Calculating Web Page Authority Using the PageRank Algorithm

Jacob Miles Prystowsky and Levi Gill

Math 45, Fall 2005

1 Introduction

1.1 Abstract

In this document, we examine how the Google Internet search engine uses the PageRank algorithm to assign quantitatively authority values to web pages in a network. We examine the PageRank algorithm as a stochastic process, an iterative summation, the solution to an eigenvector equation, and an iterative power method. Of particular focus is the relationship between the eigenvalues and eigenvectors of the Google matrix to PageRank values, the guarantee that there is a unique vector to which the PageRank algorithm converges, how the different methods of computing PageRank relate to one another, and issues of efficiency in computing PageRank with large sets of data. Finally, we will demonstrate examples of the PageRank algorithm using both contrived and real-world examples.

1.2 Motivation for PageRank

When using search engines to look for information on the Internet, we often find that, while much of what we find is useful, a great deal of it is also irrelevant to our queries, making it difficult to separate the useful information from the useless. Furthermore, search results also sometimes include irrelevant commercial web pages masquerading as relevant sources of information, further complicating matters.

Thus, search engines are charged with a major task: to decide quantitatively what content is relevant and authoritative, thus making it easier to find useful information. Google, one of the most popular search engines at the time of writing, uses a well-known algorithm, PageRank, whose job is to assign authority rankings to every page in Google's World Wide Web index. This document discusses PageRank.

2 Contrived Example

Consider the illustration in **figure 1** of a simple network of web pages, which we will use throughout this document to illustrate PageRank.

Mathematically, we can think of this network as a **graph,** where each page is a **vertex**, and a link from one page to another is a **graph edge** [1]. Furthermore, we can represent this network using an $n \times n$ **adjacency matrix** A, where $A_{ij} = 1$ if there is a link from vertex i to vertex j, and 0 otherwise [2]. In the language of PageRank, vertices are **nodes** (web pages), the edges from a node are **forward links,** and the edges into a node are **backlinks** [3]. Thus, the 5×5 adjacency matrix for the graph in **figure 1** is

Title Page
Previous Page
Next Page
Back
Close
Quit

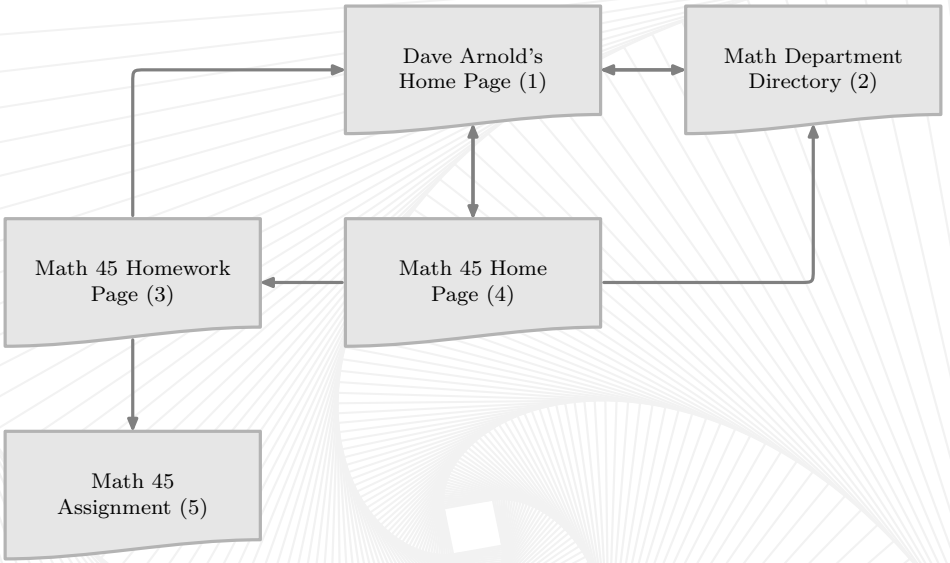

Figure 1 A simple network

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{1}$$

Notice that there are no forward links from node five. Pages with no outlinks are called **dangling nodes,** and require special treatment [4]. Furthermore, notice that the diagonal entries are all zero. We assume that links from a page to itself are ignored when constructing an adjacency matrix.

