A note on the PageRank algorithm

Huan Sun 1, Yimin Wei *

School of Mathematical Sciences, Fudan University and Key Laboratory of Mathematics for Nonlinear Sciences, Ministry of Education, Shanghai 200433, PR China

Abstract

In this paper we present some notes of the PageRank algorithm, including its $L_1$ condition number and some observation of the numerical tests of two variant algorithms which are based on the extrapolation method.

Keywords: PageRank; Condition number; Eigenvalue problem

1. Introduction

With the booming development of the Internet, web search engines [2–5,10,12,14,16–18] have become the most important Internet tools to retrieve information. Among thousands of web search engines based on various algorithms that have emerged in recent years (there had been more than 3500 different search engines by the year 2000 according to [13,15]), Google has become one of the most popular and successful one. Google’s triumph should largely attribute to its simple but elegant algorithm: PageRank.

PageRank is a link structure-based algorithm, which gives a rank of importance of all the pages crawled in the Internet by the Google’s web crawler. To compute a PageRank is actually to compute the stable distribution of a transition matrix, also called the Google matrix, which is based on the web graph structure. The overwhelming size of the Google matrix pales many popular and intricate algorithms, which may otherwise have excellent performances in those normal scale computing. By comparison, the simple power method stands out for its stable and reliable performances in spite of its low efficiency. So variants are introduced to improve its convergence speed, which is part of our work in this paper.

We will begin our paper with a brief review of PageRank algorithm and then display our work in the following sections.

---

* This project is supported by the National Natural Science Foundation of China under Grant 10471027 and Shanghai Education Committee.

* Corresponding author.

E-mail addresses: sun_h@math.psu.edu (H. Sun), ymwei@fudan.edu.cn (Y. Wei).

1 Present address: Department of Mathematics, The Pennsylvania State University, USA.

0096-3003/S - see front matter C 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2005.11.120
2. Notations and preliminaries

We consider a directed graph based on the hyperlink structure of the global web \([4,5]\), in which every page stands for a node and there is an edge from node \(u\) to \(v\) if there is a link from page \(u\) to \(v\). Then we define the matrix \(P\) elementwisely as follows:

\[
p_{i,j} = \begin{cases} 
1/|P_i|, & \text{if page } P_i \text{ links to page } P_j, \\
0, & \text{otherwise,}
\end{cases}
\]

where \(|P_i|\) denotes the outdegree of page \(P_i\). To make \(P\) a transition probability matrix, we modified it as:

\[
P = P + d v^T,
\]

where \(d = \{1, \ deg(i) = 0, 0\}\) is a distribution vector. Google usually set it as uniformly distributed, i.e., \(v = (1,1,\ldots,1)/n\).

In order to force the irreducibility, we make a further modification of \(\bar{P}\):

\[
\bar{P} = cP + (1-c)e v^T,
\]

where \(e = (1,1,\ldots,1)^T\) and \(c\) is a parameter ranging from 0 to 1.

For convenience, we usually use the transpose of \(\bar{P}\) in computation and call it the Google matrix, denoted as \(G\).

\[
G = \bar{P}^T = [cP + (1-c)e v^T]^T
\]

and the very famous PageRank is the stable distribution of \(G\), i.e.,

\[
G \pi = \pi.
\]

3. \(L_1\) norm of the PageRank problem

We convert the eigen problem of the Google matrix into a following linear equations problem by transforming Eq. (4):

\[
e^T \pi = \|\pi\|_1 = 1,
\]

and

\[
\pi = \bar{P}^T \pi = cP^T \pi + (1-c)v,
\]

and

\[
(1-c\bar{P}^T)\pi = (1-c)v.
\]

Then the following proposition holds:

**Proposition 1.** If \(c \neq 1\), the \(L_1\) norm of \(1 - c\bar{P}^T\) equals to \(\frac{1+c}{1-c}\), i.e., \(\kappa_1(1-c\bar{P}^T) = \frac{1+c}{1-c}\).

