

# *N*-Step PageRank for Web Search

Li Zhang<sup>1</sup>, Tao Qin<sup>2</sup>, Tie-Yan Liu<sup>3</sup>, Ying Bao<sup>4</sup>, Hang Li<sup>3</sup>

<sup>1</sup> Department of Mathematics, Beijing Jiaotong University,  
Beijing, 100044, P.R.China  
li.zhang1982@gmail.com

<sup>2</sup>MSPLAB, Dept. Electronic Engineering, Tsinghua University  
Beijing, 100084, P.R. China  
tsintao@gmail.com

<sup>3</sup>Microsoft Research Asia, No. 49, Zhichun Road, Haidian District,  
Beijing, 100080, P. R. China  
{tyliu, hangli}@microsoft.com

<sup>4</sup>Academy of Mathematics and Systems Science, Chinese Academy of Sciences  
Beijing, 100080, P. R. China  
ybao@amss.ac.cn

**Abstract** PageRank has been widely used to measure the importance of web pages based on their interconnections in the web graph. Mathematically speaking, PageRank can be explained using a Markov random walk model, in which only the direct outlinks of a page contribute to its transition probability. In this paper, we propose improving the PageRank algorithm by looking  $N$ -step ahead when constructing the transition probability matrix. The motivation comes from the similar “looking  $N$ -step ahead” strategy that is successfully used in computer chess. Specifically, we assume that if the random surfer knows the  $N$ -step outlinks of each web page, he/she can make a better decision on choosing which page to navigate for the next time. It is clear that the classical PageRank algorithm is a special case of our proposed  $N$ -step PageRank method. Experimental results on the dataset of TREC Web track show that our proposed algorithm can boost the search accuracy of classical PageRank by more than 15% in terms of mean average precision.

## 1 Introduction

PageRank [6] is one of the most successful link analysis algorithms for Web search. PageRank simulates a random walk on the web graph (nodes in the graph represent web pages, and edges represent hyperlinks), and uses the stationary probability of visiting each webpage to represent the importance of that page. Consider a random surfer that is visiting web page  $a$  at present. At each of successive steps, the surfer will proceed from page  $a$  to a web page randomly chosen from all the pages that  $a$  links to. Take *Fig.1* for instance. There are three hyperlinks starting from page  $a$  to pages  $b$ ,  $c$  and  $d$  respectively. The surfer will then visit each of these three pages with a probability of  $1/3$ . In other words, the transition probability of this Markov random walk only depends on the information of the page that is currently being visited.

To our knowledge, however, such a Markov model is not always the best choice in many real-world applications. Take computer chess game for example. The key to the winning of computer “Deep Blue” [2] over human is that it can predict all the situations within much more steps than a human being can do at the same time. That is, if one knows more information, he/she may have more opportunities to make the right decision. Similarly, we argue that if the surfer in the PageRank model can have more information than the direct outlinks, (for example, how many pages one can find by  $N$ -step jumps after choosing one of the direct outlinks), he/she may choose his/her next step with quite different probabilities in order to maximize his/her information gain. This is just the motivation of our proposed  $N$ -step PageRank method. It is clear that this  $N$ -step PageRank is a generalized version of classical PageRank.

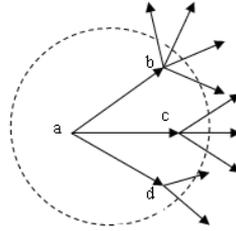


Fig. 1. Illustration of  $N$ -step

To better elaborate on this new method, in Section 2, we will briefly introduce the classical PageRank algorithm. Then in Section 3, we will give the mathematical formulation of the  $N$ -step PageRank and prove how it can be effectively computed. Experimental results are reported in Section 4. Finally, we give the conclusions and future work in the last section.

## 2 PageRank Review

Since our  $N$ -step PageRank is based on PageRank, we shall introduce PageRank in this section.

