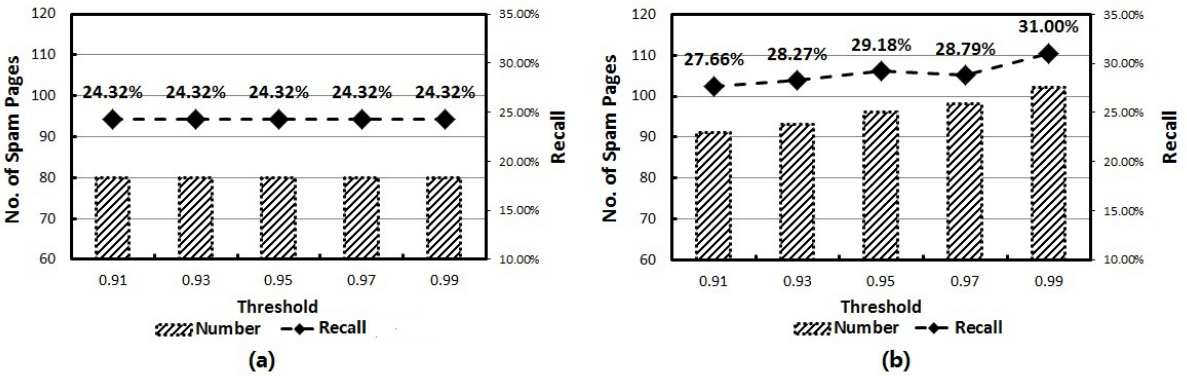

FIGURE 3. Results of PageRank and Sim-PageRank

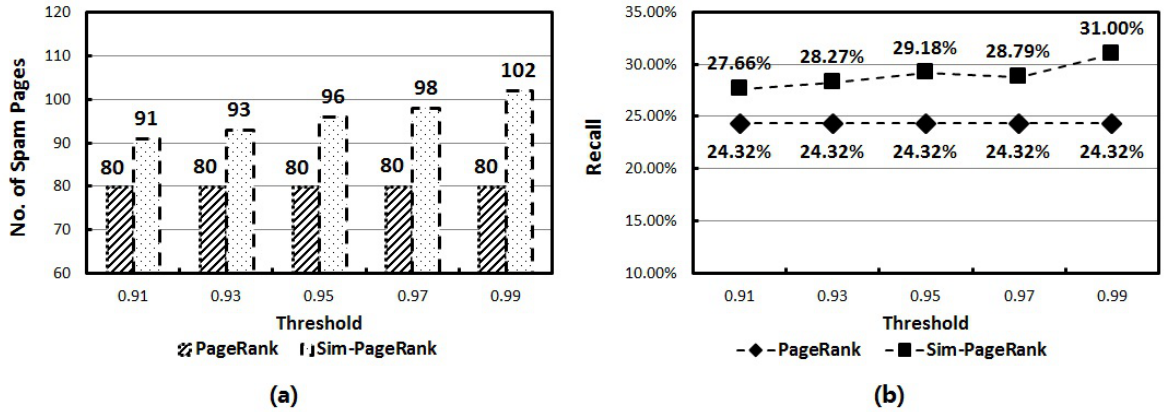

FIGURE 4. Comparison between Sim-PageRank and PageRank

pages and the recall are. Moreover, the values will be maximum when the threshold is set to 0.99.

In Figure 4(a), the horizontal axis represents different thresholds, and the vertical axis indicates the number of detected spam pages. As can be seen from the graph, the Sim-PageRank is better than the PageRank algorithm in the number of detected spam pages. The maximum difference between the two is 22 when the threshold is set to 0.99. In Figure 4(b), the horizontal axis represents different thresholds, and the vertical axis indicates the recall. As we can see from this graph, the performance of the Sim-PageRank is superior to that of the PageRank algorithm in the recall. The minimum difference in recall takes 3.34% when the threshold reaches 0.91, and the maximum difference in recall takes 6.68% when the threshold reaches 0.99.

In conclusion, it could be seen from the above experiment and comparison that the Sim-PageRank algorithm outperforms PageRank algorithm in both the number of detected spam pages and recall. What is more, the effect of spam detection obtained through the improved algorithm is more obvious.

