SIAM J. MATRIX ANAL. APPL.
Vol. 29, No. 4, pp. 1281–1296

PAGERANK COMPUTATION, WITH SPECIAL ATTENTION TO DANGLING NODES*

ILSE C. F. IPSEN† AND TERESA M. SELEE†

Abstract. We present a simple algorithm for computing the PageRank (stationary distribution) of the stochastic Google matrix G. The algorithm lumps all dangling nodes into a single node. We express lumping as a similarity transformation of G and show that the PageRank of the nondangling nodes can be computed separately from that of the dangling nodes. The algorithm applies the power method only to the smaller lumped matrix, but the convergence rate is the same as that of the power method applied to the full matrix G. The efficiency of the algorithm increases as the number of dangling nodes increases. We also extend the expression for PageRank and the algorithm to more general Google matrices that have several different dangling node vectors, when it is required to distinguish among different classes of dangling nodes. We also analyze the effect of the dangling node vector on the PageRank and show that the PageRank of the dangling nodes depends strongly on that of the nondangling nodes but not vice versa. Last we present a Jordan decomposition of the Google matrix for the (theoretical) extreme case when all Web pages are dangling nodes.

Key words. stochastic matrix, stationary distribution, lumping, rank-one matrix, power method, Jordan decomposition, similarity transformation, Google

AMS subject classifications. 65F10, 65F50, 65C40, 15A06, 15A18, 15A21, 15A51, 68P20

DOI. 10.1137/060664331

1. Introduction.

The order in which the search engine Google displays the Web pages is determined, to a large extent, by the *PageRank* vector [7, 33]. The PageRank vector contains, for every Web page, a ranking that reflects the importance of the Web page. Mathematically, the PageRank vector π is the stationary distribution of the so-called *Google matrix*, a sparse stochastic matrix whose dimension exceeds 11.5 billion [16]. The Google matrix G is a convex combination of two stochastic matrices

$$G = \alpha S + (1-\alpha)E, \qquad 0 \le \alpha < 1,$$

where the matrix S represents the link structure of the Web, and the primary purpose of the rank-one matrix E is to force uniqueness for π. In particular, element (i, j) of S is nonzero if Web page i contains a link pointing to Web page j.

However, not all Web pages contain links to other pages. Image files or pdf files, and uncrawled or protected pages have no links to other pages. These pages are called *dangling nodes*, and their number may exceed the number of nondangling pages [11, section 2]. The rows in the matrix S corresponding to dangling nodes would be zero if left untreated. Several ideas have been proposed to deal with the zero rows and force S to be stochastic [11]. The most popular approach adds artificial links to the dangling nodes, by replacing zero rows in the matrix with the same vector, w, so that the matrix S is stochastic.

It is natural as well as efficient to exclude the dangling nodes with their artificial links from the PageRank computation. This can be done, for instance, by

*Received by the editors July 5, 2006; accepted for publication (in revised form) by D. Boley May 7, 2007; published electronically December 19, 2007.
http://www.siam.org/journals/simax/29-4/66433.html
†Department of Mathematics, North Carolina State University, P.O. Box 8205, Raleigh, NC 27695-8205 (ipsen@ncsu.edu, tmselee@ncsu.edu, http://www4.ncsu.edu/˜ipsen, http://www4.ncsu.edu/˜tmselee).

"lumping" all the dangling nodes into a single node [32]. In section 3, we provide a rigorous justification for lumping the dangling nodes in the Google matrix G, by expressing lumping as a similarity transformation of G (Theorem 3.1). We show that the PageRank of the nondangling nodes can be computed separately from that of the dangling nodes (Theorem 3.2), and we present an efficient algorithm for computing PageRank by applying the power method only to the much smaller, lumped matrix (section 3.3). Because the dangling nodes are excluded from most of the computations, the operation count depends, to a large extent, on only the number of nondangling nodes, as opposed to the total number of Web pages. The algorithm has the same convergence rate as the power method applied to G, but is much faster because it operates on a much smaller matrix. The efficiency of the algorithm increases as the number of dangling nodes increases.

Many other algorithms have been proposed for computing PageRank, including classical iterative methods [1, 4, 30], Krylov subspace methods [13, 14], extrapolation methods [5, 6, 20, 26, 25], and aggregation/disaggregation methods [8, 22, 31]; see also the survey papers [2, 28] and the book [29]. Our algorithm is faster than the power method applied to the full Google matrix G, but retains all the advantages of the power method: It is simple to implement and requires minimal storage. Unlike Krylov subspace methods, our algorithm exhibits predictable convergence behavior and is insensitive to changes in the matrix [13]. Moreover, our algorithm should become more competitive as the Web frontier expands and the number of dangling nodes increases. The algorithms in [30, 32] are special cases of our algorithm because our algorithm allows the dangling node and personalization vectors to be different, and thereby facilitates the implementation of TrustRank [18]. TrustRank is designed to diminish the harm done by link spamming and was patented by Google in March 2005 [35]. Moreover, our algorithm can be extended to a more general Google matrix that contains several different dangling node vectors (section 3.4).

In section 4 we examine how the PageRanks of the dangling and nondangling nodes influence each other, as well as the effect of the dangling node vector w on the PageRanks of dangling and nondangling nodes. In particular we show (Theorem 4.1) that the PageRanks of the dangling nodes depend strongly on the PageRanks of the nondangling nodes but not vice versa. Finally, in section 5, we consider a (theoretical) extreme case, where the Web consists solely of dangling nodes. We present a Jordan decomposition for general rank-one matrices (Theorems 5.1 and 5.2) and deduce from it a Jordan decomposition for a Google matrix of rank one (Corollary 5.3).

2. The ingredients.

Let n be the number of Web pages and k the number of nondangling nodes among the Web pages, $1 \leq k < n$. We model the link structure of the Web by the $n \times n$ matrix

$$H \equiv \begin{bmatrix} H_{11} & H_{12} \\ 0 & 0 \end{bmatrix},$$

where the $k \times k$ matrix H_{11} represents the links among the nondangling nodes, and H_{12} represents the links from nondangling to dangling nodes; see Figure 2.1. The $n - k$ zero rows in H are associated with the *dangling nodes*.

