

Fiddling with PageRank

Amy N. Langville[†] and Carl D. Meyer*

August 15, 2003

Abstract

This paper deals with the various changes that can be made to the basic PageRank model. We document the recent findings and add a few new contributions. These contributions concern (1) the sensitivity of the PageRank vector, (2) another method of forcing the Markov chain to be irreducible, and (3) a proof of the full spectrum of the PageRank matrix.

Key words: Markov chains, power method, convergence, stationary vector, PageRank

[†] Department of Mathematics,
N. Carolina State University,
Raleigh, NC 27695-8205, USA
anlangvi@unity.ncsu.edu
Phone: (919) 515-8146,
Fax: (919) 513-3798

* Department of Mathematics,
Center for Research in Scientific Computation,
N. Carolina State University, Raleigh, N.C. 27695-8205, USA
meyer@math.ncsu.edu
Phone: (919) 515-2384,
Fax: (919) 515-3798

Research supported in part by NSF CCR-ITR-0113121 and NSF DMS 9714811.

1 Introduction

Many of today's search engines use a two-step process to retrieve pages related to a user's query. In the first step, traditional text processing is done to find all documents using the query terms, or related to the query terms by semantic meaning. This can be done by a lookup into an inverted file, with a vector space method, or with a query expander that uses a thesaurus. With the massive size of the Web, this first step can result in thousands of retrieved pages related to the query. To make this list manageable, many search engines sort this list by some ranking criterion. One popular way to create this ranking is

to exploit the additional information inherent in the Web due to its hyperlinking structure. Thus, link analysis has become the means to ranking. One successful and well-publicized link-based ranking system is PageRank, the ranking system used by the Google search engine. This paper begins with a review of the most basic PageRank model for determining the importance of a webpage. This basic model, so simple and so elegant, works well, but part of the model’s beauty and attraction lies in its seemingly endless capacity for “tinkering”. Some such tinkering have been proposed and tested. In this paper, we explore these previously suggested tinkering to the basic PageRank model and add a few more suggestions and connections of our own. For example, why has the PageRank convex combination scaling parameter traditionally been set to .85? One answer, presented in section 3.1, concerns convergence to the solution. However, we provide another answer to this question in section 4.4 by presenting the condition number of the problem. Another area of fiddling is the uniform matrix \mathbf{E} added to the hyperlinking Markov matrix \mathbf{P} . What other alternatives to this uniform matrix exist? In section 4.5, we present the common answer, followed by an analysis of our alternative answer. The numerous alterations to the basic PageRank model presented in this paper give an appreciation of the model’s beauty and usefulness, and hopefully, will inspire future and greater improvements.

2 The Basic PageRank model

The original Brin and Page model for PageRank uses the hyperlink structure of the Web to build a stochastic irreducible Markov chain with transition probability matrix \mathbf{P} . The irreducibility of the chain guarantees that the long-run stationary vector $\boldsymbol{\pi}^T$, known as the PageRank vector, exists. It is well-known that the power method applied to a stochastic irreducible matrix \mathbf{P} will converge to this stationary vector. Further, the convergence rate of the power method is determined by the magnitude of the subdominant eigenvalue of \mathbf{P} [12].

2.1 The Markov model of the Web

We begin by showing how Brin and Page, the founders of the PageRank model, force the transition probability matrix \mathbf{P} , which is built from the hyperlink structure of the Web, to be stochastic and irreducible. Consider the hyperlink structure of the Web as a directed graph. The nodes of this digraph represent webpages and the directed arcs represent hyperlinks. For example, consider the small document collection consisting of 6 webpages linked as in Figure 1.

The Markov model represents this graph with a square transition probability matrix \mathbf{P} whose element p_{ij} is the probability of moving from state i (page i) to state j (page j) in one time step. For example, assume that, starting from any node (webpage), it is equally likely to follow any of the outgoing links to arrive at another node. Thus,

$$\mathbf{P} = \begin{pmatrix} 0 & .5 & .5 & 0 & 0 & 0 \\ .5 & 0 & .5 & 0 & 0 & 0 \\ 0 & .5 & 0 & .5 & 0 & 0 \\ 0 & 0 & 0 & 0 & .5 & .5 \\ 0 & 0 & 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Any suitable probability distribution may be used across the rows. For example, if web usage logs show that a random surfer accessing page 2 is twice as likely to jump to page 1 as he or she is to jump to page 3, then the second row of \mathbf{P} , denoted \mathbf{p}_2^T , becomes

$$\mathbf{p}_2^T = (.6667 \quad 0 \quad .3333 \quad 0 \quad 0 \quad 0).$$

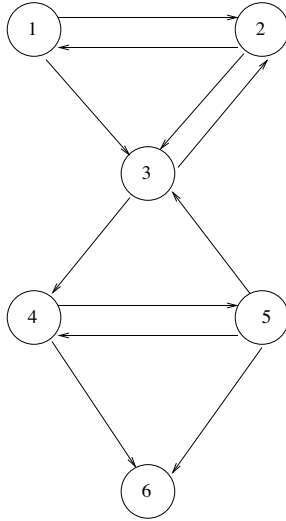


Figure 1: Directed graph representing web of 6 pages

(Columns are similarly denoted. Column i of \mathbf{P} is \mathbf{p}_i .) One problem with solely using the Web's hyperlink structure to build the Markov matrix is apparent. Some rows of the matrix, such as row 6 in our example above, contain all zeros. Thus, \mathbf{P} is not stochastic. This occurs whenever a node contains no outlinks; many such nodes exist on the Web. The remedy is to replace all zero rows, $\mathbf{0}^T$, with $\frac{1}{n}\mathbf{e}^T$, where \mathbf{e}^T is the row vector of all ones and n is the order of \mathbf{P} . The revised transition probability matrix is

$$\mathbf{P} = \begin{pmatrix} 0 & .5 & .5 & 0 & 0 & 0 \\ .5 & 0 & .5 & 0 & 0 & 0 \\ 0 & .5 & 0 & .5 & 0 & 0 \\ 0 & 0 & 0 & 0 & .5 & .5 \\ 0 & 0 & 1/3 & 1/3 & 0 & 1/3 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \end{pmatrix}.$$