The directed link graph of the Web is usually modeled as  $G = \langle V, E \rangle$ , where  $V = \{1, 2, \dots, n\}$  is the set of vertices, the elements of which correspond to all the pages on the Web;  $E = \{ \langle i, j \rangle \mid i, j \in V \}$  is the set of edges, the elements of which correspond to the links between web pages (from page  $i$  to page  $j$ ).

Based on this modeling, we can further define the adjacency matrix  $A$  of the link graph as follows,

$$a_{ij} := \begin{cases} 1, & \text{if } \langle i, j \rangle \in E \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

That is, if there is a hyperlink from page  $i$  to page  $j$ ,  $a_{ij}=1$ ; otherwise,  $a_{ij}=0$ .

Most link analysis algorithms are based on this adjacency matrix, including PageRank. PageRank simulates a random walk process on the Web to calculate the

importance of each web page. Suppose there is a surfer in an arbitrary page. At each step, he/she can transit to one of the destination pages of the hyperlinks on the current page with probability  $\alpha$  or to another page randomly in the graph with probability  $1-\alpha$ . Normalize each non-zero row of the adjacency matrix  $A$  with its sum, we will get an initial transition matrix  $P$ . Dealing with zero rows<sup>1</sup>, we will get a probability transition matrix  $\bar{P}$ . Then the above random walk behavior can be modeled as

$$\bar{P} = \alpha \bar{P} + (1-\alpha)U \quad (2)$$

Where  $U$  is a uniform probability transition matrix, all elements of which equal to  $1/n$  ( $n$  is the dimension of  $U$ ).

If we use  $\pi = (\pi_1, \pi_2, \dots, \pi_n)^T$  to denote the stationary distribution of the probability transition matrix  $\bar{P}$ , by the ergodic theory [3], we have:

$$\lim_{m \rightarrow \infty} E \left\{ \frac{1}{m} \sum_{k=0}^{m-1} I_{\{\text{visiting } i \text{ at the } k\text{-th step}\}} \right\} = \lim_{m \rightarrow \infty} \frac{1}{m} \sum_{k=0}^{m-1} p_{ji}^{(k)} := \pi_i, a.s. \quad (3)$$

With the above interpretations, it is reasonable to assume that the more clicks on a webpage, the more important it is. Then the clicking ratio can be interpreted as a measurement of the relative importance of web pages. Thus the stationary distribution  $\pi = (\pi_1, \pi_2, \dots, \pi_n)^T$  can be regarded as the rank scores of the web pages. In fact,  $\pi$  can be computed through the following iterative process.

$$\pi(t+1) = \bar{P}^T \pi(t) \quad (4)$$

Furthermore, it has been proved that the stationary value of  $\pi$  corresponds to the principal eigenvector of  $\bar{P}^T$  when  $\bar{P}^T$  has its unique principal eigenvalue [5].

For clarity, we will denote the above PageRank algorithm [6] as classical PageRank in the following discussions.

### 3 N-step PageRank

In classical PageRank, when the surfer chooses the next webpage, he/she uses only the information of direct outlinks of the current page, i.e., chooses one of the outlink pages with equal probability. In other words, classical PageRank assumes that outlinks are non-distinguishable to the surfer. In this paper, we argue that outlinks can actually be distinguished from many aspects. For example, the surfer may find more useful information or more hyperlinks to new pages after clicking one outlink than the other. Inspired by the “look  $N$ -step ahead” strategy in a computer chess, we propose using the information contained in the next  $N$ -step surfing to represent the information

<sup>1</sup> If the sum of a row is 0, the elements in this row are all given value  $1/n$  after normalization [5].

capacity of an outlink, and thus distinguish different outlinks. To make it clearer, we take *Fig. 1* for example.