The elements in the nonzero rows of H are nonnegative and sum to one,

$$H_{11} \geq 0, \qquad H_{12} \geq 0, \qquad H_{11}e + H_{12}e = e, \qquad \text{where} \quad e \equiv \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix},$$

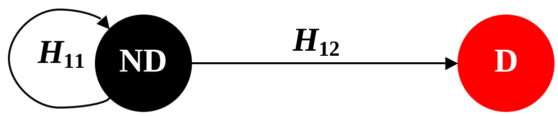

FIG. 2.1. *A simple model of the link structure of the Web. The sphere ND represents the set of nondangling nodes, and D represents the set of dangling nodes. The submatrix H_{11} represents all the links from nondangling nodes to nondangling nodes, while the submatrix H_{12} represents links from nondangling to dangling nodes.*

and the inequalities are to be interpreted elementwise. To obtain a stochastic matrix, we add artificial links to the dangling nodes. That is, we replace each zero row in H by the same *dangling node vector*

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}, \qquad w \geq 0, \qquad \|w\| \equiv w^T e = 1.$$

Here w_1 is $k \times 1$, w_2 is $(n-k) \times 1$, $\|\cdot\|$ denotes the one norm (maximal column sum), and the superscript T denotes the transpose. The resulting matrix

$$S \equiv H + dw^T = \begin{bmatrix} H_{11} & H_{12} \\ ew_1^T & ew_2^T \end{bmatrix}, \qquad \text{where} \quad d \equiv \begin{bmatrix} 0 \\ e \end{bmatrix},$$

is stochastic, that is, $S \geq 0$ and $Se = e$.

Finally, so as to work with a stochastic matrix that has a unique stationary distribution, one selects a *personalization vector*

$$v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \qquad v \geq 0, \qquad \|v\| = 1,$$

where v_1 is $k \times 1$ and v_2 is $(n-k) \times 1$, and defines the Google matrix as the convex combination

$$G \equiv \alpha S + (1-\alpha) e v^T, \qquad 0 \leq \alpha < 1.$$

Although the stochastic matrix G may not be primitive or irreducible, its eigenvalue 1 is distinct and the magnitude of all other eigenvalues is bounded by α [12, 19, 25, 26, 34]. Therefore G has a unique stationary distribution,

$$\pi^T G = \pi^T, \qquad \pi \geq 0, \qquad \|\pi\| = 1.$$

The stationary distribution π is called *PageRank*. Element i of π represents the PageRank for Web page i.

If we partition the PageRank conformally with G,

$$\pi = \begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix},$$

then π_1 represents the PageRank associated with the nondangling nodes and π_2 represents the PageRank of the dangling nodes.

The identity matrix of order n will be denoted by $I_n \equiv [e_1 \cdots e_n]$, or simply by I.

3. Lumping. We show that lumping can be viewed as a similarity transformation of the Google matrix; we derive an expression for PageRank in terms of the stationary distribution of the lumped matrix; we present an algorithm for computing PageRank that is based on lumping; and we extend everything to a Google matrix that has several different dangling node vectors, when it is required to distinguish among different classes of dangling nodes.

It was observed in [32] that the Google matrix represents a lumpable Markov chain. The concept of *lumping* was originally introduced for general Markov matrices, to speed up the computation of the stationary distribution or to obtain bounds [9, 17, 24, 27]. Below we paraphrase lumpability [27, Theorem 6.3.2] in matrix terms: Let P be a permutation matrix and

$$PMP^T = \begin{bmatrix} M_{11} & \cdots & M_{1,k+1} \\ \vdots & & \vdots \\ M_{k+1,1} & \cdots & M_{k+1,k+1} \end{bmatrix}$$

be a partition of a stochastic matrix M. Then M is *lumpable* with respect to this partition if each vector $M_{ij}e$ is a multiple of the all-ones vector e, $i \neq j$, $1 \leq i, j \leq k+1$.

The Google matrix G is lumpable if all dangling nodes are lumped into a single node [32, Proposition 1]. We condense the notation in section 2 and write the Google matrix as

$$G = \begin{bmatrix} G_{11} & G_{12} \\ eu_1^T & eu_2^T \end{bmatrix}, \qquad \text{where} \quad u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \equiv \alpha w + (1-\alpha)v, \tag{3.1}$$

G_{11} is $k \times k$, and G_{12} is $(n-k) \times k$. Here element (i, j) of G_{11} corresponds to block M_{ij}, $1 \leq i, j \leq k$; row i of G_{12} corresponds to block $M_{i,k+1}$, $1 \leq i \leq k$; column i of eu_1^T corresponds to $M_{k+1,i}$, $1 \leq i \leq k$; and eu_2^T corresponds to $M_{k+1,k+1}$.

3.1. Similarity transformation. We show that lumping the dangling nodes in the Google matrix can be accomplished by a similarity transformation that reduces G to block upper triangular form.

THEOREM 3.1. *With the notation in section* 2 *and the matrix G as partitioned in* (3.1)*, let*

$$X \equiv \begin{bmatrix} I_k & 0 \\ 0 & L \end{bmatrix}, \qquad \textit{where} \quad L \equiv I_{n-k} - \frac{1}{n-k}\hat{e}e^T \quad \textit{and} \quad \hat{e} \equiv e - e_1 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Then

$$XGX^{-1} = \begin{bmatrix} G^{(1)} & * \\ 0 & 0 \end{bmatrix}, \qquad \textit{where} \quad G^{(1)} \equiv \begin{bmatrix} G_{11} & G_{12}e \\ u_1^T & u_2^T e \end{bmatrix}.$$

The matrix $G^{(1)}$ is stochastic of order $k+1$ with the same nonzero eigenvalues as G.