3 Informal Description

An informal description of PageRank is given by an intuitive definition of how one might measure the importance of a web page on the Internet. A page p's PageRank is based upon how many other pages link

to p. Specifically, the PageRank of p is the sum of the PageRanks of each page q_i that links to p divided by the number of pages to which q_i links [5]. This intuitive definition makes sense: if p is linked to only by pages with low PageRanks, it is probably not authoritative. Further, if p is linked to by a page v with a high PageRank, but v links to many other pages, p should not receive the full weight of v's PageRank.

4 Formal Definition

Now that we have an idea of how PageRank works, we will formally define the algorithm. Drawing on [3], we define PageRank as follows:

- Let u be a web page.
- Let F_u be the set of forward links from u.
- Let B_u be the set of backlinks into u.
- Let $N_u = |F_u|$ be the number of forward links from u.
- Let c be a normalization factor so that "the total rank of all web pages is constant" [3].
- Let $E(u)$ be "some vector over the Web pages that corresponds to a source of rank" [3].

Then, the PageRank of page u is given by

$$R'(u) = c \sum_{v \in B_u} \frac{R'(v)}{N_v} + cE(u), \tag{2}$$

which is iterated until the values have converged sufficiently [4].

4.1 Random Surfer

PageRank also has a second definition based upon the model of a random web surfer navigating the Internet. In short, the model states that PageRank models the behavior of someone who "keeps clicking on successive links at random" [3]. However, occasionally the surfer " 'gets bored' and jumps to a random page chosen based on the distribution in E." [3].

5 The Google Matrix

Although PageRank can be described using **equation 2**, the summation method is neither the most interesting nor the most illustrative of the algorithm's properties. The preferable method is to rewrite

Title Page
Previous Page
Next Page
Back
Close
Quit

the sum as a matrix multiplication. If we let π^T be the $1 \times n$ PageRank row vector, and H the $n \times n$ **row-normalized** adjacency matrix, then we can describe the PageRank vector at the kth iteration as

$$\pi^{(k+1)T} = \pi^{(k)T} H, \tag{3}$$

so that successive iterations $\pi^{(k+1)T}$ converge to the PageRank vector π^T [4].

5.1 Row-Normality

Although we have mentioned it, we have not yet formed the matrix H. Thus, the first step is to let

$$H_i = \frac{A_i}{\sum_{k=1}^{n} A_{ik}},$$

so that each row A_i of A is divided by its sum (the reasons for which will become clear later). Using A from **equation 1**, we have

$$H = \begin{pmatrix} 0 & 1/2 & 0 & 1/2 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{4}$$

5.2 Stochasticity

Using the matrix H is insufficient for the PageRank algorithm, however, because the iteration using H alone might not converge properly — "it can cycle or the limit may be dependent on the starting vector" [4]. Part of the explanation for this is that the matrix H is not yet necessarily **stochastic** [4]. A matrix is stochastic when it is a "matrix of transition probabilities for a **Markov chain,**"[1] with the property that "all elements are non-negative and all its row sums are unity" (one) [7]. Thus, to ensure that H is stochastic, we must ensure that every row sums to one. But clearly, the sum of the last row of the H in **equation 4** is zero. Therefore, we define the stochastic S as

$$S = H + \frac{\mathbf{a}\mathbf{e}^T}{n},$$

[1] A Markov chain is "[a] collection of random variables" whose future values are independent of their past values [6].

where $\mathbf{a}$ is a column vector such that $\mathbf{a}_i = 1$ if $\sum_{k=1}^{n} H_{ik} = 0$ (i.e., page i is a dangling node) and 0 otherwise, and $\mathbf{e}$ is a column vector of ones [4]. Our S, using H from **equation 4**, is

$$S = \begin{pmatrix} 0 & 1/2 & 0 & 1/2 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 1/2 \\ 1/3 & 1/3 & 1/3 & 0 & 0 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{pmatrix}. \tag{5}$$