**Proof.** According to the definition of \(\bar{P}\), it is easily verified that

\[
\|I - c\bar{P}^T\|_1 = \|I\|_1 + c\|P^T\|_1 = 1 + c.
\]

On the other hand, the Gerschgorin Theorem [6] assures the existence of \((I - c\bar{P}^T)^{-1}\), so we have

\[
(1-c\bar{P}^T)\pi = (1-c)v,
\]

and

\[
\pi = (1-c)(I - c\bar{P}^T)^{-1}v,
\]

and

\[
\|\pi\|_1 \leq (1-c)\|(I - c\bar{P}^T)^{-1}\|_1\|v\|_1,
\]

and

\[
\|(I - c\bar{P}^T)^{-1}\|_1 \geq \frac{1}{1-c}.
\]
and because \( \|c \mathbf{P}\|_1 < 1 \), the following holds
\[
\|(I - c \mathbf{P})^{-1}\|_1 \leq \frac{1}{1 - \|c \mathbf{P}\|_1} = \frac{1}{1 - c}.
\]
Therefore,
\[
\|(I - c \mathbf{P})^{-1}\|_1 = \frac{1}{1 - c}
\]
and
\[
\kappa_1(I - c \mathbf{P}) = \|I - c \mathbf{P}\|_1 \|(I - c \mathbf{P})^{-1}\| = \frac{1 + c}{1 - c}.
\]

Note. S. Kamvar and T. Haveliwala gives the same equation in [11]. But their conclusion is based on the presupposition that all pages have a outdegree larger than 0. □

4. Extrapolation methods for accelerating PageRank method

Here we quote two extrapolation methods: Aitken extrapolation and Quadratic extrapolation [21,12], both of which try to approximate the stationary distribution.

Suppose we obtain \( \pi^{(k)} \) after \( k \) iterations. And now we approximate it with the first two eigenvectors \( \pi, u_2 \):
\[
\pi^{(k)} = \pi + \alpha u_2.
\] (8)

Then
\[
\pi^{(k+1)} = G \pi^{(k)} = \pi + \alpha \lambda_2 u_2,
\]
\[
\pi^{(k+2)} = G \pi^{(k+1)} = \pi + \alpha \lambda^2 u_2.
\] (9)

Define \( g, h \) as
\[
g_i = (\pi^{(k+1)}_i - \pi^{(k)}_i)^2,
\]
\[
h_i = \pi^{(k+2)}_i - 2 \pi^{(k+1)}_i + \pi^{(k)}_i,
\] (10) (11)

So we get
\[
g_i = \alpha^2 (\lambda_2 - 1)^2 (u_2(i))^2,
\]
\[
h_i = \alpha (\lambda_2 - 1)^2 (u_2(i)).
\] (12) (13)

If \( h_i \neq 0 \), then define vector \( f \)
\[
f_i = \frac{g_i}{h_i} = \alpha u_2(i),
\]
i.e.,
\[
f = \alpha u_2.
\]

So the dominating eigenvector can be expressed by:
\[
\pi = \pi^{(k)} - f.
\] (14)

Then we use a combination of the previous iterations to get a new approximation to the true eigenvector. As we regard \( \pi^{(k+2)} \) is closer to the true value than \( \pi^{(k)} \), so we replace the latter with the former in our practice. Besides, we have to re-normalize the computing result by setting as zero those negative items caused by the subtraction operation.

**Algorithm 1 Aitken extrapolation**

\[
\text{function } y = \text{Aitken extrapolation}(\pi^{(k)}, \pi^{(k+1)}, \pi^{(k+2)})
\]
\[
g = (\pi^{(k+1)} - \pi^{(k)})^2;
\]
\[ h = \pi^{(k+2)} + 2\pi^{(k+1)} + \pi^{(k)}; \]
\[ f = g/h; \]
\[ y = \pi^{(k+2)} - f; \]
\[ \text{renormalize } y \]
\[ \text{return} \]

We apply this extrapolation in the power method to accelerate it. The numerical test results are illustrated in the following graph (see Fig. 1).