Proof. From

$$X^{-1} = \begin{bmatrix} I_k & 0 \\ 0 & L^{-1} \end{bmatrix}, \qquad L^{-1} = I_{n-k} + \hat{e}e^T,$$

it follows that

$$XGX^{-1} = \begin{bmatrix} G_{11} & G_{12}(I + \hat{e}e^T) \\ e_1 u_1^T & e_1 u_2^T (I + \hat{e}e^T) \end{bmatrix}$$

has the same eigenvalues as G. In order to reveal the eigenvalues, we choose a different partitioning and separate the leading $k+1$ rows and columns, and observe that

$$G_{12}(I + \hat{e}e^T)e_1 = G_{12}e, \qquad u_2^T (I + \hat{e}e^T)e_1 = u_2^T e$$

to obtain the block triangular matrix

$$XGX^{-1} = \begin{bmatrix} G^{(1)} & * \\ 0 & 0 \end{bmatrix}$$

with at least $n-k-1$ zero eigenvalues. ◻

3.2. Expression for PageRank. We give an expression for the PageRank π in terms of the stationary distribution σ of the small matrix $G^{(1)}$.

THEOREM 3.2. *With the notation in section* 2 *and the matrix* G *as partitioned in* (3.1)*, let*

$$\sigma^T \begin{bmatrix} G_{11} & G_{12}e \\ u_1^T & u_2^T e \end{bmatrix} = \sigma^T, \qquad \sigma \geq 0, \qquad \|\sigma\| = 1$$

and partition $\sigma^T = \begin{bmatrix} \sigma_{1:k}^T & \sigma_{k+1} \end{bmatrix}$*, where* σ_{k+1} *is a scalar. Then the PageRank equals*

$$\pi^T = \begin{bmatrix} \sigma_{1:k}^T & \sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix} \end{bmatrix}.$$

Proof. As in the proof of Theorem 3.1, we write

$$XGX^{-1} = \begin{bmatrix} G^{(1)} & G^{(2)} \\ 0 & 0 \end{bmatrix},$$

where

$$G^{(2)} \equiv \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} (I + \hat{e}e^T)[e_2 \cdots e_{n-k}].$$

The vector $\begin{bmatrix} \sigma^T & \sigma^T G^{(2)} \end{bmatrix}$ is an eigenvector for XGX^{-1} associated with the eigenvalue $\lambda = 1$. Hence

$$\hat{\pi} \equiv \begin{bmatrix} \sigma^T & \sigma^T G^{(2)} \end{bmatrix} X$$

is an eigenvector of G associated with $\lambda = 1$ and a multiple of the stationary distribution π of G. Since $G^{(1)}$ has the same nonzero eigenvalues as G, and the dominant eigenvalue 1 of G is distinct [12, 19, 25, 26, 34], the stationary distribution σ of $G^{(1)}$ is unique.

Next we express $\hat{\pi}$ in terms of quantities in the matrix G. We return to the original partitioning which separates the leading k elements,

$$\hat{\pi}^T = \begin{bmatrix} \sigma_{1:k}^T & \begin{pmatrix} \sigma_{k+1} & \sigma^T G^{(2)} \end{pmatrix} \end{bmatrix} \begin{bmatrix} I_k & 0 \\ 0 & L \end{bmatrix}.$$

Multiplying out

$$\hat{\pi}^T = \begin{bmatrix} \sigma_{1:k}^T & \begin{pmatrix} \sigma_{k+1} & \sigma^T G^{(2)} \end{pmatrix} L \end{bmatrix}$$

shows that $\hat{\pi}$ has the same leading k elements as σ.

We now examine the trailing $n-k$ components of $\hat{\pi}^T$. To this end we partition the matrix $L = I_{n-k} - \frac{1}{n-k}\hat{e}e$ and distinguish the first row and column,

$$L = \begin{bmatrix} 1 & 0 \\ -\frac{1}{n-k}e & I - \frac{1}{n-k}ee^T \end{bmatrix}.$$

Then the eigenvector part associated with the dangling nodes is

$$z^T \equiv \begin{bmatrix} \sigma_{k+1} & \sigma^T G^{(2)} \end{bmatrix} L = \begin{bmatrix} \sigma_{k+1} - \frac{1}{n-k}\sigma^T G^{(2)} e & \sigma^T G^{(2)} \left(I - \frac{1}{n-k}ee^T \right) \end{bmatrix}.$$

To remove the terms containing $G^{(2)}$ in z, we simplify

$$(I + \hat{e}e^T)[e_2 \cdots e_{n-k}]e = (I + \hat{e}e^T)\hat{e} = (n-k)\hat{e}.$$

Hence

$$G^{(2)} e = (n-k) \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} \hat{e} \tag{3.2}$$

and

$$\begin{aligned} \frac{1}{n-k}\sigma^T G^{(2)} e &= \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} \hat{e} = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e - \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e_1 \\ &= \sigma_{k+1} - \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e_1, \end{aligned}$$

where we used $\hat{e} = e - e_1$, and the fact that σ is the stationary distribution of $G^{(1)}$, so

$$\sigma_{k+1} = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e.$$

Therefore the leading element of z equals

$$z_1 = \sigma_{k+1} - \frac{1}{n-k}\sigma^T G^{(2)} e = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} e_1.$$

For the remaining elements of z, we use (3.2) to simplify

$$G^{(2)} \left(I - \frac{1}{n-k}ee^T \right) = G^{(2)} - \frac{1}{n-k}G^{(2)}ee^T = G^{(2)} - \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} \hat{e}e^T.$$

Replacing

$$(I + \hat{e}e^T)[e_2 \cdots e_{n-k}] = [e_2 \cdots e_{n-k}] + \hat{e}e^T$$

in $G^{(2)}$ yields

$$z_{2:n-k}^T = \sigma^T G^{(2)} \left(I - \frac{1}{n-k}ee^T \right) = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix} [e_2 \cdots e_{n-k}].$$

Therefore the eigenvector part associated with the dangling nodes is

$$z = \begin{bmatrix} z_1 & z_{2:n-k}^T \end{bmatrix} = \sigma^T \begin{bmatrix} G_{12} \\ u_2^T \end{bmatrix}$$

and

$$\hat{\pi} = \begin{bmatrix} \sigma_{1:k}^T & \sigma^T \begin{pmatrix} G_{12} \\ u_2^T \end{pmatrix} \end{bmatrix}.$$