5.3 Irreducibility

However, we are still not completely done, because there is no guarantee that S has a unique **stationary distribution vector** (i.e., there might not be a single "correct" PageRank vector) [4]. For us to guarantee that there is a single stationary distribution vector π^T to which we can converge, we must ensure that S is **irreducible** as well as stochastic [4]. A matrix is irreducible if and only if its graph is **strongly connected** [8](defined shortly), so we define the irreducible row-stochastic matrix G as

$$G = \alpha S + (1 - \alpha)E, \tag{6}$$

"where $0 \leq \alpha \leq 1$ and $E = \mathbf{e}\mathbf{e}^T/n$" [4]. G is the Google matrix, and we can use it to define

$$\pi^{(k+1)T} = \pi^{(k)T} G \tag{7}$$

as the new iterative method for PageRank to replace **equation 3**.

Google uses the value of 0.85 for α [4] (the reasons for this are discussed later), and we will use this same value. Thus, using S from **equation 5**, our G is

$$G = \begin{pmatrix} 3/100 & 91/200 & 3/100 & 91/200 & 3/100 \\ 22/25 & 3/100 & 3/100 & 3/100 & 3/100 \\ 91/200 & 3/100 & 3/100 & 3/100 & 91/200 \\ 47/150 & 47/150 & 47/150 & 3/100 & 3/100 \\ 1/5 & 1/5 & 1/5 & 1/5 & 1/5 \end{pmatrix}.$$

Consider **figure 1** again. We say that the graph $G(A)$ of a matrix A is "the directed graph on n nodes $\{N_1, N_2, \ldots, N_n\}$ in which there is a directed edge leading *from* N_i *to* N_j if and only if $a_{ij} \neq 0$" [8]. We say that a graph is strongly connected when "for each pair of nodes (N_i, N_k) there is a sequence of directed edges leading from N_i to N_k" [8]. Finally, we say that a matrix is irreducible if and only if its graph is

strongly connected [8]. Looking at it from another angle, we can say that a matrix is irreducible when there exists no "permutation matrix P such that

$$P^T AP = \begin{pmatrix} X & Y \\ 0 & Z \end{pmatrix},$$

where X and Z are both square" [8]. Since the matrix G is completely dense, and the graph of G now has every node connected to every other node, the graph of G is strongly connected and G is irreducible. Notice, also, that G at this point will be completely positive, by its definition.

6 Proof of Uniqueness and Convergence

6.1 The Spectral Radius and Perron Root

We will now prove conclusively that, by its construction, the Google matrix will have a unique PageRank vector to which we can converge. First, we define the **spectrum** (set of distinct eigenvalues) of A as $\sigma(A)$, and the **spectral radius** of a square matrix A as $\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda|$ [8]. We now examine the **Perron-Frobenius Theorem,** which applies to nonnegative, irreducible square matrices A. Borrowing from [8], the Perron-Frobenius Theorem tells us that:

- $r = \rho(A)$ is the **Perron root**, $r > 0$ and $r \in \sigma(A)$.
- The algebraic multiplicity of r is 1.
- There exists a positive $\mathbf{x} > 0$ such that $A\mathbf{x} = r\mathbf{x}$.
- There is a unique **Perron vector** such that:
  - $A\mathbf{p} = r\mathbf{p}$.
  - $\mathbf{p} > 0$.
  - $\|\mathbf{p}\|_1 = 1$ (where $\|\mathbf{p}\|_1$ represents the L_1-Norm of the vector $\mathbf{p}$, such that $\|\mathbf{x}\|_1 = \sum_{r=1}^{n} |\mathbf{x}_r|$ [9]).
  - There are no other nonnegative eigenvectors for A except for positive multiples of $\mathbf{p}$.

Furthermore, we can apply **Perron's Theorem** to A, which requires that A be positive with $r = \rho(A)$ [8]. Perron's Theorem gives us that "r is the only eigenvalue on the spectral circle of A" [8].

