Link-Based Weighted PageRank

Bing-yuan PU\textsuperscript{1,*} and Chao WANG\textsuperscript{2,3}

\textsuperscript{1}Department of Basic Courses, Chengdu Textile College, Chengdu, China
\textsuperscript{2}School of Math., Phy. and Info. Sci., Zhejiang Ocean University, Zhoushan, China
\textsuperscript{3}Key Lab. of Oceanographic Big Data Mining and Appl. of Zhejiang Province, Zhoushan, China

*Corresponding author

Keywords: Link analysis, PageRank, Algorithm, Weighted PageRank.

Abstract. PageRank has been known to be one of the most popular algorithms in Web ranking. In this algorithm, all links have been treated equally when distributing rank scores. This is not the case in the real world, since some Web resources are important than others and different links should be treated non-identically. In this paper, we adopt three strategies and propose the corresponding weighted algorithms to solve this problem. Numerical experiments illustrate the efficiency of our new algorithms.

Introduction

With the rapid growth of the Web, surfers are much more easily trapped by the rich hyper structure and it is a challenging work to provide surfer with relevant and valuable information. Fortunately, there are various Web search engines (about 3500 different search engines \cite{1,2}, by which we can find useful information efficiently). Notess \cite{3} and Ward \cite{4} found that Google is unique in its focus on developing the perfect Web search engine that understands accurately what users mean and gives them back precisely the desired information. As you know, PageRank has become a key element in the success of Google and has contributed considerably to Google's lasting dominance in the search engine market \cite{5}.

PageRank, a link-based algorithm formulated by Brin and Page \cite{6,7}, gives a rank list of importance of pages related to user's query terms. This algorithm follows rules as \cite{6}: a link from page A to page B is a vote from A to B, highly linked pages are more important than pages with few links. In this algorithm, the rank score of a page, named page's PageRank value(PR), is evenly divided among its outlinks, and an approximation iterative computation is usually applied to calculate the PR. Despite this method can be carried out easily, however, its weakness is obvious. In view of the fact that some Web resources are more important than others and some pages outlinked from a page A may be more important than other pages, the more important pages should receive higher votes from A, not totally equivalent with the less important ones do. Distributing rank scores evenly among its outlinked pages, not only influences the ranking quality, but also provides chances for business speculation or even spamming.

Many algorithms have been developed to improve the performance of PageRank \cite{8-12}. These algorithms mainly focus attention on the convergence, iteration, acceleration, damping factor, etc. However, discussion about PR average-distribution appears lacking in this area. I'd like to mention that \cite{13} attempts to resolve this problem by introducing the number of inlinks and outlinks of a page to weigh page's popularity. In our opinion, page's popularity shouldn't depend only on its number of links, and obviously web spam is a typical example of most convincing. In view of this, when distributing a page's score among its outlinked pages, we should reference the numbers of inlinks and outlinks since they carry valuable information, and at the same time, the importance of pages should also be taken into account. Inspired by this, we adopt three strategies to assign different weights to each link and then propose the corresponding link-based weighted PageRank algorithms. Theoretical
analysis manifests the convergence of the new algorithms. Numerical experiments show that the proposed algorithms perform much better than the standard method for PageRank problems.

The remainder of this paper is organized as follows. In Section 2, we introduce the brief review of PageRank algorithm. In Section 3, we propose three weighted strategies to improve the standard PageRank algorithm. Convergence analysis is also investigated in this Section. Finally, some numerical experiments are provided in Section 4.

**Brief Review of PageRank**

The basic idea behind PageRank is that if page \( v \) has a link to page \( u \), then page \( v \) confers some importance to page \( u \). Link-based PageRank views the Web as a directed graph \( G \), where each node represents a page and each edge from node \( v \) to node \( u \) represents the existence of a link from page \( v \) to page \( u \). Let \( n \) be the total page number, \( n_v \) the outdegree of page \( v \), and let \( PR(p) \) represents the importance (PageRank value) of page \( p \). Then by link\((v,u)\), page \( v \) distributes its \( PR(v) \) evenly to all its outlinks and so confers \( PR(v)/n_v \) to page \( u \). Thus suggests an iterative fixed-point computation for determining the importance for every page on the Web. The simplified version of PageRank is defined as follows [7]:

\[
PR(u) = \sum_{v \in B(u)} \frac{PR(v)}{n_v},
\]

(1)

where \( B(u) \) is the set of pages that point to \( u \).