Since π is unique, as discussed in section 2, we conclude that $\hat{\pi} = \pi$ if $\hat{\pi}^T e = 1$. This follows, again, from the fact that σ is the stationary distribution of $G^{(1)}$ and $\sigma^T \left[\begin{smallmatrix} G_{12} \\ u_2^T \end{smallmatrix}\right] e = \sigma_{k+1}$. $\square$

3.3. Algorithm. We present an algorithm, based on Theorem 3.2, for computing the PageRank π from the stationary distribution σ of the lumped matrix

$$G^{(1)} \equiv \begin{bmatrix} G_{11} & G_{12}e \\ u_1^T & u_2^T e \end{bmatrix}.$$

The input to the algorithm consists of the nonzero elements of the hyperlink matrix H, the personalization vector v, the dangling node vector w, and the amplification factor α. The output of the algorithm is an approximation $\hat{\pi}$ to the PageRank π, which is computed from an approximation $\hat{\sigma}$ of σ.

ALGORITHM 3.1.

% Inputs: H, v, w, α Output: $\hat{\pi}$

% Power method applied to $G^{(1)}$:

Choose a starting vector $\hat{\sigma}^T = \begin{bmatrix} \hat{\sigma}_{1:k}^T & \hat{\sigma}_{k+1} \end{bmatrix}$ with $\hat{\sigma} \geq 0$, $\|\hat{\sigma}\| = 1$.

While not converged

$\hat{\sigma}_{1:k}^T = \alpha \hat{\sigma}_{1:k}^T H_{11} + (1-\alpha) v_1^T + \alpha \hat{\sigma}_{k+1} w_1^T$

$\hat{\sigma}_{k+1} = 1 - \hat{\sigma}_{1:k}^T e$

end while

% Recover PageRank:

$\hat{\pi}^T = \begin{bmatrix} \hat{\sigma}_{1:k}^T & \alpha \hat{\sigma}_{1:k}^T H_{12} + (1-\alpha) v_2^T + \alpha \hat{\sigma}_{k+1} w_2^T \end{bmatrix}$.

Each iteration of the power method applied to $G^{(1)}$ involves a sparse matrix vector multiply with the $k \times k$ matrix H_{11} as well as several vector operations. Thus the dangling nodes are excluded from the power method computation. The convergence rate of the power method applied to G is α [23]. Algorithm 3.1 has the same convergence rate, because $G^{(1)}$ has the same nonzero eigenvalues as G (see Theorem 3.1), but is much faster because it operates on a smaller matrix whose dimension does not depend on the number of dangling nodes. The final step in Algorithm 3.1 recovers π via a single sparse matrix vector multiply with the $k \times (n-k)$ matrix H_{12}, as well as several vector operations.

Algorithm 3.1 is significantly faster than the power method applied to the full Google matrix G, but it retains all advantages of the power method: It is simple to implement and requires minimal storage. Unlike Krylov subspace methods, Algorithm 3.1 exhibits predictable convergence behavior and is insensitive to changes in the matrix [13]. The methods in [30, 32] are special cases of Algorithm 3.1 because they allow the dangling node vector to be different from the personalization vector, thereby facilitating the implementation of TrustRank [18]. TrustRank allows zero elements in the personalization vector v in order to diminish the harm done by link spamming. Algorithm 3.1 can also be extended to the situation when the Google matrix has several different dangling node vectors; see section 3.4.

The power method in Algorithm 3.1 corresponds to Stage 1 of the algorithm in [32]. However, Stage 2 of that algorithm involves the power method on a rank-two matrix of order $n-k+1$. In contrast, Algorithm 3.1 simply performs a single matrix vector multiply with the $k \times (n-k)$ matrix H_{12}. There is no proof that the two-stage algorithm in [32] does compute the PageRank.

3.4. Several dangling node vectors. So far we have treated all dangling nodes in the same way, by assigning them the same dangling node vector w. However, one dangling node vector may be inadequate for an advanced Web search. For instance, one may want to distinguish different types of dangling node pages based on their functions (e.g., text files, image files, videos, etc.); or one may want to personalize a Web search and assign different vectors to dangling node pages pertaining to different topics, different languages, or different domains; see the discussion in [32, section 8.2].

To facilitate such a model for an advanced Web search, we extend the single class of dangling nodes to $m \geq 1$ different classes, by assigning a different dangling node vector w_i to each class, $1 \leq i \leq m$. As a consequence we need to extend lumping to a more general Google matrix that is obtained by replacing the $n-k$ zero rows in the hyperlink matrix H by $m \geq 1$ possibly different dangling node vectors $w_1, \ldots, w_m$. The more general Google matrix is

$$F \equiv \begin{array}{c} \\ k \\ k_1 \\ \vdots \\ k_m \end{array} \begin{array}{c} \begin{array}{cccc} k & k_1 & \ldots & k_m \end{array} \\ \begin{pmatrix} F_{11} & F_{12} & \cdots & F_{1,m+1} \\ eu_{11}^T & eu_{12}^T & \cdots & eu_{1,m+1}^T \\ \vdots & \vdots & & \vdots \\ eu_{m1}^T & eu_{m2}^T & \cdots & eu_{m,m+1}^T \end{pmatrix} \end{array},$$

where

$$u_i \equiv \begin{bmatrix} u_{i1} \\ \vdots \\ u_{i,m+1} \end{bmatrix} \equiv \alpha w_i + (1-\alpha)v.$$

Let $\tilde{\pi}$ be the PageRank associated with F,

$$\tilde{\pi}^T F = \tilde{\pi}^T, \qquad \tilde{\pi} \geq 0, \qquad \|\tilde{\pi}\| = 1.$$

