Chapter 6

PageRank

Bing Liu and Philip S. Yu

Contents

6.1 Introduction

Link-based ranking has contributed significantly to the success of Web search. PageRank [1, 7] is perhaps the best known link-based ranking algorithm, which also powers the Google search engine. Due to the huge business success of Google, PageRank has emerged as the dominant link analysis model on the Web.

The PageRank algorithm was first introduced by Sergey Brin and Larry Page at the *Seventh International World Wide Web Conference* (*WWW7*) in April 1998, with the aim of tackling some major difficulties with the content-based ranking algorithms of early search engines. These early search engines essentially retrieved relevant pages for the user based on content similarities of the user query and the indexed pages of the search engines. The retrieval and ranking algorithms were simply direct implementation of those from information retrieval. However, starting from 1996, it became clear that the content similarity alone was no longer sufficient for search due to two main reasons. First, the number of Web pages grew rapidly during the middle to late 1990s. Given any query, the number of relevant pages can be huge. For example, given the search query "classification technique," the Google search engine estimates that there are about 10 million relevant pages. This abundance of information causes a major problem for ranking, that is, how to choose only 10 to 30 pages and rank them suitably to present to the user. Second, content similarity methods are easily spammed. A page owner can repeat some important words and add many remotely related words in his/her pages to boost the rankings of the pages and/or to make the pages relevant to a large number of possible queries.

From around 1996, researchers in academia and search engine companies began to work on the problem. They resort to hyperlinks. Unlike text documents used in traditional information retrieval, which are often considered independent of one another (i.e., with no explicit relationships or links among them except in citation analysis), Web pages are connected through hyperlinks, which carry important information. Some hyperlinks are used to organize a large amount of information at the same Web site, and thus only point to pages in the same site. Other hyperlinks point to pages in other Web sites. Such outgoing hyperlinks often indicate an implicit conveyance of authority to the pages being pointed to. For example, if your page points to an outside page, you obviously believe that this outside page contains quality and useful information to you. Hence, those pages that are pointed to by many other pages are likely to contain authoritative or quality information. Such linkages should obviously be used in page evaluation and ranking in search engines. PageRank precisely exploits such links to provide a powerful ranking algorithm. In essence, PageRank relies on the democratic nature of the Web by using its vast link structure as an indicator of an individual page's quality. It interprets a hyperlink from page x to page y as a vote, by page x, for page y. Additionally, PageRank looks at more than just the sheer number of votes or links that a page receives. It also analyzes the page that casts the vote. Votes cast by pages that are themselves "important" weigh more heavily and help to make other pages more "important." This is the *rank prestige* idea in social networks [9]. In this chapter, we introduce the PageRank algorithm. Along with it, an extension to the algorithm is also presented, which is called Timed-PageRank. Timed-PageRank adds the temporal dimension to search to deal with the dynamic nature of the Web and the aging of Web pages.

6.2 PageRank Algorithm

PageRank produces a static ranking of Web pages in the sense that a PageRank value is computed for each page off-line, and the value is not dependent on search queries. In other words, the PageRank computation is purely based on the existing links on the Web and has nothing to do with each query issued by users. Before introducing the PageRank formula, let us first state some main concepts.

In-links of page i: These are the hyperlinks that point to page i from other pages. Usually, hyperlinks from the same site are not considered.

Out-links of page i: These are the hyperlinks that point out to other pages from page i. Usually, links to pages of the same site are not considered.

The following ideas based on rank prestige [9] are used to derive the PageRank algorithm.

1. A hyperlink from a page pointing to another page is an implicit conveyance of authority to the target page. Thus, the more in-links that a page i receives, the more prestige the page i has.
2. Pages that point to page i also have their own prestige scores. A page with a higher prestige score pointing to i is more important than a page with a lower prestige score pointing to i. In other words, a page is important if it is pointed to by other important pages.

According to rank prestige in social networks, the importance of page i (i's PageRank score) is determined by summing up the PageRank scores of all pages that point to i. Because a page may point to many other pages, its prestige score should be shared among all the pages to which it points.