However, this adjustment alone is not enough to insure the existence of the stationary vector of the chain, i.e. the PageRank vector. Were the chain irreducible, the PageRank vector is guaranteed to exist. By its very nature, with probability 1, the Web unaltered creates a reducible Markov chain. Thus, one more adjustment, to make \mathbf{P} irreducible, is implemented. The revised stochastic and irreducible matrix $\bar{\mathbf{P}}$ is

$$\bar{\mathbf{P}} = \alpha\mathbf{P} + \frac{(1-\alpha)}{n}\mathbf{E},$$

where $0 \leq \alpha \leq 1$ and \mathbf{E} is the matrix of all 1s. This convex combination of the original matrix \mathbf{P} and a perturbation matrix \mathbf{E} insures that $\bar{\mathbf{P}}$ is both stochastic and irreducible. Every node is now directly connected to every other node, making the chain irreducible by definition. Although the probability of transitioning may be small in some cases, it is always nonzero.

3 Solution Methods for Solving the PageRank Problem

Regardless of the method for filling in the entries of $\bar{\mathbf{P}}$, PageRank is determined by computing the stationary solution $\boldsymbol{\pi}^T$ of the Markov chain. The row vector $\boldsymbol{\pi}^T$ can be found by solving either the

eigenvector problem

$$\boldsymbol{\pi}^T \bar{\mathbf{P}} = \boldsymbol{\pi}^T,$$

or by solving the homogeneous linear system

$$\boldsymbol{\pi}^T (\mathbf{I} - \bar{\mathbf{P}}) = \mathbf{0}^T,$$

where \mathbf{I} is the identity matrix. Both formulations are subject to an additional equation, the normalization equation $\boldsymbol{\pi}^T \mathbf{e} = 1$, where \mathbf{e} is the column vector of all 1's. The normalization equation insures that $\boldsymbol{\pi}^T$ is a probability vector. The i^{th} element of $\boldsymbol{\pi}^T$, π_i , is the PageRank of page i .

3.1 The Power Method

Traditionally, computing the PageRank vector has been viewed as an eigenvector problem, $\boldsymbol{\pi}^T \bar{\mathbf{P}} = \boldsymbol{\pi}^T$, and the notoriously slow power method has been the method of choice. There are several good reasons for using the power method. First, consider iterates of the power method applied to $\bar{\mathbf{P}}$ (a completely dense matrix, were it to be formed explicitly). Note that $\mathbf{E} = \mathbf{e}\mathbf{e}^T$. For any starting vector $\mathbf{x}^{(0)T}$ (generally, $\mathbf{x}^{(0)T} = \frac{1}{n}\mathbf{e}^T$),

$$\begin{aligned} \mathbf{x}^{(k)T} &= \mathbf{x}^{(k-1)T} \bar{\mathbf{P}} = \alpha \mathbf{x}^{(k-1)T} \mathbf{P} + \frac{1-\alpha}{n} \mathbf{x}^{(k-1)T} \mathbf{e}\mathbf{e}^T \\ &= \alpha \mathbf{x}^{(k-1)T} \mathbf{P} + \frac{1-\alpha}{n} \mathbf{e}^T, \end{aligned} \tag{1}$$

since $\mathbf{x}^{(k-1)T}$ is a probability vector, and thus, $\mathbf{x}^{(k-1)T} \mathbf{e} = 1$. Written in this way, it becomes clear that the power method applied to $\bar{\mathbf{P}}$ can be implemented with vector-matrix multiplications on the extremely sparse \mathbf{P} and $\bar{\mathbf{P}}$ is never formed or stored. On the other hand, direct methods on the linear system $\boldsymbol{\pi}^T (\mathbf{I} - \bar{\mathbf{P}}) = \mathbf{0}^T$ require the storage and manipulation of $\mathbf{I} - \bar{\mathbf{P}}$, a practical impossibility for Web-sized document collections, such as the 3.4 billion by 3.4 billion matrix used by Google. Thus, a matrix-free method such as the power method, is required. Fortunately, since \mathbf{P} is sparse, each vector-matrix multiplication required by the power method can be computed in $nnz(\mathbf{P})$ flops, where $nnz(\mathbf{P})$ is the number of nonzeros in \mathbf{P} . Further, at each iteration, the power method only requires the storage of one vector, the next iterate, whereas other accelerated matrix-free methods, such as restarted GMRES or BiCGStab, require storage of at least several vectors, depending on the size of the subspace chosen. Finally, the power method on Brin and Page's $\bar{\mathbf{P}}$ matrix converges at the rate at which α^k goes to zero. See [7] (and the appendix for a shorter alternate proof) that $\lambda_2(\bar{\mathbf{P}}) = \alpha$ for a reducible \mathbf{P} , and hence the rate of convergence for the power method applied to $\bar{\mathbf{P}}$ is governed by α . Brin and Page, the founders of Google, use $\alpha = .85$. Thus, a rough estimate of the number of iterations needed to converge to a tolerance level τ (measured by the residual, $\mathbf{x}^{(k)T} \bar{\mathbf{P}} - \mathbf{x}^{(k)T} = \mathbf{x}^{(k+1)T} - \mathbf{x}^{(k)T}$) is $\frac{\log_{10} \tau}{\log_{10} \alpha}$. For $\tau = 10^{-6}$ and $\alpha = .85$, one can expect roughly $\frac{-6}{\log_{10} .85} \approx 85$ iterations until convergence to the PageRank vector. For $\tau = 10^{-8}$, about 114 iterations and for $\tau = 10^{-10}$, about 142 iterations. Brin and Page report success using only 50 to 100 power iterations, implying that τ could range from 10^{-3} to 10^{-7} .