Suppose a user is browsing webpage *a*. According to the classical PageRank algorithm, when he selects the next webpage, he can only choose from *b*, *c*, and *d* with equal probability. In our *N*-step PageRank algorithm, the user has more information to decide which page to select. That is, he also knows outlinks of page *b*, *c*, and *d*. For the case of looking 2-step ahead, the probabilities that he selects *b*, *c*, and *d* can be defined as proportional to the outlink numbers of *b*, *c*, and *d*. That is,  $p_{ab}=4/9$ ,  $p_{ac}=3/9$ , and  $p_{ad}=2/9$ . For the case of looking 3-step ahead, the probabilities that he selects *b*, *c*, and *d* will be proportional to the number of web pages he can reach after the next two steps. Recursively, we can come to the conclusion that for the case of looking *N*-step ahead, the probabilities that the surfer selects *b*, *c*, and *d* are proportional to the number of web pages he can reach after the next (*N*-1) steps.

### 3.1 Transition Matrix $P^{(n)}$

After the above intuitive explanation of *N*-step PageRank method, we will give the expression of the corresponding transition probability matrix. Actually, for two

arbitrary vertexes *i* and *j*, we have  $P_{ij}^{(N)} = \frac{d_j^{(N-1)} \mathbb{1}_{\{(i,j) \in E(G)\}}}{\sum_{(i,k) \in E(G)} d_k^{(N-1)}}$ , where  $d_j^{(N)}$  is the vertex number after vertex *j* jump *N* steps, and  $d^{(0)} = (1, 1, \dots, 1)_n^T$ .

**Theorem 1** *The transition matrix of N-step PageRank algorithm  $P^{(N)}$  can be computed as following:*

$$P^{(N)} = (D^{(N)})^{-1} A D^{(N-1)} \quad (5)$$

where *A* is the adjacent matrix of the directed graph *G*,  $D^{(N)}$  is a diagonal matrix generated by the vector  $d^{(N)}$ ,  $d^{(N)} = A d^{(N-1)} = A^N d^{(0)}$ , the elements in  $d^{(N)}$  is the vertex number after each vertex jumps *N* steps<sup>2</sup>

*Proof:* Considering

$$A D^{(N-1)} = \begin{bmatrix} a_{11} \times d_1^{(N-1)}, a_{12} \times d_2^{(N-1)}, \dots, a_{1n} \times d_n^{(N-1)} \\ a_{21} \times d_1^{(N-1)}, a_{22} \times d_2^{(N-1)}, \dots, a_{2n} \times d_n^{(N-1)} \\ \dots \\ a_{n1} \times d_1^{(N-1)}, a_{n2} \times d_2^{(N-1)}, \dots, a_{nn} \times d_n^{(N-1)} \end{bmatrix},$$

we have

---

<sup>2</sup>  $(D^{(N)})^{-1}$  is the extended inverse matrix, and if a certain diagonal element in  $D^{(N)}$  is 0, the corresponding element in  $(D^{(N)})^{-1}$  is 0.

$$(D^{(N)})^{-1}AD^{(N-1)} = \begin{bmatrix} \frac{a_{11} \times d_1^{(N-1)}}{d_1^{(N)}}, \frac{a_{12} \times d_2^{(N-1)}}{d_1^{(N)}}, \dots, \frac{a_{1n} \times d_n^{(N-1)}}{d_1^{(N)}} \\ \frac{a_{21} \times d_1^{(N-1)}}{d_2^{(N)}}, \frac{a_{22} \times d_2^{(N-1)}}{d_2^{(N)}}, \dots, \frac{a_{2n} \times d_n^{(N-1)}}{d_2^{(N)}} \\ \dots \\ \frac{a_{n1} \times d_1^{(N-1)}}{d_n^{(N)}}, \frac{a_{n2} \times d_2^{(N-1)}}{d_n^{(N)}}, \dots, \frac{a_{nn} \times d_n^{(N-1)}}{d_n^{(N)}} \end{bmatrix}$$

Note that  $d^{(N)} = Ad^{(N)}$ , and  $d_i^{(N)} = \sum_{j=1}^n a_{ij}d_j^{(N-1)}$ , then it is easy to verify that

$$P_{ij}^{(N)} = \frac{d_j^{N-1} \mathbf{1}_{\{(i,j) \in E(G)\}}}{\sum_{(i,k) \in E(G)} d_k^{N-1}} = \left[ (D^{(N)})^{-1}AD^{(N-1)} \right]_{ij}$$

The above theorem shows that we can easily get the transition matrix  $P^{(N)}$  of  $N$ -step PageRank from the adjacent matrix  $A$ . Since  $D$  is a diagonal matrix, and some of its diagonal elements may be zeros, the matrix  $P^{(N)}$  is more sparse than the initial matrix  $P$  in classical PageRank which can be used to speed the computation of the stationary distribution.