To formulate the above ideas, we treat the Web as a directed graph $G = (V, E)$, where V is the set of vertices or nodes, that is, the set of all pages, and E is the set of directed edges in the graph, that is, hyperlinks. Let the total number of pages on the Web be n (i.e., $n = |V|$). The PageRank score of the page i [denoted by $P(i)$] is defined by:

$$P(i) = \sum_{(j,i)\in E} \frac{P(j)}{O_j} \tag{6.1}$$

where O_j is the number of out-links of page j. Mathematically, we have a system of n linear equations [Equation (6.1)] with n unknowns. We can use a matrix to represent all the equations. As a notational convention, we use bold and italic letters to represent matrices. Let $\boldsymbol{P}$ be an n-dimensional column vector of PageRank values, that is,

$$\boldsymbol{P} = (P(1), P(2), \ldots, P(n))^T$$

Let $\boldsymbol{A}$ be the adjacency matrix of our graph with

$$A_{ij} = \begin{cases} \dfrac{1}{O_i} & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

We can write the system of n equations with

$$\boldsymbol{P} = \boldsymbol{A}^T \boldsymbol{P} \tag{6.3}$$

This is the characteristic equation of the *eigensystem,* where the solution to $\boldsymbol{P}$ is an *eigenvector* with the corresponding *eigenvalue* of 1. Because this is a circular definition, an iterative algorithm is used to solve it. It turns out that if some conditions are satisfied (which will be described shortly), 1 is the largest eigenvalue and the PageRank vector $\boldsymbol{P}$ is the *principal eigenvector*. A well-known mathematical technique called *power iteration* [2] can be used to find $\boldsymbol{P}$.

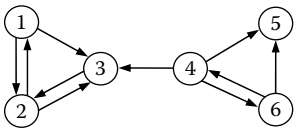


Figure 6.1 An example of a hyperlink graph.

The conditions are that $\boldsymbol{A}$ is a *stochastic matrix* and that it is *irreducible* and *aperiodic*. However, the Web graph does not meet these conditions. In fact, Equation (6.3) can also be derived based on the *Markov chain*. Then some theoretical results from Markov chains can be applied [8], which is where the above three conditions come from.

In the Markov chain model, each Web page or node in the Web graph is regarded as a state. A *hyperlink* is a transition, which leads from one state to another state with a probability. Thus, this framework models Web surfing as a stochastic process. It models a *Web surfer* randomly surfing the Web as a state transition in the Markov chain.

Now let us look at the Web graph and see why all three conditions are not satisfied. First of all, $\boldsymbol{A}$ is not a *stochastic* (*transition*) *matrix*. A stochastic matrix is the transition matrix for a finite Markov chain whose entries in each row are nonnegative real numbers and sum to 1. This requires that every Web page must have at least one out-link. This is not true on the Web because many pages have no out-links, which are reflected in transition matrix $\boldsymbol{A}$ by some rows of complete 0's. Such pages are called *dangling pages* (nodes).

Example 6.2.1 Figure 6.1 shows an example of a hyperlink graph.

If we assume that the Web surfer will click the hyperlinks in a page uniformly at random, we have the following transition probability matrix:

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix} \tag{6.4}$$

For example, $A_{12} = A_{13} = 1/2$ because node 1 has two out-links. We can see that $\boldsymbol{A}$ is not a stochastic matrix because the fifth row is all 0's, that is, page 5 is a dangling page.

We can fix this problem by adding a complete set of outgoing links from each such page i to all the pages on the Web. Thus, the transition probability of going from i to every page is $1/n$, assuming a uniform probability distribution. That is, we replace each row containing all 0's with $\boldsymbol{e}/n$, where $\boldsymbol{e}$ is n-dimensional vector of

all 1's, giving us the following matrix:

$$\bar{\boldsymbol{A}} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix} \tag{6.5}$$

Below, we assume that the above is done to make $\boldsymbol{A}$ a stochastic matrix.

Second, $\boldsymbol{A}$ is not *irreducible,* which means that the Web graph G is not strongly connected.

Definition of strongly connected graphs: A directed graph $G = (V, E)$ is *strongly connected* if and only if, for each pair of nodes $u, v \in V$, there is a path from u to v.

The general Web graph represented by $\boldsymbol{A}$ is not irreducible because for some pairs of nodes u and v, there is no path from u to v. For example, in Figure 6.1, there is no directed path from node 3 to node 4. The adjustment in Equation (6.5) is not enough to ensure irreducibility. This problem and the next problem can be dealt with using a single strategy (described below).

Finally, $\boldsymbol{A}$ is not *aperiodic*. A state i in a Markov chain being periodic means that there exists a directed cycle that the chain has to traverse.