We conclude this section with a brief discussion of more detailed storage issues for implementation. The decomposition of $\mathbf{P} = \mathbf{D}^{-1} \mathbf{G}$ into the product of the inverse of the diagonal matrix \mathbf{D} holding outdegrees of the nodes and the adjacency matrix \mathbf{G} of 0s and 1s is useful in saving storage and reducing work at each power iteration. $\mathbf{P} = \mathbf{D}^{-1} \mathbf{G}$ can be used to reduce the number of multiplications required in each $\mathbf{x}^T \mathbf{P}$ vector-matrix multiplication needed by the power method. Without the $\mathbf{P} = \mathbf{D}^{-1} \mathbf{G}$ decomposition, this requires $nnz(\mathbf{P})$ multiplications and $nnz(\mathbf{P})$ additions. Using the vector $diag(\mathbf{D}^{-1})$, $\mathbf{x}^T \mathbf{P}$ can be accomplished as $\mathbf{x}^T \mathbf{D}^{-1} \mathbf{G} = (\mathbf{x}^T) .* (diag(\mathbf{D}^{-1})) \mathbf{G}$, where $.*$ represents componentwise multiplication of the elements in the two vectors. The first part, $(\mathbf{x}^T) .* (diag(\mathbf{D}^{-1}))$ requires n multiplications. Since \mathbf{G}

is an adjacency matrix, $(\mathbf{x}^T) \cdot * (\text{diag}(\mathbf{D}^{-1}))\mathbf{G}$ now requires an additional $\text{nnz}(\mathbf{P})$ additions for a total savings of $\text{nnz}(\mathbf{P}) - n$ multiplications. Also, it is likely that Web-sized implementations of the PageRank model store the \mathbf{P} or \mathbf{G} matrix as an adjacency list of the transposed matrix.

3.2 The Linear System Formulation

In this section, we formulate the Google problem as a linear system. The eigenvalue problem $\boldsymbol{\pi}^T(\alpha\mathbf{P} + \frac{(1-\alpha)}{n}\mathbf{e}\mathbf{e}^T) = \boldsymbol{\pi}^T$ can be rewritten, with some algebra as,

$$\boldsymbol{\pi}^T(\mathbf{I} - \alpha\mathbf{P}) = \frac{(1-\alpha)}{n}\mathbf{e}^T. \quad (2)$$

We note some interesting properties of the coefficient matrix in this equation.

Properties of $(\mathbf{I} - \alpha\mathbf{P})$:

1. $(\mathbf{I} - \alpha\mathbf{P})$ is nonsingular.

Proof: To prove nonsingularity we show that $\mathbf{x} \neq 0$ implies that $(\mathbf{I} - \alpha\mathbf{P})\mathbf{x} \neq 0$. Suppose $\mathbf{x} \neq 0$. Then $\|(\mathbf{I} - \alpha\mathbf{P})\mathbf{x}\| = \|\mathbf{x} - \alpha\mathbf{P}\mathbf{x}\| \geq \|\mathbf{x}\| - \alpha\|\mathbf{P}\|\|\mathbf{x}\| \geq (1 - \alpha\|\mathbf{P}\|)\|\mathbf{x}\| = (1 - \alpha)\|\mathbf{x}\| > 0$, since $1 - \alpha > 0$ and $\mathbf{x} \neq 0$ implies $\|\mathbf{x}\| > 0$. Thus, $\mathbf{x} \neq 0$ implies $\|(\mathbf{I} - \alpha\mathbf{P})\mathbf{x}\| \neq 0$, which implies that $(\mathbf{I} - \alpha\mathbf{P})\mathbf{x} \neq 0$. Hence, $(\mathbf{I} - \alpha\mathbf{P})$ is nonsingular. \square

2. $(\mathbf{I} - \alpha\mathbf{P})$ is an \mathbf{M} -matrix.

Proof: Straightforward from the definition of \mathbf{M} -matrix given by Berman and Plemmons [1]. \square

3. The row sums of $(\mathbf{I} - \alpha\mathbf{P})$ are $1 - \alpha$.

Proof: $(\mathbf{I} - \alpha\mathbf{P})\mathbf{e} = (1 - \alpha)\mathbf{e}$. \square

4. $\|\mathbf{I} - \alpha\mathbf{P}\|_\infty = 1 + \alpha$.

Proof: The ∞ -matrix norm is the maximum absolute row sum. If a page i has a positive number of outlinks, then the corresponding diagonal element of $\mathbf{I} - \alpha\mathbf{P}$ is 1. All other off-diagonal elements are negative, but sum to α in absolute value. \square

5. Since $(\mathbf{I} - \alpha\mathbf{P})$ is an \mathbf{M} -matrix, $(\mathbf{I} - \alpha\mathbf{P})^{-1} \geq 0$.

Proof: Again, see Berman and Plemmons [1]. \square

6. The row sums of $(\mathbf{I} - \alpha\mathbf{P})^{-1}$ are $\frac{1}{1-\alpha}$. Therefore, $\|(\mathbf{I} - \alpha\mathbf{P})^{-1}\|_\infty = \frac{1}{1-\alpha}$.

Proof: Using the Neumann series, $(\mathbf{I} - \alpha\mathbf{P})^{-1} = \mathbf{I} + \alpha\mathbf{P} + \alpha^2\mathbf{P}^2 + \dots$. So, $(\mathbf{I} - \alpha\mathbf{P})^{-1}\mathbf{e} = \mathbf{e} + \alpha\mathbf{e} + \alpha^2\mathbf{e} + \dots = \frac{1}{1-\alpha}\mathbf{e}$. \square

7. Thus, the condition number $\kappa_\infty(\mathbf{I} - \alpha\mathbf{P}) = \frac{1+\alpha}{1-\alpha}$.

Proof: By virtue of points 4 and 6 above, $\kappa_\infty(\mathbf{I} - \alpha\mathbf{P}) = \|(\mathbf{I} - \alpha\mathbf{P})\|_\infty \|(\mathbf{I} - \alpha\mathbf{P})^{-1}\|_\infty = \frac{1+\alpha}{1-\alpha}$. \square

Of course, a direct method on the Web-sized sparse \mathbf{P} requires much more calculation than the power method, which recall, requires $\frac{\log_{10}\tau}{\log_{10}\alpha} \text{nnz}(\mathbf{P})$ flops. However, for an Intranet, such as the 517-page Mathworks site with $\text{nnz}(\mathbf{P}) = 13,531$, a direct method with $\alpha = .99$ may require less flops than the power method's estimated $1833 \cdot 13531 = 2.48 \cdot 10^7$ flops with $\alpha = .99$ and $\tau = 10^{-8}$. Also, execution of the Matlab or Netlib built-in linear system solver is faster than the execution of the coded power method for small matrices.