5. **Conclusion and Future Work.** As the rapid development of society, recent years have witnessed a growing interest in the scalability issue of spam detection. In this paper, Sim-PageRank algorithm is proposed, which combines content similarity with PageRank algorithm to avoid the spam pages having a higher ranking. The overall performance shows that the modified algorithm outperforms the PageRank. Besides, the larger the threshold is, the better the detection effect will be. It is our hope that our work will help users enjoy a better search experience on the web.

Victory does not require perfection; in the future, machine learning, such as SVM, can be joined to the Sim-PageRank to further detect spam pages [17]. In addition, the modified algorithm can be applied to the life. When people surf the Internet, search engines not only return the web sites, but also return the similarity scores between the page and the spam pages, decreasing the probability of visiting the spam pages.

Acknowledgment. This work is partially supported by our teacher and friends. We also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] N. Spirin and J. Han, Survey on web spam detection: Principles and algorithms, *ACM SIGKDD Explorations Newsletter*, vol.13, no.2, pp.50-64, 2012.

[2] S. Brin and L. Page, The anatomy of a large-scale hypertextual web search engine, *Computer Networks and ISDN Systems*, vol.30, nos.1-7, pp.107-117, 1998.

[3] L. Page, S. Brin, R. Motwani and T. Winograd, The PageRank citation ranking: Bringing order to the web, *Tech. Rep.*, Stanford University, 1998.

[4] T. Haveliwala, Topic-sensitive PageRank, *Proc. of the 11th International Conference on World Wide Web*, 2002.

[5] T. Maehara, T. Akiba, Y. Iwata et al., Computing personalized PageRank quickly by exploiting graph structures, *Proc. of the VLDB Endowment*, vol.7, no.12, pp.1023-1034, 2014.

[6] W. Xie, D. Bindel, A. Demers et al., Edge-weighted personalized PageRank: Breaking a decade-old performance barrier, *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.1325-1334, 2015.

[7] B. Bahmani, R. Kumar, M. Mahdian and E. Upfal, Pagerank on an evolving graph, *Proc. of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.24-32, 2012.

[8] D. Fetterly, M. Manasse and M. Najork, Detecting phrase-level duplication on the world wide web, *Proc. of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp.170-177, 2005.

[9] K. S. Divya, R. Subha and S. Palaniswami, Similar words identification using naive and TF-IDF method, *American Corrective Therapy Journal*, vol.22, no.5, pp.139-144, 2014.

[10] H. A. Wahsheh, M. N. Alkabi and I. M. Alsmadi, A link and content hybrid approach for arabic web spam detection, *International Journal of Intelligent Systems Technologies and Applications*, vol.5, no.1, pp.30-43, 2013.

[11] M. Zhang, H. Hu, Z. He et al., Efficient link-based similarity search in web networks, *Expert Systems with Applications*, vol.42, no.22, pp.8868-8880, 2015.

[12] X. Wang, J. Ma, K. Bi et al., A improved PageRank algorithm based on page link weight, *Algorithms and Architectures for Parallel Processing*, pp.720-727, 2014.

[13] I. L. Bessas, F. L. C. Pádua, G. T. de Assis et al., Automatic and online setting of similarity thresholds in content-based visual information retrieval problems, *EURASIP Journal on Advances in Signal Processing*, vol.2016, no.1, pp.1-16, 2016.

[14] Z. Gyöngyi, H. Garcia-Molina and J. Pedersen, Combating web spam with TrustRank, *The 30th International Conference on Very Large Data Bases*, pp.576-587, 2004.

[15] F. J. Ortega, C. Macdonald, J. A. Troyano et al., Combining textual content and Hyperlinks in web spam detection, *Proc. of the 16th International Conference on Natural Language Processing and Information Systems*, pp.266-269, 2011.

[16] D. Saraswathi, A. V. Kathiravan and R. Kavitha, Link farm detection using SVMLight tool, *International Conference on Computer Communication and Informatics*, pp.1-5, 2012.

[17] S. P. Algur and N. T. Pendari, Hybrid spamicity score approach to web spam detection, *International Conference on Pattern Recognition, Informatics and Medical Engineering*, pp.36-40, 2012.