Definition of aperiodic graphs: A state i is *periodic* with period $k > 1$ if k is the smallest number such that all paths leading from state i back to state i have a length that is a multiple of k. If a state is not periodic (i.e., $k = 1$), it is aperiodic. A Markov chain is aperiodic if all states are aperiodic.

Example 6.2.2 Figure 6.2 shows a periodic Markov chain with $k = 3$. The transition matrix is given on the left. Each state in this chain has a period of 3. For example, if we start from state 1, the only path to come back to state 1 is 1-2-3-1 for some number of times, say h. Thus, any return to state 1 will take $3h$ transitions. In the Web, there could be many such cases.

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

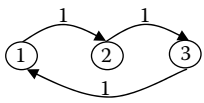


Figure 6.2 A periodic Markov chain with $k = 3$.

It is easy to deal with the above two problems with a single strategy.

- We add a link from each page to every page and give each link a small transition probability controlled by a parameter d.

The augmented transition matrix clearly becomes irreducible and also aperiodic. After this augmentation, we obtain an improved PageRank model:

$$\boldsymbol{P} = \left((1-d)\frac{\boldsymbol{E}}{n} + d\boldsymbol{A}^T\right)\boldsymbol{P} \tag{6.6}$$

where $\boldsymbol{E}$ is $\boldsymbol{e}\boldsymbol{e}^T$ ($\boldsymbol{e}$ is a column vector of all 1's) and thus $\boldsymbol{E}$ is an $n \times n$ square matrix of all 1's. n is the total number of nodes in the Web graph and $1/n$ is the probability of jumping to a random page. Note that Equation (6.6) assumes that $\boldsymbol{A}$ has already been made a stochastic matrix. After scaling, we obtain

$$\boldsymbol{P} = (1-d)\boldsymbol{e} + d\boldsymbol{A}^T\boldsymbol{P} \tag{6.7}$$

This gives us the PageRank formula for each page i:

$$P(i) = (1-d) + d\sum_{j=1}^{n} A_{ji}P(j) \tag{6.8}$$

which is equivalent to the formula given in the original PageRank papers [1, 7]:

$$P(i) = (1-d) + d\sum_{(j,i)\in E}\frac{P(j)}{O_j} \tag{6.9}$$

The parameter d, called the *damping factor,* can be set to a value between 0 and 1. $d = 0.85$ is used in [1, 7].

The computation of PageRank values of the Web pages can be done using the power iteration method [2], which produces the principal eigenvector with an eigenvalue of 1. The algorithm is quite simple (see Figure 6.3). One can start with any initial assignments of PageRank values. The iteration ends when the PageRank values do not change much or converge. In Figure 6.3, the iteration ends after the 1-norm of the residual vector is less than a prespecified threshold ε.

In Web search, we are only interested in the ranking of the pages. Thus, the actual convergence may not be necessary and fewer iterations are needed. In [1], it is reported that on a database of 322 million links the algorithm converges to an acceptable tolerance in roughly 52 iterations.

Since PageRank was presented in [1], researchers have proposed many enhancements to the model, alternative models, and improvements for its computation. The books by Liu [5] and by Langville and Meyer [4] contain in-depth analyses of PageRank and several other link-based algorithms, including HIT [3], which is another well-known algorithm.

PageRank-Iterate(G)

$\boldsymbol{P}_0 \leftarrow \boldsymbol{e}/n$

$k \leftarrow 1$

repeat

$\boldsymbol{P}_k \leftarrow (1-d)\boldsymbol{e} + d\boldsymbol{A}^T\boldsymbol{P}_{k-1}$;

$k \leftarrow k+1$;

until $||\boldsymbol{P}_k - \boldsymbol{P}_{k-1}||_1 < \varepsilon$

return $\boldsymbol{P}_k$

Figure 6.3 The power iteration method for PageRank.

6.3 An Extension: Timed-PageRank

One aspect that is not considered by PageRank is the timeliness of search results. The Web is a dynamic environment. It changes constantly. Quality pages in the past may not be quality pages now or in the future. The temporal aspect of search is important as users are often interested in the latest information. Apart from well-established facts and classics which do not change much over time, most contents on the Web change constantly. New pages or contents are added and outdated contents and pages are deleted. However, in practice many outdated pages and links are not deleted. This causes problems for search engines because such outdated pages can still be ranked high due to the fact that they have existed on the Web for a long time and have accumulated a large number of in-links. High-quality new pages with the most up-to-date information will be ranked low because they have few or no in-links, making it difficult for users to find the latest information on the Web.