Experiments with a direct method show that, although increasing α negatively affects the convergence, and thus execution time, of the power method, it has no effect on the number of flops, and thus execution time, required by a direct method. Nevertheless, the sensitivity issues discussed later in section 4.4 increase as α increases. In summary, the exclusive use of the power method, as opposed to direct methods, on Web-sized PageRank problems seems justified for several reasons.

4 Tinkering with the basic PageRank model

Varying α , although perhaps the most obvious alteration, is just one way to fiddle with the basic PageRank model presented above. In this paper, we explore many others, devoting a subsection to each. Sections 4.2 and 4.5 discuss ideas for changing the “fudge factor” matrix \mathbf{E} required to make \mathbf{P} irreducible. In section 4.3, the possibility of using different convergence criteria, aside from the residual, are presented. In section 4.4, we compute the condition number for the PageRank problem, showing its implications for sensitivity and stability issues.

4.1 Changing α

One of the most obvious places to begin fiddling with the basic PageRank model is α . Brin and Page, the founders of Google, have reported using $\alpha = .85$. One wonders why this choice for α ? Might a different choice produce a very different ranking of retrieved webpages?

As mentioned in section 3.1, there are good reasons for using $\alpha = .85$, one being the speedy convergence of the power method. With this value for α , we can expect the power method to converge to the PageRank vector in about 114 iterations for a convergence tolerance level of $\tau = 10^{-8}$. Obviously, this choice of α brings faster convergence than higher values of α . Compare with $\alpha = .99$, whereby 1833 iterations are required to achieve a residual less than 10^{-8} . When working with a sparse 3.4 billion by 3.4 billion matrix, each iteration counts; over a few hundred power iterations is more than Google is willing to compute. However, in addition to the computational reasons for choosing $\alpha = .85$, this choice for α also carries some intuitive weight: $\alpha = .85$ implies that roughly 5/6 of the time a Web surfer randomly clicks on hyperlinks (i.e, following the structure of the Web, as captured by the $\alpha\mathbf{P}$ part of the formula), while 1/6 of the time this Web surfer will go to the URL line and type the address of a random new page to “teleport” to (as captured by the $\frac{(1-\alpha)}{n}\mathbf{e}\mathbf{e}^T$ part of the formula). Perhaps this was the original motivation behind Brin and Page’s choice of $\alpha = .85$; it produces accurate model for Web surfing behavior. Whereas $\alpha = .99$, not only slows convergence of the power method, but also places much greater emphasis on the hyperlink structure of the Web and much less on the teleportation tendencies of surfers.

Perhaps the PageRank vector derived from $\alpha = .99$ is vastly different from that obtained using $\alpha = .85$. Perhaps it gives a “truer” PageRanking. Experiments with various α 's show significant variation in rankings produced by different values of α . As expected, the top section of the ranking changes only slightly, yet as we proceed down the ranked list we see more and more variation. Recall that the PageRank algorithm pulls a subset of elements from this ranked list, namely those elements that use the query terms.

This means that the greater variation witnessed toward the latter half of the PageRank vector could lead to substantial variation in the ranking results returned to the user. Which ranking (i.e., which α) is preferred? This is a hard question to answer without doing extensive user verification tests on various datasets and queries. However, there are other ways to answer this question. In terms of convergence time, we've already emphasized the fact that $\alpha = .85$ is preferable, but later, in section 4.4, we present another good reason, in addition to speedy convergence, for choosing α near .85.

4.2 The personalization vector \mathbf{v}^T

One of the first modifications to the basic PageRank model suggested by its founders was a change to the teleportation matrix \mathbf{E} . Rather than using $\frac{1-\alpha}{n}\mathbf{e}\mathbf{e}^T$, they used $\mathbf{e}\mathbf{v}^T$, where $\mathbf{v}^T > 0$ is a probability vector called the personalization vector. Since \mathbf{v}^T is a probability vector with positive elements, every node is still directly connected to every other node, i.e., $\bar{\mathbf{P}}$ is irreducible. Using \mathbf{v}^T in place of $\frac{1}{n}\mathbf{e}^T$ means that the teleportation probabilities are no longer uniformly distributed. Instead, at each webpage, if a surfer teleports, he follows the probability distribution given in \mathbf{v}^T to find the next page. Notice that this slight modification retains the advantageous properties of the power method applied to $\bar{\mathbf{P}}$.

$$\begin{aligned}\mathbf{x}^{(k)T} &= \mathbf{x}^{(k-1)T}\bar{\mathbf{P}} = \alpha\mathbf{x}^{(k-1)T}\mathbf{P} + (1-\alpha)\mathbf{x}^{(k-1)T}\mathbf{e}\mathbf{v}^T \\ &= \alpha\mathbf{x}^{(k-1)T}\mathbf{P} + (1-\alpha)\mathbf{v}^T,\end{aligned}\tag{3}$$

The simple formulas of equations (1) and (3) differ only by the constant vector added at each iteration. Thus, the convergence properties of (3) are identical to those of (1).