The above numerical test is based on the database \textsc{stanford.edu}, which is available at http://www.stanford.edu/~sdkamvar [6–8].

The graph illustrates that in the next iteration after Aitken extrapolation is implemented, the residual increases sharply, which causes a big spike. But the residual drops more quickly than the standard power method after the spike. Though such an optimization is not guaranteed, as is illustrated in the case \( c = 0.99 \), Aitken extrapolation is still valuable considering its small cost of flop.

In a similar manner, we approximate the iterations with the first three eigenvectors of the Google matrix, which is called the quadratic extrapolation.

Suppose the Google matrix \( G \) has only three linear independent eigenvectors \( \pi, u_2, u_3 \). And we approximate the \( k \)th iteration with the three ones

\[ \pi^{(k)} = \pi + \alpha_1 u_2 + \alpha_2 u_3. \]

Since we assume \( G \) has three independent eigenvectors, the minimal polynomial of \( G \) is cubic.

\[ p_G(\lambda) = \gamma_0 + \gamma_1 \lambda + \gamma_2 \lambda^2 + \gamma_3 \lambda^3. \]
For the polynomial has a root 1, then
\[ \gamma_0 + \gamma_1 + \gamma_2 + \gamma_3 = 0. \]

Because
\[ p_G(G) = 0, \]
and \[ p_G(G)\pi^{(k)} = 0, \]
and
\[ \gamma_0\pi^{(k)} + \gamma_1\pi^{(k+1)} + \gamma_2\pi^{(k+2)} + \gamma_3\pi^{(k+3)} = 0, \]
and
\[ (-\gamma_1 - \gamma_2 - \gamma_3)\pi^{(k)} + \gamma_1\pi^{(k+1)} + \gamma_2\pi^{(k+2)} + \gamma_3\pi^{(k+3)} = 0. \]

We define the following vectors:
\[
\begin{align*}
\gamma^{(k+3)} &= \pi^{(k+3)} - \pi^{(k)}, \\
\gamma^{(k+2)} &= \pi^{(k+2)} - \pi^{(k)}, \\
\gamma^{(k+1)} &= \pi^{(k+1)} - \pi^{(k)}
\end{align*}
\]

and
\[ \gamma = (\gamma_1, \gamma_2, \gamma_3)^T. \]

So we have
\[ (\gamma^{(k+1)}, \gamma^{(k+2)}, \gamma^{(k+3)})\gamma = 0. \]

We can set \( \gamma_3 = 1 \) for convenience, and get
\[ (\gamma^{(k+1)}, \gamma^{(k+2)}) \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = Y \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = -\gamma^{(k+3)}. \]

This is an overdetermined equation and we can approximate the solution by Moore-Penrose generalized inverse [9,20]:
\[ \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = -Y^+\gamma^{(k+3)}. \]

Then we define
\[ q_G(\lambda) = \frac{p_G(\lambda)}{\lambda - 1} = \beta_0 + \beta_1\lambda + \beta_2\lambda^2 \]
and according to the definition of minimal polynomial, we have
\[ q_G(G)u_i = 0, \quad (i = 2, 3), \]
where \( u_2 \) and \( u_3 \) are the second and the third eigenvectors.

Then we can approximate the iteration with the three eigenvectors
\[ \pi = G\pi \]
\[ = Gq_G(G)\pi^{(k)} / (\beta_0 + \beta_1 + \beta_2) \]
\[ = (q_G(G)\pi^{(k+1)}) / (\beta_0 + \beta_1 + \beta_2) \]
\[ = (\beta_0\pi^{(k+1)} + \beta_1\pi^{(k+2)} + \beta_2\pi^{(k+3)}) / (\beta_0 + \beta_1 + \beta_2). \]