### 3.2 Convergence rate

Similar to the classical PageRank algorithm, we can obtain irreducible random matrix  $\bar{\bar{P}}^{(N)}$  from  $P^{(N)}$ . According to the ergodic theory [3], the corresponding Markov chain has a unique stationary distribution  $\pi^{(N)}$ , and  $\pi^{(N)} = (\bar{\bar{P}}^{(N)})^T \pi^{(N)}$ . We still use  $\pi^{(N)}$  to measure the importance of web pages. We can use power iteration method to compute the final distribution  $\pi^{(N)}$ , just like equation (5). In this sub section, we will discuss the convergence rate of the corresponding power iteration method.

**Theorem 2** *The stationary distribution is*

$$\pi^{(N)} = \alpha (P^{(N)})^T \pi^{(N)} + \left( \alpha \left( \sum_{i \in D} r_i \pi_i^{(N)} \right) + \frac{1}{n} (1 - \alpha) \right) e_n,$$

where  $\pi^{(N)}(0)$  denotes the initial vector of the iteration,  $\pi^{(N)}(t)$  denotes the vector after  $t$  steps of iteration, and  $D$  is a collection of nodes,

$$D = \left\{ i \mid \sum_{i=1}^n P_{ij}^{(N)} = 0, i \in V \right\}$$

And the convergence rate is

$$\left\| \pi^{(N)} - \pi^{(N)}(t) \right\|_1 \leq \alpha^t \left\| \pi^{(N)} - \pi^{(N)}(0) \right\|_1 \leq 2\alpha^t$$

*Proof:* Omitted since the proof process is very similar to that of classical PageRank.

## 4 Experimental Results

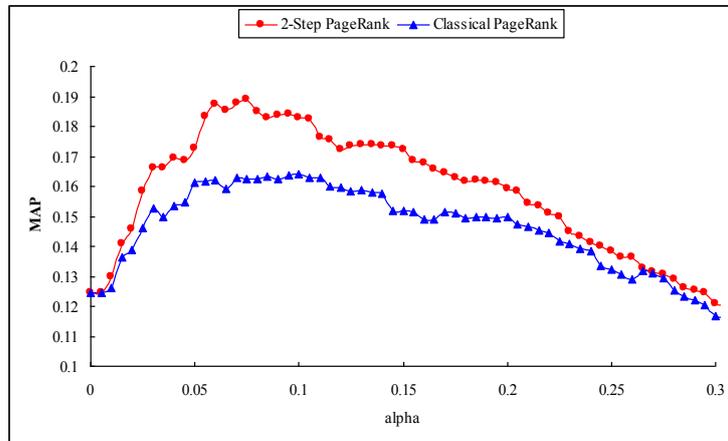
### 4.1 Settings and Results

To compare our  $N$ -step PageRank algorithms with classical PageRank (PR) [6], we chose the topic distillation task in Web track of TREC 2003 as the benchmark. For each query, we first use BM25 [7] to get a list relevance pages. We then choose the top 2000 pages from this list, and combine the relevance score with importance score as follow:

$$score_{combination} = (1 - \alpha) \times score_{importance} + \alpha \times score_{relevance}$$

To evaluate the search accuracy, we adopted two widely-used criteria in our experiments: mean precision at  $n$  ( $P@n$ ) [8], and mean average precision (MAP) [8].

The MAPs of two PageRank algorithms under investigation are shown in *Fig.2*. We can see that our 2-step PageRank almost uniformly outperforms classical PageRank, which shows the effectiveness of considering multi-step hyperlink information.