Similarly, the linear system formulation of the PageRank problem changes only slightly when the personalization vector is used.

$$\boldsymbol{\pi}^T(\mathbf{I} - \alpha\mathbf{P}) = (1 - \alpha)\mathbf{v}^T.\tag{4}$$

We surmise that the name personalization vector comes from the fact that Google intended to have many different \mathbf{v}^T vectors for the many different classes of surfers. Surfers in one class, if teleporting, may be much more likely to jump to pages about sports, while surfers in another class may be much more likely to jump to pages pertaining to news and current events. Such differing teleportation tendencies can be captured in two different personalization vectors. This seems to have been Google's original intent in introducing the personalization vector [3]. However, it makes the once query-independent, user-independent PageRankings user-dependent and more calculation-laden. Nevertheless, it appears this little personalization vector has had more significant side effects. Google has recently used this personalization vector to control spamming done by the so-called link farms.

Link farms are set up by spammers to fool information retrieval systems into increasing the rank of its client's pages. For example, suppose a business owner has decided to move a portion of his business online. The owner creates a webpage. However, this page rarely gets hits or is returned on Web searches on his product. The owner contacts a search engine optimization company whose sole efforts are aimed at increasing the PageRank (and ranking among other search engines) of its clients' pages. One way a search engine optimizer attempts to do this is with link farms. Knowing that PageRank increases when the number of important inlinks to a client's page increases, optimizers add such links to a client's page. A link farm might have several interconnected nodes about important topics with reasonable PageRanks. These interconnected nodes then link to a client's page, thus, in essence, sharing some of their PageRank with the client's page. The paper by Bianchini et.al. [2] presents other scenarios for successfully boosting one's PageRank and provides helpful pictorial representations. Obviously, link farms are very troublesome for search engines. It appears that Google has tinkered with elements of \mathbf{v}^T to annihilate the PageRank

of link farms and their clients. Interestingly, this caused a court case between Google and the search engine optimization company SearchKing. The case ended in Google’s favor [13].

Several researchers have taken the personalization idea beyond its spam prevention abilities, creating personalized PageRanking systems. See the Stanford research papers [8, 9, 6] for the various efficient implementations of PageRank that incorporate user preferences. This is clearly a hot area since some predict personalized engines as the future of search.

4.3 Convergence Criteria

The power method applied to $\bar{\mathbf{P}}$ is the predominant method for finding the important PageRank vector. Being an iterative method, the power method continues until some termination criterion is met. In section 3.1, we mentioned the traditional termination criterion for the power method: stop when the residual (as measured by the difference between successive iterates) is less than some predetermined tolerance. However, Haveliwala has rightfully noted that the exact values of the PageRank vector are not as important as the correct ordering of the values in this vector [5]. That is, iterate until the ordering of the approximate PageRank vector obtained by the power method converges. Considering the scope of this problem, saving just a handful of iterations is praiseworthy. Haveliwala’s experiments show that the savings could be even more substantial on some datasets. As few as 10 iterations produced a good approximate ordering, competitive with the exact ordering produced by the traditional convergence measure.

4.4 Sensitivity, Stability and Condition Numbers

So far we have discussed ideas for changing some parameters in the PageRank model. A natural question is how such changes affect the PageRank vector. Regarding the issues of sensitivity and stability, one would like to know how changes in \mathbf{P} affect $\boldsymbol{\pi}^T$. The two different formulations of the PageRank problem, the linear system formulation and the eigenvector formulation, give some insight. The PageRank problem in its most general linear system form is

$$\boldsymbol{\pi}^T(\mathbf{I} - \alpha\mathbf{P}) = (1 - \alpha)\mathbf{v}^T.$$

Section 3.2 listed a property pertaining to the condition number of the linear system, $\kappa_\infty(\mathbf{I} - \alpha\mathbf{P}) = \frac{1+\alpha}{1-\alpha}$. As $\alpha \rightarrow 1$, the linear system becomes more ill-conditioned, meaning that a small change in the coefficient matrix creates a large change in the solution vector. However, $\boldsymbol{\pi}^T$ is actually an eigenvector for the corresponding Markov chain. While elements in the solution vector may change greatly for small changes in the coefficient matrix, the direction of the vector may change minutely. Once the solution is normalized to create a probability vector, the effect is minimal. The ill-conditioning of the linear system does not imply that the corresponding eigensystem is ill-conditioned, a fact documented by Wilkinson (with respect to the inverse iteration method) [14].

To answer the questions about how changes in \mathbf{P} affect $\boldsymbol{\pi}^T$ what we need to examine is eigenvector sensitivity, not linear system sensitivity. A crude statement about eigenvector sensitivity is that if a simple eigenvalue is close to the other eigenvalues, then the corresponding eigenvector is very sensitive to perturbations in \mathbf{P} . Then for the PageRank problem, one would conclude that $\boldsymbol{\pi}^T$ is insensitive for α away from 1, since its eigenvalue $\lambda_1 = 1$ is well-separated from $\lambda_2 = \alpha$. However, if a simple eigenvalue is well-separated from the other eigenvalues, its corresponding eigenvector is not necessarily insensitive to perturbations. A more rigorous measure of eigenvector sensitivity for Markov chains was developed by Meyer and Stewart [11] and Meyer and Golub [4]. In those papers, the condition number with respect to

the ∞ -norm for the stationary vector is defined as

$$\kappa_\infty(\boldsymbol{\pi}^T) = \|(\mathbf{I} - \mathbf{P})^\# \|_\infty,$$

where $(\mathbf{I} - \mathbf{P})^\#$ is the group inverse of the singular, rank $n - 1$ transition rate matrix $(\mathbf{I} - \mathbf{P})$. If the elements in $(\mathbf{I} - \mathbf{P})^\#$ are large, then small changes in the coefficient matrix \mathbf{P} can produce large changes in the eigenvector $\boldsymbol{\pi}^T$.