An algorithm called Timed-PageRank given in [6, 10] adds the temporal dimension to PageRank. The idea of Timed-PageRank is simple. It still follows the random surfer and Markov chain model in PageRank. However, instead of using a constant damping factor d, Timed-PageRank uses a function of time $f(t)$ $(0 \leq f(t) \leq 1)$ to "penalize" old links and pages, where t is the difference between the current time and the time when the page was last updated. $f(t)$ returns a probability that the Web surfer will follow an actual link on the page. $1 - f(t)$ returns the probability that the surfer will jump to a random page. Thus, at a particular page i, the Web surfer has two options:

1. With probability $f(t_i)$, he/she randomly chooses an outgoing link to follow.
2. With probability $1 - f(t_i)$, he/she jumps to a random page without a link.

The intuition here is that if the page was last updated (or created) a long time ago, the pages that it points to are even older and are probably out of date. Then the $1 - f(t)$ value for such a page should be large, which means that the surfer will have a high

probability of jumping to a random page. If a page is new, then its $1 - f(t)$ value should be small, which means that the surfer will have a high probability to follow an out-link of the page and a small probability of jumping to a random page. For a complete new page in a Web site, which does not have any in-links at all, the method given uses the average Timed-PageRank value of the past pages in the Web site. This is reasonable because a quality site in the past usually publishes quality new pages. The Timed-PageRank algorithm has been evaluated based on research publication search and has given promising results. Interested readers, please refer to [6] for additional details.

6.4 Summary

Link-based ranking for search has been instrumental for Web search. PageRank is the best known algorithm for the purpose. It is practically very effective and also well-founded theoretically. This chapter provides introductory material only; further details can be found in [1, 4, 5, 7]. An extension to the PageRank algorithm is also briefly discussed, which adds the temporal dimension to search. Finally, we should note that link-based ranking is not the only strategy used in a search engine. Many other information retrieval and data mining methods and heuristics based on the page content and user clicks are also employed.

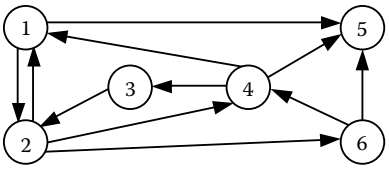
6.5 Exercises

1. Given $\boldsymbol{A}$ below, obtain $\boldsymbol{P}$ by solving Equation (6.7) directly.

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 1/4 & 0 & 1/4 & 1/4 \\ 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}$$

2. Given $\boldsymbol{A}$ as in problem 1, use the power iteration method to show the first 10 iterations of $\boldsymbol{P}$.
3. Calculate the squared error on each iteration in problem 2 where the squared error is defined to be the sum of the squared error on each entry of $\boldsymbol{P}$.
4. Plot a curve on the squared errors derived from problem 3 using the number of iterations as the X axis and the squared error as the Y axis. Does the squared error gradually decrease? After how many iterations do the ranking of the pages stabilize?

5. Given the graph G below, what is $\boldsymbol{A}$?



6. For the graph G given in problem 5, what is $\boldsymbol{P}$ after seven iterations based on the power iteration method?
7. Pick a URL, and construct a Web graph containing Web pages within three hops from the starting URL.
8. For the graph derived in problem 7, what is $\boldsymbol{A}$?
9. For the graph derived in problem 7, use the power iteration method to give the first seven iterations of $\boldsymbol{P}$.

References

[1] S. Brin, and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30, 1998.

[2] G. H. Golub, and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1983.

[3] J. Kleinberg. Authoritative sources in a hyperlinked environment. *ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[4] A. N. Langville, and C. D. Meyer. *Google's PageRank and Beyond: The Science of Search Engine Rankings*. Princeton University Press, 2006.

[5] B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data*. Springer, 2007.

[6] X. Li, B. Liu, and P. S. Yu. *Time Sensitive Ranking with Application to Publication Search*. Conference on Data Mining 2008.

[7] L. Page, S. Brin, R. Motwami, and T. Winograd. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report 1999–0120, Computer Science Department, Stanford University, 1999.

[8] W. Steward. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.

[9] S. Wasserman, and K. Raust. *Social Network Analysis*. Cambridge University Press, 1994.

[10] P. S. Yu, X. Li, and B. Liu. Adding the Temporal Dimension to Search—A Case Study in Publication Search. *WI-2005*.