The PR of every page can be calculated recursively starting with any set of ranks and iterated until it converges. This definition, however, raises a small problem: rank sink (a group of pages pointing to each other have some links to the group but no links going out). To solve this problem, by introducing a decay factor \( c \), Page and Brin[7] deduce the classical formula for PageRank computation as follows:

\[
PR(u) = \frac{1-c}{N} + c \times \sum_{v \in B(u)} \frac{PR(v)}{N_v},
\]

(2)

where \( c \) is a damping factor that is usually set to 0.85. This definition of PageRank has another intuitive display of random walk on the graph of the Web. If viewing a surfer as a random walker on the Web, the constant \( c \) can be explained below: a Web surfer visiting a page periodically “gets bored” and jumps to any other page on the Web with probability \( 1-c \) rather than following an outlink. Furthermore, the computation of PageRank can be convert to the computation of stationary distribution vector of the linear system

\[
\pi^T = \pi^T P' = \pi^T (c(P + dV^T) + (1-c)EV^T),
\]

(3)

where \( P \) is the adjacency matrix of the web graph \( G \) normalized so that each row sums to 1, \( E \) is the vector with all entries equal to 1, \( V \) is the column vector representing a uniform probability distribution over all nodes with \( V = [1/n]_{\text{all}} \), \( d = 1 \) while page \( i \) is dangling and \( d = 0 \) otherwise. It is generally known that matrix \( P' \) is usually called the Google matrix. By the Perron-Frobenius Theorem[14], this calculation can converge to a unique rank vector, i.e., the principal eigenvector of the Google matrix.

**Link-Based PageRank**

As aforementioned, a page distributes its PR evenly to all its outlinks in the original PageRank algorithm. However, in fact, different hyperlinks should have different importance. For example, a hyperlink from the portal of a website should have more stronger recommendation than a hyper-link from a common page to the portal. Hence, when distributing \( PR \) to outlinked pages, a link ranking
algorithm that gives different weights to outlinks should improve over the uniform weight distributing ranking algorithm, i.e., PageRank. With such a motivation, we propose three strategies for improving the standard PageRank algorithm. The first one involves taking into account the number of inlinks and outlinks of a page. The second one introduces a weight according to the importance of each page i.e., applying the PageRank values that have been obtained in the recursive iterating circle. In the end, we propose the hybrid strategy based on the former two. In following sections, we will show the performance of these strategies in details.

**Strategy I.** Firstly, let us be assured of one fact: a popular web page is often referred to by other pages, and so an important webpage contains a high number of outlinks. Furthermore, one page outlinked to by the important page should be important. So the number of hyperlinks (outlinks and inlinks) must be a key index to evaluate a page's importance. Specifically, when there is a link from page \( v \) to page \( u \) and we consider the \( PR \) distributed from \( v \), the page \( u \) gets a value proportional to its weight,

\[
W_{(v,u)}^I = \frac{I_u + O_u}{\sum_{i \in B(v)} (I_i + O_i)},
\]

where \( I_u \) and \( O_u \) represent the number of inlinks and outlinks of page \( u \), respectively, \( I_i \) and \( O_i \) represent the number of inlinks and outlinks of page \( i \), respectively. \( B(v) \) denotes all the outlinked pages from page \( v \).

The modified PageRank algorithm is as:

\[
PR(u) = \frac{1-c}{N} + c \times \sum_{v \in B(u)} W_{(v,u)}^I PR(v).
\]

Accordingly, the linear system (3) can be refined with only a little modification to the adjacent matrix of Web graph. So, the PageRank vector \( \pi = (\pi_1, \pi_2, \cdots, \pi_n)^T \) can be computed through the below iterative process:

\[
\pi^{(k+1)} = A^T \pi^{(k)},
\]

where \( A \) is the modified Google matrix according (4). Next, let's briefly discuss the convergence of the proposed weighted PageRank algorithm and the continuing other weighted algorithms can be done similarly.

Assume \( A' \) is the adjacency matrix of the web Graph by Strategy I. Replacing \( P \) in (3) by \( A' \), we get \( A = c(A' + dV^T) + (1-c)EV^T \). Because \( A' \) is finite and non-negative, \( A' + dV^T \) is finite and non-negative. Thus \( A \) must be finite and absolutely positive. Therefore \( A \) is an irreducible probability transition matrix. By Perron-Frobenius Theorem [14], it must have stationary distribution which can be computed as follows.

\[
\pi = A^\pi \pi.
\]

Obviously, the stable value of \( \pi \) corresponds to the principle eigenvector of \( A^\pi \).