In order to understand the sensitivity of the PageRank vector, we calculate the group inverse of $(\mathbf{I} - \bar{\mathbf{P}})$, the transition rate matrix for the PageRank problem.

$$(\mathbf{I} - \bar{\mathbf{P}})^\# = (\mathbf{I} - \alpha\mathbf{P} - (1 - \alpha)\mathbf{e}\mathbf{v}^T)^\#.$$

The rank-one update formula for group inversion developed by Meyer and Shoaf [10] gives the following equation.

$$(\mathbf{I} - \bar{\mathbf{P}})^\# = [\mathbf{I} + (1 - \alpha)\mathbf{e}\mathbf{v}^T(\mathbf{I} - \alpha\mathbf{P})^{-1} + (2 - \alpha)\mathbf{e}\mathbf{v}^T](\mathbf{I} - \alpha\mathbf{P})^{-1}. \quad (5)$$

As $\alpha \rightarrow 1$, the bracketed factor in equation (5) approaches the matrix $[\mathbf{I} + \mathbf{e}\mathbf{v}^T]$. The remaining factor $(\mathbf{I} - \alpha\mathbf{P})^{-1}$ explodes as $\alpha \rightarrow 1$. Combining these two facts, we can conclude that the condition number of the PageRank problem explodes as $\alpha \rightarrow 1$. That is, as α increases, the PageRank vector becomes more and more sensitive to small changes in \mathbf{P} . Thus, Google's choice of $\alpha = .85$, while staying further from the true hyperlink structure of the Web, gives a much more stable PageRank than the "truer to the Web" choice of $\alpha = .99$.

Equation (5) also shows the minimal effect that the personalization probability vector \mathbf{v}^T has on the sensitivity of the PageRank vector.

4.5 Forcing Irreducibility

In the presentation of the PageRank model, we described the problem of reducibility. Simply put, the Markov chain produced from the hyperlink structure of the Web will undoubtedly be reducible and thus a long-run stationary vector will not exist for the subsequent Markov chain. The original solution of Brin and Page uses the method of maximal irreducibility, whereby every node is directly connected to every other node, hence irreducibility is trivially enforced. However, maximal irreducibility does alter the true nature of the Web, whereas other methods of forcing irreducibility seem less invasive and more inline with the Web's true nature. One such means, we label the method of minimal irreducibility, whereby a dummy node is added to the Web, which connects to every other node and to which every other node is connected, making the chain irreducible in a minimal sense. Our minimally irreducible $(n + 1) \times (n + 1)$ Markov matrix $\bar{\bar{\mathbf{P}}}$ is

$$\bar{\bar{\mathbf{P}}} = \left(\begin{array}{c|c} \frac{n}{n+1}\mathbf{P} & \frac{1}{n+1}\mathbf{e} \\ \hline \frac{1}{n+1}\mathbf{e}^T & \frac{1}{n+1} \end{array} \right).$$

This is clearly irreducible and hence $\boldsymbol{\pi}^T$, the PageRank vector, exists. Our goal is to examine the PageRank vector associated with this new $\bar{\bar{\mathbf{P}}}$ (after the weight of π_{n+1} , the PageRank of the dummy node, has been removed) and the convergence properties of the power method applied to $\bar{\bar{\mathbf{P}}}$. We begin by comparing the spectrum of $\bar{\bar{\mathbf{P}}}$ to the spectrum of \mathbf{P} .

Theorem 4.1 *Given the stochastic matrix \mathbf{P} with spectrum $\{1, \lambda_2, \lambda_3, \dots, \lambda_n\}$, the spectrum of $\bar{\bar{\mathbf{P}}} =$*

$$\left(\begin{array}{c|c} \frac{n}{n+1}\mathbf{P} & \frac{1}{n+1}\mathbf{e} \\ \hline \frac{1}{n+1}\mathbf{e}^T & \frac{1}{n+1} \end{array} \right) \text{ is } \{1, \frac{n}{n+1}\lambda_2, \frac{n}{n+1}\lambda_3, \dots, \frac{n}{n+1}\lambda_n, 0\}.$$

NOTE: \mathbf{P} may be reducible with $1 = \lambda_1 = \lambda_2, \dots, \lambda_k$ or irreducible with $\lambda_2 < 1$.

Proof: The eigenvalues of $\bar{\mathbf{P}}$ are the scalars λ that satisfy the characteristic equation

$$\begin{aligned}
& \det \left(\left(\begin{array}{c|c} \frac{n}{n+1} \mathbf{P} & \frac{1}{n+1} \mathbf{e} \\ \hline \frac{1}{n+1} \mathbf{e}^T & \frac{1}{n+1} \end{array} \right) - \lambda \mathbf{I}_{n+1} \right) = 0. \\
& \det \left(\left(\begin{array}{c|c} \frac{n}{n+1} \mathbf{P} & \frac{1}{n+1} \mathbf{e} \\ \hline \frac{1}{n+1} \mathbf{e}^T & \frac{1}{n+1} \end{array} \right) - \lambda \mathbf{I}_{n+1} \right) = \det \left(\begin{array}{c|c} \frac{n}{n+1} \mathbf{P} - \lambda \mathbf{I}_n & \frac{1}{n+1} \mathbf{e} \\ \hline \frac{1}{n+1} \mathbf{e}^T & \frac{1}{n+1} - \lambda \end{array} \right) \\
& = \det \left(\begin{array}{c|c} \frac{n}{n+1} \mathbf{P} - \lambda \mathbf{I}_n & \frac{1}{n+1} \mathbf{e} \\ \hline \frac{1}{n+1} \mathbf{e}^T & \frac{1-n\lambda-\lambda}{n+1} \end{array} \right) \\
& = \det \left(\frac{1-n\lambda-\lambda}{n+1} \right) \det \left(\frac{n}{n+1} \mathbf{P} - \lambda \mathbf{I}_n - \frac{1}{n+1} \mathbf{e} \left(\frac{1-n\lambda-\lambda}{n+1} \right)^{-1} \frac{1}{n+1} \mathbf{e}^T \right) \\
& = \frac{1-n\lambda-\lambda}{n+1} \det \left(\frac{n}{n+1} \mathbf{P} - \lambda \mathbf{I}_n - \frac{1}{(n+1)(1-n\lambda-\lambda)} \mathbf{e} \mathbf{e}^T \right) \\
& = \frac{1-n\lambda-\lambda}{n+1} \det \left(\frac{n}{n+1} \mathbf{P} - \lambda \mathbf{I}_n \right) \left(1 - \frac{1}{(n+1)(1-n\lambda-\lambda) \left(\frac{n}{n+1} - \lambda \right)} \mathbf{e}^T \mathbf{e} \right) \\
& = \frac{1-n\lambda-\lambda}{n+1} \left(1 - \frac{n}{(n+1)(1-n\lambda-\lambda) \left(\frac{n}{n+1} - \lambda \right)} \right) \det \left(\frac{n}{n+1} \mathbf{P} - \lambda \mathbf{I}_n \right) \\
& = \frac{\lambda^2(n^2+2n+1) - \lambda(n^2+2n+1)}{(n+1)(n-n\lambda-\lambda)} \det \left(\frac{n}{n+1} \mathbf{P} - \lambda \mathbf{I}_n \right) \\
& = \frac{(n+1)^2 \lambda(\lambda-1)}{(n+1)^2 \left(\frac{n}{n+1} - \lambda \right)} \left(\frac{n}{n+1} \lambda_1(\mathbf{P}) - \lambda \right) \left(\frac{n}{n+1} \lambda_2(\mathbf{P}) - \lambda \right) \cdots \left(\frac{n}{n+1} \lambda_n(\mathbf{P}) - \lambda \right) \\
& = \lambda(\lambda-1) \left(\frac{n}{n+1} \lambda_2(\mathbf{P}) - \lambda \right) \left(\frac{n}{n+1} \lambda_3(\mathbf{P}) - \lambda \right) \cdots \left(\frac{n}{n+1} \lambda_n(\mathbf{P}) - \lambda \right).
\end{aligned}$$