Notice the crucial facts that $\rho(G)$ is the only eigenvalue on the spectral circle of G and that the algebraic multiplicity of $\rho(G)$ is one, by the Perron and Perron-Frobenius theories. We have established that $r = \rho(G)$ is the dominant eigenvalue of G.

6.2 Primitivity, Stationary Distribution, and Eigenvectors

Now that we have established these facts, we conclude that G is a **primitive matrix**, defined as a "nonnegative irreducible matrix A having only one eigenvalue, $r = \rho(A)$, on its spectral circle" [8].

If we know that P is primitive (as G is), we know that $\lim_{k\to\infty} P^k$ exists [8]. Specifically, where $\mathbf{p}$ and $\mathbf{q}$ are, respectively, the Perron vectors for A and A^T, $\lim_{k\to\infty} (A/r)^k = (\mathbf{p}\mathbf{q}^T)/(\mathbf{q}^T\mathbf{p})$ [8].

Now, $\mathbf{e}/n$ is a right-hand Perron vector for P (since P is row-stochastic, and $\mathbf{e}/n$ effectively takes the sum of each row), so if π is the Perron vector for P^T, then, from [8],

$$\begin{aligned}\lim_{k\to\infty} P^k &= \frac{(\mathbf{e}/n)\pi^T}{\pi^T(\mathbf{e}/n)} \\ &= \frac{\mathbf{e}\pi^T}{\pi^T\mathbf{e}} \\ &= \mathbf{e}\pi^T > 0.\end{aligned}$$

This leads us directly to the fact that π^T must, then, be the left Perron vector for P. The reason for this is that $P^T\pi = \pi$, i.e., π is a right Perron vector for P^T. However, if we transpose both sides of the equation, we immediately see that $\pi^T P = \pi^T$ directly follows. Thus, π^T is the left Perron vector for P, or G in our case.

Continuing, we can regard G as a "transition probability matrix for an irreducible Markov chain" (where such a matrix "P is an $n \times n$ irreducible stochastic matrix," with π^T as P's left-hand Perron vector: this is equivalent to our G) [8]. Drawing on [8], we make the following statements regarding P and every initial distribution vector $\mathbf{p}^{(0)T}$, where the sum of the entries of $\mathbf{p}^{(0)T}$ is one:

- $\mathbf{p}^{(k)T} = \mathbf{p}^{(0)T}P^k$.
- For a primitive P, $\lim_{k\to\infty} P^k = \mathbf{e}\pi^T$ and $\lim_{k\to\infty} \mathbf{p}^{(k)T} = \pi^T$.
- π^T is the stationary distribution vector, and is the unique vector where $\pi^T P = \pi^T$.

These results validate what we proved above. We know that π^T is a left eigenvector, and the left Perron vector, for G, and that its eigenvalue is $r = \rho(G) = 1$. Furthermore, we know that π^T is the stationary distribution vector of G. We also know that the PageRank algorithm can be iterated using **equation 7** until the values converge sufficiently to the PageRank vector, i.e., until $\pi^{(k+1)T} \approx \pi^{(k)T}G$. Thus, we conclude that the PageRank vector, the left Perron vector, and the stationary distribution vector of G are

one and the same, and so finding the PageRank vector of G is equivalent to finding the dominant right eigenvector of G^T, or the dominant left eigenvector (left Perron vector) of G.

6.3 Perron Vector of G

For illustration purposes, we use MATLAB and find that the left Perron vector of G from **equation 6** is

$$\mathbf{p} \approx \begin{pmatrix} 0.35961320922905 \\ 0.25380393805204 \\ 0.10096832412970 \\ 0.19776930237822 \\ 0.08784522621099 \end{pmatrix}. \tag{8}$$