We explain our approach for the case when F has two types of dangling nodes,

$$F = \begin{array}{c} \\ k \\ k_1 \\ k_2 \end{array} \begin{array}{c} \begin{array}{ccc} k & k_1 & k_2 \end{array} \\ \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ eu_{11}^T & eu_{12}^T & eu_{13}^T \\ eu_{21}^T & eu_{22}^T & eu_{23}^T \end{pmatrix} \end{array}.$$

We perform the lumping by a sequence of similarity transformations that starts at the bottom of the matrix. The first similarity transformation lumps the dangling nodes represented by u_2 and leaves the leading block of order $k+k_1$ unchanged,

$$X_1 \equiv \begin{array}{c} \\ k+k_1 \\ k_2 \end{array} \begin{array}{c} \begin{array}{cc} k+k_1 & k_2 \end{array} \\ \begin{pmatrix} I & 0 \\ 0 & L_1 \end{pmatrix} \end{array},$$

where L_1 lumps the k_2 trailing rows and columns of F,

$$L_1 \equiv I - \frac{1}{k_2}\hat{e}e^T, \quad L_1^{-1} \equiv I + \hat{e}e^T, \qquad \hat{e} = e - e_1 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

Applying the similarity transformation to F gives

$$X_1FX_1^{-1} = \begin{array}{c c} & \begin{array}{cccc} k & k_1 & 1 & k_2-1 \end{array} \\ \begin{array}{l} k \\ k_1 \\ 1 \\ k_2-1 \end{array} & \begin{pmatrix} F_{11} & F_{12} & F_{13}e & \tilde{F}_{13} \\ eu_{11}^T & eu_{12}^T & (u_{13}^Te)e & e\tilde{u}_{13}^T \\ u_{21}^T & u_{22}^T & u_{23}^Te & \tilde{u}_{23}^T \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

with

$$\tilde{F}_{13} \equiv F_{13}L_1^{-1}\begin{bmatrix} e_2 & \cdots & e_{k_2} \end{bmatrix}, \qquad \tilde{u}_{j3}^T \equiv u_{j3}^TL_1^{-1}\begin{bmatrix} e_2 & \cdots & e_{k_2} \end{bmatrix}, \quad j = 1, 2.$$

The leading diagonal block of order $k + k_1 + 1$ is a stochastic matrix with the same nonzero eigenvalues as F. Before applying the second similarity transformation that lumps the dangling nodes represented by u_1, we move the rows with u_1 (and corresponding columns) to the bottom of the nonzero matrix, merely to keep the notation simple. The move is accomplished by the permutation matrix

$$P_1 \equiv [e_1 \cdots e_k \quad e_{k+k_1+1} \quad e_{k+1} \cdots e_{k+k_1} \quad e_{k+k_1+2} \cdots e_n].$$

The symmetrically permuted matrix

$$P_1X_1FX_1^{-1}P_1^T = \left[\begin{array}{ccc|c} F_{11} & F_{13}e & F_{12} & \tilde{F}_{13} \\ u_{21}^T & u_{23}^Te & u_{22}^T & \tilde{u}_{23}^T \\ eu_{11}^T & (u_{13}^Te)e & eu_{12}^T & e\tilde{u}_{13}^T \\ \hline 0 & 0 & 0 & 0 \end{array}\right]$$

retains a leading diagonal block that is stochastic. Now we repeat the lumping on dangling nodes represented by u_1, by means of the similarity transformation

$$X_2 \equiv \begin{array}{c c} & \begin{array}{ccc} k+1 & k_1 & k_2-1 \end{array} \\ \begin{array}{l} k+1 \\ k_1 \\ k_2-1 \end{array} & \begin{pmatrix} I & 0 & 0 \\ 0 & L_2 & 0 \\ 0 & 0 & I \end{pmatrix} \end{array},$$

where L_2 lumps the trailing k_1 nonzero rows,

$$L_2 \equiv I - \frac{1}{k_1}\hat{e}e^T, \qquad L_2^{-1} \equiv I + \hat{e}e^T.$$

The similarity transformation produces the lumped matrix

$$X_2P_1X_1FX_1^{-1}P_1^TX_2^{-1} = \begin{array}{c c} & \begin{array}{ccccc} k & 1 & 1 & k_1-1 & k_2-1 \end{array} \\ \begin{array}{l} k \\ 1 \\ 1 \\ k_1-1 \\ k_2-1 \end{array} & \begin{pmatrix} F_{11} & F_{13}e & F_{12}e & \tilde{F}_{12} & \tilde{F}_{13} \\ u_{21}^T & u_{23}^Te & u_{22}^Te & \tilde{u}_{22}^T & \tilde{u}_{23}^T \\ u_{11}^T & u_{13}^Te & u_{12}^Te & \tilde{u}_{12}^T & \tilde{u}_{13}^T \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}.$$

Finally, for notational purposes, we restore the original ordering of dangling nodes by permuting rows and columns $k + 1$ and $k + 2$,

$$P_2 \equiv [e_1 \cdots e_k \quad e_{k+2} \quad e_{k+1} \quad e_{k+3} \cdots e_n].$$

The final lumped matrix is

$$P_2X_2P_1X_1FX_1^{-1}P_1^TX_2^{-1}P_2^T = \begin{bmatrix} F_{11} & F_{12}e & F_{13}e & * \\ u_{11}^T & u_{12}^Te & u_{13}^Te & * \\ u_{21}^Te & u_{22}^Te & u_{23}^Te & * \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} F^{(1)} & * \\ 0 & 0 \end{bmatrix}.$$

The above discussion for $m = 2$ illustrates how to extend Theorems 3.1 and 3.2 to any number m of dangling node vectors.