It is easy to determine the coefficients \( \beta_i \) from the relation of \( p_G \) and \( q_G \):
\[ \beta_0 = \gamma_1 + \gamma_2 + \gamma_3, \]
\[ \beta_1 = \gamma_2 + \gamma_3, \]
\[ \beta_2 = \gamma_3. \]
Algorithm 2 Quadratic extrapolation algorithm

function y = Quadratic extrapolation($\mathbf{p}^{(k)}$, $\mathbf{p}^{(k+1)}$, $\mathbf{p}^{(k+2)}$, $\mathbf{p}^{(k+3)}$)
for $i = k + 1$ to $k + 3$
    $y^{(i)} = \frac{\mathbf{p}^{(i)}}{C_0}$;
end
$Y = (y^{(k+1)}, y^{(k+2)})$;
$\gamma_3 = 1$;
$\begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix} = -Y^T y^{(k+3)}$;
$\beta_0 = \gamma_1 + \gamma_2 + \gamma_3$;
$\beta_1 = \gamma_2 + \gamma_3$;
$\beta_2 = \gamma_3$;
$y = \beta_0 \mathbf{p}^{(k+1)} + \beta_1 \mathbf{p}^{(k+2)} + \beta_0 \mathbf{p}^{(k+3)}$;
renormalize $y$
return

We note that though both two extrapolation methods suffer spikes in the residual, the quadratic extrapolation have a smaller one because it approximates the iterations with three eigenvectors while Aitken extrapolation does with two (see Fig. 2).

5. Restarting power method

To efficiently update the stable distribution of a Markov chain [19] has been studied for a long time and there are various methods. Restarting power method with the previous stable distribution as the initial iteration, simple as it is, has proved quite efficient in our numerical experiment.
In our experiment, we update the web by adding some new pages, whose outdegrees meet the normal distribution of total web outdegrees. And we compare the effect of restarting with the old distribution with those restarting with uniform distribution and randomized distribution as the initial vector, respectively. The result is illustrated as follows:

It is obvious that restarting with the old PageRank converges more quickly than with the other two initial vectors. But the preponderance is diminished as the number of new pages added becomes larger (see Fig. 3).

6. Conclusion and areas of future work

This paper gives some propositions, algorithms and numerical tests on the PageRank problem. The $L_1$ norm of PageRank problem shows that it is a well-posed problem when $c$ is not close to 1, thus computing PageRank will not cause large round-off.

The numerical tests shows the extrapolation method is a promising variant to compute PageRank, though not stable yet. Further improvements is needed to make the algorithm have a stable performance [1] (see Table 1).

Table 1
The comparison of the iteration steps of power method and its variants

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>$c = 0.85$</th>
<th>$c = 0.9$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard power method</td>
<td>91</td>
<td>138</td>
</tr>
<tr>
<td>Aitken extrapolation at 10th iteration</td>
<td>100</td>
<td>145</td>
</tr>
<tr>
<td>Aitken extrapolation at 20th iteration</td>
<td>90</td>
<td>134</td>
</tr>
<tr>
<td>Quadratic extrapolation at 10th iteration</td>
<td>81</td>
<td>124</td>
</tr>
<tr>
<td>Quadratic extrapolation at 20th iteration</td>
<td>77</td>
<td>119</td>
</tr>
</tbody>
</table>

In our experiment, we update the web by adding some new pages, whose outdegrees meet the normal distribution of total web outdegrees. And we compare the effect of restarting with the old distribution with those restarting with uniform distribution and randomized distribution as the initial vector, respectively. The result is illustrated as follows:

It is obvious that restarting with the old PageRank converges more quickly than with the other two initial vectors. But the preponderance is diminished as the number of new pages added becomes larger (see Fig. 3).

6. Conclusion and areas of future work

This paper gives some propositions, algorithms and numerical tests on the PageRank problem. The $L_1$ norm of PageRank problem shows that it is a well-posed problem when $c$ is not close to 1, thus computing PageRank will not cause large round-off.

The numerical tests shows the extrapolation method is a promising variant to compute PageRank, though not stable yet. Further improvements is needed to make the algorithm have a stable performance [1] (see Table 1).

References