7 The Power Method

When dealing with data sets as large as Google uses (more than eight billion web pages [10]), it is unrealistic to form a matrix G and find its dominant left eigenvector. It is more efficient to compute the PageRank vector using the power method, where we iterate using the sparse matrix H by rewriting **equation 7**. In other words, from [4],

$$\begin{aligned}
\pi^{(k)T} &= \pi^{(k-1)T} G \\
&= \pi^{(k-1)T} \left(\alpha S + (1-\alpha)\mathbf{e}\frac{\mathbf{e}^T}{n} \right) \\
&= \alpha \pi^{(k-1)T} S + (1-\alpha)\pi^{(k-1)T}\mathbf{e}\frac{\mathbf{e}^T}{n} \\
&= \alpha \pi^{(k-1)T} S + (1-\alpha)\frac{\mathbf{e}^T}{n} \\
&= \alpha \pi^{(k-1)T} \left(H + \mathbf{a}\frac{\mathbf{e}^T}{n} \right) + (1-\alpha)\frac{\mathbf{e}^T}{n} \\
&= \alpha \pi^{(k-1)T} H + (\alpha \pi^{(k-1)T}\mathbf{a} + (1-\alpha))\frac{\mathbf{e}^T}{n}.
\end{aligned}$$

7.1 Convergence Rate

One of the benefits of using the power method to compute the PageRank vector is the speed with which it converges. Specifically, the power method on G "converges at the rate at which α^k goes to zero" [11]. This gives us the ability to estimate the number of iterations required to reach a tolerance level τ ("measured by the residual," $\pi^{(k+1)T} - \pi^{(k)T}$), where the number of iterations is approximately $\log_{10} \tau / \log_{10} \alpha$ [11]. The founders of Google, Lawrence Page and Sergey Brin, use $\alpha = 0.85$ and find success with only 50 to 100 power iterations [11].

The choice of 0.85 for α is not arbitrary. While it is not in the purview of this document, Haveliwala and Kamvar proved in [12] "that the subdominant eigenvalue of G satisfies $|\lambda_2| \leq \alpha$" [4]. The implications of this are significant — using the power method, the convergence rate of the power method is $|\lambda_2|/|\lambda_1|$ [12]. But, since we have already shown that $\lambda_1 = 1$, the convergence rate of PageRank is thus λ_2, ensuring that the power method will quickly converge to π^T, "even as the web scales" [12].

8 PageRank Meaning

Now that we have shown how the PageRank vector can be computed, it is time to analyze its meaning. First, the PageRank of page i is given by the ith element of π^T, π_i [11]. However, since π^T is a Perron vector, and so $\|\pi^T\|_1 = 1$, the PageRank vector is also a probability vector [11], reflecting the likelihood of ending up at page i.

In the case of the $\mathbf{p}$ from **equation 8**, we can see that the ordering relevance is "Dave Arnold's Home Page," "Math Department Directory," "Math 45 Home Page," "Math 45 Homework Page," and "Math 45 Assignment," which one might guess from examining **figure 1**. Furthermore, $\mathbf{p}$ suggests that, starting anywhere within **figure 1** and transitioning according to the "random surfer" model, the likelihood of ending up at any of the pages is, respectively, about 36%, 25%, 10%, 20%, and 9%.

9 Real-World Example

Finally, we will briefly examine a real-world use of PageRank. On December 10th, 2005, we analyzed the link structure of `http://online.redwoods.edu`, the College of the Redwoods Mathematics and Science server. After removing links outside of the subdomain, a 2698×2698 adjacency matrix remained. Due to the size of this matrix, we will not be including it in this document.

We used $\alpha = 0.85$ with 100 iterations of the power method. While the PageRank vector is too large to include here, we include the plot in **figure 2** of that vector's elements, where each point (i, π_i) represents the PageRank of page i.