THEOREM 3.3. *Define X_i as*

$$\begin{array}{l} \\ k+(i-1)+\sum_{j=1}^{m-i} k_j \\ k_{m-i+1} \\ 1-i+\sum_{j=m-i+2}^{m} k_j \end{array} \begin{array}{c} \begin{array}{ccc} k+(i-1)+\sum_{j=1}^{m-i} k_j & k_{m-i+1} & 1-i+\sum_{j=m-i+2}^{m} k_j \end{array} \\ \begin{pmatrix} I & 0 & 0 \\ 0 & L_i & 0 \\ 0 & 0 & I \end{pmatrix} \end{array}$$

and

$$P_i \equiv [e_1 \cdots e_k \quad e_{r+i} \quad e_{k+1} \cdots e_{r+i-1} \quad e_{r+i+1} \cdots e_n], \qquad r = k + \sum_{j=1}^{m-i} k_j.$$

Then

$$P_mX_mP_{m-1}X_{m-1}\cdots P_1X_1FX_1^{-1}P_1^T\cdots X_m^{-1}P_m^T = \begin{bmatrix} F^{(1)} & * \\ 0 & 0 \end{bmatrix},$$

where the lumped matrix

$$F^{(1)} \equiv \begin{bmatrix} F_{11} & F_{12}e & \cdots & F_{1,m+1}e \\ u_{11}^T & u_{12}^Te & \cdots & u_{1,m+1}^Te \\ \vdots & \vdots & & \vdots \\ u_{m1}^T & u_{m2}^Te & \cdots & u_{m,m+1}^Te \end{bmatrix}$$

is stochastic of order $k+m$ with the same nonzero eigenvalues as F.

THEOREM 3.4. *Let ρ be the stationary distribution of the lumped matrix*

$$F^{(1)} \equiv \begin{bmatrix} F_{11} & F_{12}e & \cdots & F_{1,m+1}e \\ u_{11}^T & u_{12}^Te & \cdots & u_{1,m+1}^Te \\ \vdots & \vdots & & \vdots \\ u_{m1}^T & u_{m2}^Te & \cdots & u_{m,m+1}^Te \end{bmatrix}; \tag{3.3}$$

that is,

$$\rho^T F^{(1)} = \rho^T, \qquad \rho \geq 0, \qquad \|\rho\| = 1.$$

With the partition $\rho^T = \begin{bmatrix}\rho_{1:k}^T & \rho_{k+1:k+m}^T\end{bmatrix}$, where $\rho_{k+1:k+m}$ is $m \times 1$, the PageRank of F equals

$$\tilde{\pi}^T = \left[\begin{array}{cc} \rho_{1:k}^T & \rho^T \begin{pmatrix} F_{12} & \cdots & F_{1,m+1} \\ u_{12}^T & \cdots & u_{1,m+1}^T \\ \vdots & & \vdots \\ u_{m2}^T & \cdots & u_{m,m+1}^T \end{pmatrix} \end{array}\right].$$

4. PageRanks of dangling versus nondangling nodes. We examine how the PageRanks of dangling and nondangling nodes influence each other, as well as the effect of the dangling node vector on the PageRanks.

From Theorem 3.2 and Algorithm 3.1, we see that the PageRank π_1 of the nondangling nodes can be computed separately from the PageRank π_2 of the dangling nodes, and that π_2 depends directly on π_1. The expressions below make this even clearer.

THEOREM 4.1. *With the notation in section* 2*,*

$$\pi_1^T = \left((1-\alpha)v_1^T + \rho w_1^T\right)(I - \alpha H_{11})^{-1},$$
$$\pi_2^T = \alpha \pi_1^T H_{12} + (1-\alpha)v_2^T + \alpha(1 - \|\pi_1\|)w_2^T,$$

where

$$\rho \equiv \alpha \frac{1 - (1-\alpha)v_1^T(I - \alpha H_{11})^{-1}e}{1 + \alpha w_1^T(I - \alpha H_{11})^{-1}e} \geq 0.$$

Proof. Rather than using Theorem 3.2 we found it easier just to start from scratch. From $G = \alpha(H + dw^T) + (1-\alpha)v^T$ and the fact that $\pi^T e = 1$, it follows that π is the solution to the linear system

$$\pi^T = (1-\alpha)v^T \left(I - \alpha H - \alpha d w^T\right)^{-1},$$

whose coefficient matrix is a strictly row diagonally dominant M-matrix [1, equation (5)], [4, equation (2), Proposition 2.4]. Since $R \equiv I - \alpha H$ is also an M-matrix, it is nonsingular, and the elements of R^{-1} are nonnegative [3, section 6]. The Sherman–Morrison formula [15, section 2.1.3] implies that

$$\left(R - \alpha d w^T\right)^{-1} = R^{-1} + \frac{\alpha R^{-1} d w^T R^{-1}}{1 - \alpha w^T R^{-1} d}.$$

Substituting this into the expression for π gives

$$\pi^T = (1-\alpha)v^T R^{-1} + \frac{\alpha(1-\alpha)v^T R^{-1} d}{1 - \alpha w^T R^{-1} d} w^T R^{-1}. \tag{4.1}$$

We now show that the denominator $1 - \alpha w^T R^{-1} d > 0$. Using the partition

$$R^{-1} = (I - \alpha H)^{-1} = \begin{bmatrix} (I - \alpha H_{11})^{-1} & \alpha\,(I - \alpha H_{11})^{-1} H_{12} \\ 0 & I \end{bmatrix}$$

yields

$$1 - \alpha w^T R^{-1} d = 1 - \alpha \left(\alpha w_1^T \left(I - \alpha H_{11}\right)^{-1} H_{12} e + w_2^T e\right). \tag{4.2}$$

Rewrite the term involving H_{12} by observing that $H_{11}e + H_{12}e = e$ and that $I - \alpha H_{11}$ is an M-matrix, so

$$0 \leq \alpha \left(I - \alpha H_{11}\right)^{-1} H_{12} e = e - (1-\alpha)\left(I - \alpha H_{11}\right)^{-1} e. \tag{4.3}$$

Substituting this into (4.2) and using $1 = w^T e = w_1^T e + w_2^T e$ shows that the denominator in the Sherman–Morrison formula is positive,

$$1 - \alpha w^T R^{-1} d = (1-\alpha)\left(1 + \alpha w_1^T \left(I - \alpha H_{11}\right)^{-1} e\right) > 0.$$

Furthermore, $0 \le \alpha < 1$ implies $1 - \alpha w^T R^{-1} d > 1 - \alpha$.