Thus, the eigenvalues of $\bar{\mathbf{P}}$ are $\{1, \frac{n}{n+1} \lambda_2(\mathbf{P}), \frac{n}{n+1} \lambda_3(\mathbf{P}), \dots, \frac{n}{n+1} \lambda_n(\mathbf{P}), 0\}$. □

One implication of this theorem for the PageRank problem is that now the subdominant eigenvalue, the measure affecting the convergence of the power method, is very close to 1. Recall that the Web is reducible so that $\lambda_2(\mathbf{P}) = 1$ (and possibly many more eigenvalues are 1). Thus, $\lambda_2(\bar{\mathbf{P}}) = \frac{n}{n+1} \lambda_2(\mathbf{P}) = \frac{n}{n+1}$. For even small Intranets, such the Mathworks net, $n = 517$, giving $\lambda_2(\bar{\mathbf{P}}) = .9981$, and roughly 9533 power iterations are required before the residual is less than 10^{-8} . The effect is even more pronounced for Google's Web collection with $n = 3.4$ billion, $\lambda_2(\bar{\mathbf{P}}) = .9999999970588$, requiring about 62.6 billion power iterations. What a striking contrast to Google's traditional method where $\lambda_2(\bar{\mathbf{P}}) = .85$ and only about 114 iterations are required to achieve the same precision.

Despite this bad news, we continue with our examination of the minimally irreducible method as it has some interesting connections to the maximally irreducible method. Writing the power method on the partitioned matrix $\bar{\mathbf{P}}$ gives

$$(\boldsymbol{\pi}^T \mid \pi_{n+1}) \left(\begin{array}{c|c} \frac{n}{n+1} \mathbf{P} & \frac{1}{n+1} \mathbf{e} \\ \hline \frac{1}{n+1} \mathbf{e}^T & \frac{1}{n+1} \end{array} \right) (\boldsymbol{\pi}^T \mid \pi_{n+1}),$$

which gives the following system of equations.

$$\boldsymbol{\pi}^T = \frac{n}{n+1}\boldsymbol{\pi}^T\mathbf{P} + \frac{\pi_{n+1}}{n+1}\mathbf{e}^T, \quad (6)$$

$$\pi_{n+1} = \frac{1}{n+1}\boldsymbol{\pi}^T\mathbf{e} + \frac{1}{n+1}\pi_{n+1}. \quad (7)$$

Solving for π_{n+1} in equation (7) gives $\pi_{n+1} = \frac{1}{n+1}$. Backsubstituting this value for π_{n+1} into equation (6) gives

$$\boldsymbol{\pi}^T = \frac{n}{n+1}\boldsymbol{\pi}^T\mathbf{P} + \frac{1}{(n+1)^2}\mathbf{e}^T. \quad (8)$$

Equation (8) is familiar; it looks very much like the traditional PageRank power method with $\alpha = \frac{n}{n+1}$. In fact, some more algebra shows that this is indeed the case. The PageRank power method with $\alpha = \frac{n}{n+1}$ is

$$\mathbf{x}^T\bar{\mathbf{P}} = \frac{n}{n+1}\mathbf{x}^T\mathbf{P} + \frac{1}{n(n+1)}\mathbf{x}^T\mathbf{e}\mathbf{e}^T.$$

In the minimally irreducible case, $\mathbf{x}^T\mathbf{e} = 1 - \pi_{n+1}$, giving

$$\begin{aligned} \mathbf{x}^T\bar{\mathbf{P}} &= \frac{n}{n+1}\mathbf{x}^T\mathbf{P} + \left(\frac{1}{n(n+1)}\right)\left(\frac{n}{n+1}\right)\mathbf{e}^T \\ &= \frac{n}{n+1}\mathbf{x}^T\mathbf{P} + \frac{1}{(n+1)^2}\mathbf{e}^T \end{aligned}$$

Clearly, our minimally irreducible method is a special case of PageRank's maximally irreducible method. This analysis shows the flexibility inherent in the PageRank method with the α parameter. The parameter α allows Google to cover a range of forced irreducibility methods, from the minimally irreducible method to the maximally irreducible method. We note that other researchers have suggested different means of forcing irreducibility [2].