**Strategy II.** In this section, we will propose another strategy to overcome the shortcoming of evenly-distributed PageRank. In view of this fact: surfer would more likely follow the outlinks of a popular page rather than a lower one, and pages that are referenced by good pages should have higher PageRank value, we should make a distinction among the links of a page. Then comes the problem of how to distinguish between two links. Strategy II assigns a weight \( W_{(v,u)}^II \) as follows to each link based on the importance of the outlinked pages.
where $B(v)$ is the set of all outlinked pages of page $v$.

$$W^H_{(v,u)} = \sum_{i \in B(v)} \frac{PR(u)}{PR(i)}, \quad (8)$$

For example, consider a hypothetical website in Figure 1, where the numbers represent the $PR$ values during the iteration. Page $p_5$ has two reference pages: $p_3$ and $p_4$. Then in the next iteration, the weights of link $(p_5, p_3)$ and link $(p_5, p_4)$ should be

$$W^H_{(p_5, p_3)} = \frac{0.032}{0.032 + 0.043} = 0.427, \quad W^H_{(p_5, p_4)} = \frac{0.043}{0.032 + 0.043} = 0.573.$$

The second modified PageRank algorithm is as:

$$PR(u) = \frac{1-c}{N} + c \times \sum_{v \in B(u)} W^H_{(v,u)} PR(v). \quad (9)$$

**Strategy III.** In the actual Web, the popularity of a page is not only related with the numbers of links (inlinks and outlinks), but also related with its importance (PR value). Hence, in this strategy, we combine the Strategy I with Strategy II, and then introduce a hybrid weight

$$W^H_{(v,u)} = W^I_{(v,u)} W^H_{(v,u)}, \quad (10)$$

or as the following convex linear combination,

$$W^H_{(v,u)} = \alpha W^I_{(v,u)} + (1-\alpha) W^H_{(v,u)}. \quad (11)$$

The corresponding Hybrid PageRank algorithm can be formulated as follows:

$$PR(u) = \frac{1-c}{N} + c \times \sum_{v \in B(u)} W^H_{(v,u)} PR(v). \quad (12)$$

**Numerical Experiments**

In this section, we make some numerical experiments to show the efficiency of our new algorithms. The notation used for the various algorithms is as follows. “PR” refers to the standard PageRank algorithm. “WPR1”, “WPR2” and “HPR” refer to the weighted PageRank methods with respects to Strategy I, Strategy II and Strategy III, respectively. The weight for Strategy III is calculated from (10).

**Example 1.** This experiment aims to evaluate the performance and efficiency of our proposed ranking strategy, compared to the standard PageRank algorithm. In this example, the TREC2003 test collection is used, which was crawled from .gov Web sites in early 2002. It contains 1,247,753
documents, 1,053,111 of which are text/html files together with 11,164,829 hyperlinks in it. There are totally 50 topic distillation queries with 10.32 relevant documents on average.

In order to measure the retrieval performance of different ranking algorithms, we use precision at 10 documents (P@10) and Mean Average Precision (MAP) as our evaluation metrics. As the baseline, BM2500[15] is adopted as the relevance weighting function. According to the BM2500 score, we choose the top 2000 results and thus we get two ranking lists: one is about relevance and the other is about importance (PR or WPR). We integrate the relevance score and the rank score linearly into the final score as follows:

\[ \beta \cdot \text{rank}_{\text{relevance}} + (1 - \beta) \cdot \text{rank}_{\text{PageRank}}. \]

We then rerank the 2000 documents according to the composite scores and select the top documents for evaluation. The P@10 and MAP with different \( \beta \) are shown in Figure 2 and the best results are listed in Table 1.

Table 1. Best performance of each algorithms for Example 1.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Baseline</th>
<th>PR</th>
<th>WPR1</th>
<th>WPR2</th>
<th>HPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>P@10</td>
<td>0.112</td>
<td>0.137</td>
<td>0.139</td>
<td>0.141</td>
<td>0.143</td>
</tr>
<tr>
<td>MAP</td>
<td>0.128</td>
<td>0.148</td>
<td>0.151</td>
<td>0.158</td>
<td>0.163</td>
</tr>
</tbody>
</table>

As can be seen from Figure 2, all the curves of different PageRank algorithms converge to the baseline when \( \beta=1 \). It is obvious that the weighted PageRank algorithms outperform the standard algorithm and the hybrid algorithm is the optimal one in all the listed algorithms for PageRank problem. Furthermore, it can be seen from Table 1 that the weighted strategies improve the performance of the PageRank algorithm significantly, both in terms of the best results of P@10 and MAP. For example, the best P@10 of the HPR, 0.143 when \( \beta=0.87 \), gets 27.68% and 4.38% improvements over the best results of the baseline and PR, respectively. And the best MAP of HPR, 0.163, is achieved when \( \beta=0.94 \). This corresponds to 27.34% and 10.14% improvements over the best results of the baseline and PR, respectively.

![Figure 2. Numerical comparisons of P@10 and MAP for Example 1.](image)

**Acknowledgement**

This research was supported by Key Fund Project of Sichuan Provincial Department of Education (17za0003).
References