**Fig. 2.** Search accuracy comparison.

We list the best performance of these two algorithms in Table 1 for further comparison. From this table, the best performance of 2-step PageRank is much better than classical PageRank. The MAP of 2-step PageRank is more than 2.5 points higher than that of classical PageRank, which corresponds to over 15% relative improvement. And 2-step PageRank gets more than 6% relative improvement over

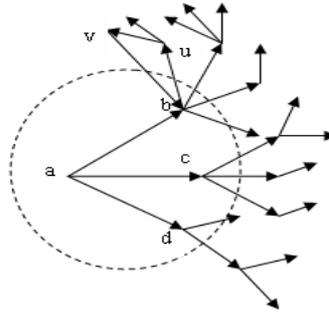
classical PageRank in terms of  $P@10$ . Note that the combination of importance score and relevance score overcomes the relevance score only, which implies the value of link analysis.

**Table 1.** Comparison of different ranking algorithm

Ranking methods	Best MAP	Best P@10
Relevance only	0.1243	0.104
2-step PageRank + relevance	0.1891	0.134
PageRank + relevance	0.1639	0.126
Best Result on TREC2003 [1]	0.1543	0.128

## 4.2 The Influence of Loops

Real-world Web graph may contain loops. One may argue that these loops will influence the result of  $N$ -step PageRank algorithm, and the hyperlink structure is too complex to quantify this influence. For example, in *Fig.3*, the loop  $b \rightarrow u \rightarrow v \rightarrow b$  will make web page  $b$  double counted when computing the outdegree of web page  $b$  in 3-step PageRank.



**Fig. 3.** Illustration of circle in 3-step outlinks

To investigate the influence of the loop, we conducted a small experiment on the case of  $N=2$ . After eliminating loops while counting the 2-step outlinks of a webpage, we find the ranking result is almost the same as the previous result. We will investigate more complicated cases in our future work.

## 5 Conclusions and Future Work

Inspired by the computer chess, in this paper, we pointed out that the random walk model in classical PageRank algorithm could be improved by considering more information. Specifically, we modified the transition matrix of classical PageRank algorithm by using multi-step outlink information, and proposed the  $N$ -step PageRank

algorithm. Experiments on the topic distillation task of TREC2003 showed that the new algorithm outperforms the classical PageRank algorithm.

In the future work, we plan to investigate the following problems:

(1) We have not given an explicit relationship between the stationary distribution of  $N$ -step PageRank and that of classical PageRank in this paper. Though the transition matrices of the two algorithms seem not very complex, they cannot represent each other by elementary transformation. As a result, it is not clear what will happen to the corresponding eigenvectors.

(2) As aforementioned, the “look  $N$ -step ahead” model is to leverage more information when computing page importance. Actually the proposed  $N$ -step PageRank algorithm is just a simple implementation. We will study how to make use of other information, such as website structure.

## 6 ACKNOWLEDGMENTS

We would like to thank anonymous reviewers for their hard work.

## REFERENCES

1. Craswell, N., Hawking, D. (2003). Overview of the TREC 2003 Web Track, in the twelfth Text Retrieval Conference (TREC 2003).
2. Hsu, F.-h. Behind Deep Blue, Princeton University Press, Princeton, NJ, 2002.
3. Kallenberg, O. Foundations of Modern Probability, Page152.
4. Kleinberg, J. Authoritative sources in a hyperlinked environment, Journal of the ACM, Vol. 46, No. 5, pp. 604-622, 1999.
5. Ng, A. Y., Zheng, A. X., and Jordan, M. I. Link analysis, eigenvectors, and stability. In Proc. 17th International Joint Conference on Artificial Intelligence, 2001.
6. Page, L., Brin, S., Motwani, R., and Winograd, T. The PageRank citation ranking: Bringing order to the web, Technical report, Stanford University, Stanford, CA, 1998.
7. Robertson, S. E. Overview of the okapi projects, Journal of Documentation, Vol. 53, No. 1, 1997, pp. 3-7.
8. Salton, G. and McGill, M. J. Introduction to Modern Information Retrieval. McGraw-Hill, 1983.