Substituting the simplified denominator into the expression (4.1) for π yields

$$\pi^T = (1-\alpha) v^T R^{-1} + \alpha \frac{v^T R^{-1} d}{1 + \alpha w_1^T (I - \alpha H_{11})^{-1} e} w^T R^{-1}. \tag{4.4}$$

We obtain for π_1

$$\pi_1^T = \left((1-\alpha) v_1^T + \alpha \frac{v^T R^{-1} d}{1 + \alpha w_1^T (I - \alpha H_{11})^{-1} e} w_1^T \right) (I - \alpha H_{11})^{-1}.$$

Combining the partitioning of R^{-1}, (4.3), and $v_1^T e + v_2^T e = 1$ gives

$$\begin{aligned} 0 \le v^T R^{-1} d &= \begin{bmatrix} v_1^T & v_2^T \end{bmatrix} \begin{bmatrix} (I - \alpha H_{11})^{-1} & \alpha (I - \alpha H_{11})^{-1} H_{12} \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 \\ e \end{bmatrix} \\ &= \alpha v_1^T (I - \alpha H_{11})^{-1} H_{12} e + v_2^T e \\ &= 1 - (1-\alpha) v_1^T (I - \alpha H_{11})^{-1} e. \end{aligned}$$

Hence $\pi_1^T = \left((1-\alpha) v_1^T + \rho w_1^T \right) (I - \alpha H_{11})^{-1}$ with $\rho > 0$.

To obtain the expression for π_2, observe that the second block element in

$$\pi^T (I - \alpha H - \alpha d w^T) = (1-\alpha) v^T$$

equals

$$-\alpha \pi_1^T H_{12} + \pi_2^T - \alpha \pi_2^T e w_2^T = (1-\alpha) v_2^T.$$

The result follows from $\pi_1^T e + \pi_2^T e = 1$. □

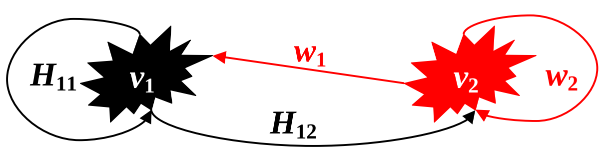

FIG. 4.1. *Sources of PageRank. Nondangling nodes receive their PageRank from v_1 and w_1, distributed through the links H_{11}. In contrast, the PageRank of the dangling nodes comes from v_2, w_2, and the PageRank of the nondangling nodes through the links H_{12}.*

Remark 4.1. We draw the following conclusions from Theorem 4.1 with regard to how dangling and nondangling nodes accumulate PageRank; see Figure 4.1.

- The PageRank π_1 of the nondangling nodes does *not* depend on the connectivity among the dangling nodes (elements of w_2), the personalization vector for the dangling nodes (elements of v_2), or the links from nondangling to dangling nodes (elements of H_{12}).
  To be specific, π_1 does not depend on individual elements of w_2, v_2, and H_{12}. Rather, the dependence is on the norms, through $\|v_2\| = 1 - \|v_1\|$, $\|w_2\| = 1 - \|w_1\|$, and $H_{12} e = e - H_{11} e$.
- The PageRank π_1 of the nondangling nodes does not depend on the PageRank π_2 of the dangling nodes or their number, because π_1 can be computed without knowledge of π_2.

- The nondangling nodes receive their PageRank π_1 from their personalization vector v_1 and the dangling node vector w_1, both of which are distributed through the links H_{11}.
- The dangling nodes receive their PageRank π_2 from three sources: the associated part v_2 of the personalization vector; the associated part w_2 of the dangling node vector; and the PageRank π_1 of the nondangling nodes filtered through the connecting links H_{12}.
  The links H_{12} determine how much PageRank flows from nondangling to dangling nodes.
- The influence of the associated dangling node vector w_2 on the PageRank π_2 of the dangling nodes diminishes as the combined PageRank $\|\pi_1\|$ of the nondangling nodes increases.

Taking norms in Theorem 4.1 gives a bound on the combined PageRank of the nondangling nodes. As in section 2, the norm is $\|z\| \equiv z^T e$ for $z \geq 0$.

COROLLARY 4.2. *With the assumptions of Theorem* 4.1*,*

$$\|\pi_1\| = \frac{(1-\alpha)\|v_1\|_H + \alpha\|w_1\|_H}{1+\alpha\|w_1\|_H},$$

where $\|z\|_H \equiv z^T(I-\alpha H_{11})^{-1}e$ *for any* $z \geq 0$ *and*

$$(1-\alpha)\|z\| \leq \|z\|_H \leq \frac{1}{1-\alpha}\|z\|.$$

Proof. Since $(I-\alpha H_{11})^{-1}$ is nonsingular with nonnegative elements, $\|\cdot\|_H$ is a norm. Let $\|\cdot\|_\infty$ be the infinity norm (maximal row sum). Then the Hölder inequality [15, section 2.2.2] implies for any $z \geq 0$,

$$\|z\|_H \leq \|z\| \, \|(I-\alpha H_{11})^{-1}\|_\infty \leq \frac{1}{1-\alpha}\|z\|.$$

As for the lower bound,

$$\|z\|_H \geq \|z\| - \alpha z^T H_{11} e \geq (1-\alpha)\|z\|. \qquad \square$$

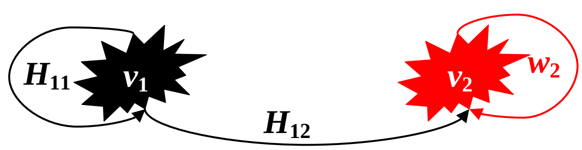

FIG. 4.2. *Sources of PageRank when $w_1 = 0$. The nondangling nodes receive their PageRank only from v_1. The dangling nodes, in contrast, receive their PageRank from v_2 and w_2, as well as from the PageRank of the nondangling nodes filtered through the links H_{12}.*