5 Conclusion

In this paper, we presented the basic PageRank model used by the popular search engine Google. We listed the various adaptations that have been made to the model and discussed the implications of each. Our new contributions consist of a discussion of the sensitivity of the PageRank vector, a presentation of an alternate method for forcing irreducibility on the Markov chain, as well as proofs of the spectrums of the PageRank and a related matrix.

Disclaimer We mention that PageRank is just one of many measures employed by Google to return relevant results to users. Many other heuristics are part of this successful engine; we have focused on only one.

Acknowledgements We thank Cleve Moler for sharing his Mathworks dataset, `mathworks.dat`, and other web-crawling m-files.

A Spectrum of Google $\bar{\mathbf{P}} = \alpha\mathbf{P} + (1 - \alpha)\mathbf{e}\mathbf{v}^T$

Theorem A.1 *Given the spectrum of the stochastic matrix \mathbf{P} is $\{1, \lambda_2, \lambda_3, \dots, \lambda_n\}$, the spectrum of $\bar{\mathbf{P}} = \alpha\mathbf{P} + (1 - \alpha)\mathbf{e}\mathbf{v}^T$ is $\{1, \alpha\lambda_2, \alpha\lambda_3, \dots, \alpha\lambda_n\}$, where \mathbf{v}^T is a probability vector.*

NOTE: \mathbf{P} may be reducible with $1 = \lambda_1 = \lambda_2, \dots, \lambda_k$ or irreducible with $\lambda_2 < 1$.

Proof:

$$\begin{aligned} \det(\alpha\mathbf{P} + (1 - \alpha)\mathbf{e}\mathbf{v}^T - \lambda\mathbf{I}) &= \det(\alpha\mathbf{P} - \lambda\mathbf{I} + (1 - \alpha)\mathbf{e}\mathbf{v}^T) \\ &= \det(\alpha\mathbf{P} - \lambda\mathbf{I})(1 + \mathbf{v}^T(\alpha\mathbf{P} - \lambda\mathbf{I})^{-1}(1 - \alpha)\mathbf{e}) \\ &= \det(\alpha\mathbf{P} - \lambda\mathbf{I})(1 + (1 - \alpha)\mathbf{v}^T\mathbf{e}(\frac{1}{\alpha - \lambda})), \end{aligned}$$

since $(\alpha\mathbf{P} - \lambda\mathbf{I})^{-1}\mathbf{e} = \frac{\mathbf{e}}{\alpha - \lambda}$. Also, since \mathbf{v}^T is a probability vector, $\mathbf{v}^T\mathbf{e} = 1$. Thus,

$$\begin{aligned} \det(\alpha\mathbf{P} + (1 - \alpha)\mathbf{e}\mathbf{v}^T - \lambda\mathbf{I}) &= \det(\alpha\mathbf{P} - \lambda\mathbf{I})(1 + \frac{1 - \alpha}{\alpha - \lambda}) \\ &= \det(\alpha\mathbf{P} - \lambda\mathbf{I})(\frac{1 - \lambda}{\alpha - \lambda}) \\ &= (\alpha - \lambda)(\alpha\lambda_2(\mathbf{P}) - \lambda)(\alpha\lambda_3(\mathbf{P}) - \lambda) \cdots (\alpha\lambda_n(\mathbf{P}) - \lambda) \frac{(1 - \lambda)}{(\alpha - \lambda)}. \end{aligned}$$

Thus, the eigenvalues of $\bar{\mathbf{P}} = \alpha\mathbf{P} + (1 - \alpha)\mathbf{e}\mathbf{v}^T$ are $\{1, \alpha\lambda_2(\mathbf{P}), \alpha\lambda_3(\mathbf{P}), \dots, \alpha\lambda_n(\mathbf{P})\}$. □

For a reducible \mathbf{P} with several unit eigenvalues, $\lambda_2(\bar{\mathbf{P}}) = \alpha$.

References

- [1] Abraham Berman and Robert J. Plemmons. *Nonnegative matrices in the mathematical sciences*. Academic Press, Inc., 1979.
- [2] Monica Bianchini, Marco Gori, and Franco Scarselli. Inside pagerank. *ACM Transactions on Internet Technology*, 2003. In press.
- [3] Sergey Brin, Rajeev Motwani, Lawrence Page, and Terry Winograd. What can you do with a web in your pocket? *Data Engineering Bulletin*, 21:37–47, 1998.
- [4] Gene H. Golub and Carl D. Meyer. Using the QR factorization and group inverse to compute, differentiate and estimate the sensitivity of stationary probabilities for Markov chains. *SIAM Journal on Algebraic and Discrete Methods*, 17:273–281, 1986.
- [5] Taher H. Haveliwala. Efficient computation of PageRank. Technical report, Computer Science Department, Stanford University, 1999.
- [6] Taher H. Haveliwala. Topic-sensitive PageRank. In *The Eleventh International WWW Conference*, May 2002.
- [7] Taher H. Haveliwala and Sepandar D. Kamvar. The second eigenvalue of the google matrix. Technical report, Stanford University, 2003.
- [8] Taher H. Haveliwala, Sepandar D. Kamvar, and Glen Jeh. An analytical comparison of approaches to personalizing pagerank. Technical report, Stanford University, 2003.
- [9] Glen Jeh and Jennifer Widom. Scaling personalized web search. Technical report, Stanford University, 2002.
- [10] Carl D. Meyer and James M. Shoaf. Updating finite Markov chains by using techniques of group matrix inversion. *Journal of Statistical Computation and Simulation*, 11:163–181, 1980.
- [11] Carl D. Meyer and G. W. Stewart. Derivatives and perturbations of eigenvectors. *SIAM Journal on Numerical Analysis*, 25:679–691, 1988.
- [12] William J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1994.
- [13] Michael Totty and Mylene Mangalindan. As google becomes web’s gatekeeper, sites fight to get in. *Wall Street Journal*, CCXLI(39), 2003. February 26.
- [14] James H. Wilkinson. *The Algebraic Eigenvalue Problem*. Claredon Press, 1965.