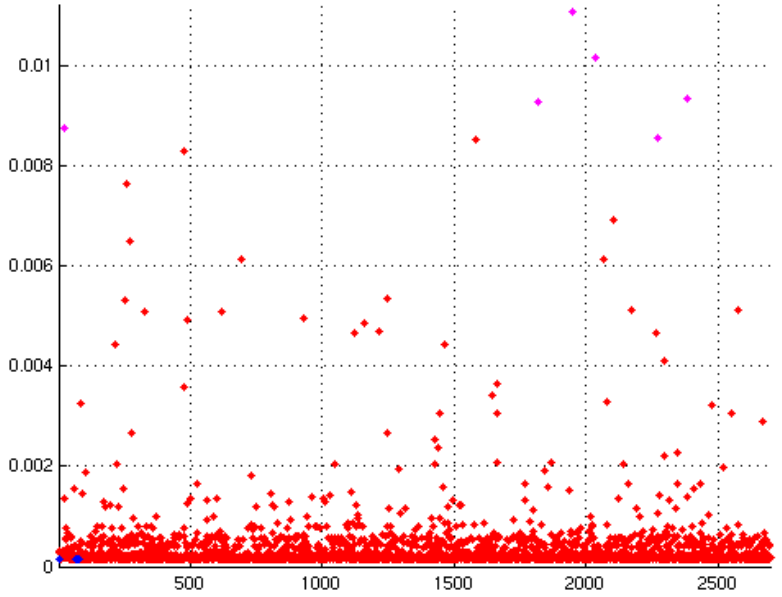

Figure 2 PageRank Vector for `http://online.redwoods.edu`

10 Acknowledgments

Neither this document nor the learning and understanding behind it would have been possible without the help and encouragement from two extraordinary teachers and mentors.

First, Professor David Arnold, who not only always goes the extra mile in order to help his students learn and understand even the most difficult material, but also inspires his students to do their greatest work.

Second, Dr. Bruce Wagner, who has generously provided us with suggestions, resources, and, more importantly, his time and knowledge, all of which have been instrumental in helping us to understand the material covered herein.

To both of them, we offer our sincerest thanks.

References

[1] Eric W. Weisstein, *Graph*, From MathWorld — A Wolfram Web Resource. Last checked December 11th, 2005. *http://mathworld.wolfram.com/Graph.html*

[2] Eric W. Weisstein, *Adjacency Matrix*, From MathWorld — A Wolfram Web Resource. Last checked December 11th, 2005. *http://mathworld.wolfram.com/AdjacencyMatrix.html*

[3] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, *The PageRank Citation Ranking: Bringing Order to the Web* (1998). *http://dbpubs.stanford.edu:8090/pub/1999-66*

[4] Amy N. Langville and Carl D. Meyer, *The Use of the Linear Algebra by Web Search Engines* (2004). *http://meyer.math.ncsu.edu/Meyer/PS_Files/IMAGE.pdf*

[5] Desmond J. Higham and Alan Taylor, *The Sleekest Link Algorithm* (2003). *http://www.ece.northwestern.edu/~nocedal/328/report.pdf*

[6] Eric W. Weisstein, *Markov Chain*, From MathWorld — A Wolfram Web Resource. Last checked December 11th, 2005. *http://mathworld.wolfram.com/MarkovChain.html*

[7] David Nelson, editor , *The Penguin Dictionary of Mathematics* (Penguin Books Ltd, London, 2003), Third ed.

[8] Carl D. Meyer, *Matrix Analysis and Applied Linear Algebra*, p. 490–693. (The Society for Industrial and Applied Mathematics, Philadelphia, 2000).

[9] Eric W. Weisstein, *L1-Norm*, From MathWorld — A Wolfram Web Resource. Last checked December 11th, 2005. *http://mathworld.wolfram.com/L1-Norm.html*

[10] Bill Coughran, *Google's index nearly doubles*, Google Inc. Last checked December 10th, 2005. *http://googleblog.blogspot.com/2004/11/googles-index-nearly-doubles.html*

[11] Amy N. Langville and Carl D. Meyer, *Fiddling with PageRank* (2003). *http://meyer.math.ncsu.edu/Meyer/PS_Files/FiddlingPageRank.pdf*

[12] Taher H. Haveliwala and Sepandar D. Kamvar, *The Second Eigenvalue of the Google Matrix*, Stanford University Technical Report (2003). *http://www.stanford.edu/~sdkamvar/papers/secondeigenvalue.pdf*