Corollary 4.2 implies that the combined PageRank $\|\pi_1\|$ of the nondangling nodes is an increasing function of $\|w_1\|$. In particular, when $w_1 = 0$, the combined PageRank $\|\pi_1\|$ is minimal among all w and the dangling vector w_2 has a stronger influence on the PageRank π_2 of the dangling nodes. The dangling nodes act like a sink and absorb more PageRank because there are no links back to the nondangling nodes; see Figure 4.2. When $w_1 = 0$ we get

$$\begin{aligned} \pi_1^T &= (1-\alpha)v_1^T(I-\alpha H_{11})^{-1}, \\ \pi_2^T &= \alpha\pi_1^T H_{12} + (1-\alpha)v_2^T + \alpha(1-\|\pi_1\|)w_2^T. \end{aligned} \tag{4.5}$$

In the other extreme case when $w_2 = 0$, the dangling nodes are not connected to each other; see Figure 4.3:

$$\begin{aligned}\pi_1^T &= \left((1-\alpha)v_1^T + \rho w_1^T\right)(I - \alpha H_{11})^{-1}, \\ \pi_2^T &= \alpha \pi_1^T H_{12} + (1-\alpha)v_2^T.\end{aligned} \tag{4.6}$$

In this case the PageRank π_1 of the nondangling nodes has only a positive influence on the PageRank of the dangling nodes.

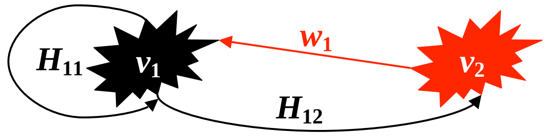

FIG. 4.3. *Sources of PageRank when $w_2 = 0$. The dangling nodes receive their PageRank only from v_2, and from the PageRank of the nondangling nodes filtered through the links H_{12}.*

An expression for π when dangling node and personalization vectors are the same, i.e., $w = v$, was given in [10],

$$\pi^T = (1-\alpha)\left(1 + \frac{\alpha v^T R^{-1} d}{1 - \alpha v^T R^{-1} d}\right) v^T R^{-1}, \qquad \text{where} \quad R \equiv I - \alpha H.$$

In this case the PageRank vector π is a multiple of the vector $v^T(I - \alpha H)^{-1}$.

5. Only dangling nodes. We examine the (theoretical) extreme case when all Web pages are dangling nodes. In this case the matrices S and G have rank one. We first derive a Jordan decomposition for general matrices of rank one, before we present a Jordan form for a Google matrix of rank one.

We start with rank-one matrices that are diagonalizable. The vector e_j denotes the jth column of the identity matrix I.

THEOREM 5.1 (eigenvalue decomposition). *Let $A = yz^T \neq 0$ be a real square matrix with $\lambda \equiv z^T y \neq 0$. If z has an element $z_j \neq 0$, then $X^{-1}AX = \lambda\, e_j e_j^T$, where*

$$X \equiv I + y e_j^T - \frac{1}{z_j} e_j z^T, \qquad X^{-1} = I - e_j e_j^T - \frac{1}{\lambda} y z^T + \frac{1 + y_j}{\lambda} e_j z^T.$$

Proof. The matrix A has a repeated eigenvalue zero and a distinct nonzero eigenvalue λ with right eigenvector y and left eigenvector z. From $\lambda y e_j^T = AX = \lambda X e_j e_j^T$ and $X^{-1}A = e_j z^T$ it follows that $X^{-1}X = I$ and $X^{-1}AX = \lambda e_j e_j^T$. $\square$

Now we consider rank-one matrices that are not diagonalizable. In this case all eigenvalues are zero, and the matrix has a Jordan block of order two.

THEOREM 5.2 (Jordan decomposition). *Let $A = yz^T \neq 0$ be a real square matrix with $z^T y = 0$. Then y and z have elements $y_j z_j \neq 0 \neq y_k z_k$, $j < k$. Define a symmetric permutation matrix P so that $Pe_k = e_{j+1}$ and $Pe_j = e_j$. Set $\hat{y} \equiv Py$ and $\hat{u} \equiv Pz - e_{j+1}$. Then $X^{-1}AX = e_j e_{j+1}^T$ with*

$$X \equiv P\left(I + \hat{y} e_j^T - \frac{1}{\hat{u}_j} e_j \hat{u}^T\right), \qquad X^{-1} = \left(I - e_j e_j^T + \frac{1}{\hat{y}_k} \hat{y} \hat{u}^T - \frac{1 + \hat{y}_j}{\hat{y}_k} e_j \hat{u}^T\right) P.$$

Proof. To satisfy $z^T y = 0$ for $y \neq 0$ and $z \neq 0$, we must have $y_j z_j \neq 0$ and $y_k z_k \neq 0$ for some $j < k$.

Since A is a rank-one matrix with all eigenvalues equal to zero, it must have a Jordan block of the form $\left[\begin{smallmatrix}0&1\\0&0\end{smallmatrix}\right]$. To reveal this Jordan block, set $\hat{z} \equiv Pz$,

$$\hat{X} \equiv \left(I + \hat{y}e_j^T - \frac{1}{\hat{u}_j}e_j\hat{u}^T\right), \qquad \hat{X}^{-1} = \left(I - e_je_j^T + \frac{1}{\hat{y}_k}\hat{y}\hat{u}^T - \frac{1+\hat{y}_j}{\hat{y}_k}e_j\hat{u}^T\right).$$

Then the matrix $\hat{A} \equiv \hat{y}\hat{z}^T$ has a Jordan decomposition $\hat{X}^{-1}\hat{A}\hat{X} = e_je_{j+1}^T$. This follows from $u_j = z_j$, $\hat{y}e_{j+1}^T = \hat{A}\hat{X} = \hat{X}e_je_{j+1}^T$, and $\hat{X}^{-1}\hat{A} = e_j\hat{z}^T$.

Finally, we undo the permutation by means of $X \equiv P\hat{X}$, $X^{-1} = \hat{X}^{-1}P$, so that $X^{-1}X = I$ and $X^{-1}AX = e_je_{j+1}^T$. □

Theorems 5.1 and 5.2 can also be derived from [21, Theorem 1.4].

In the (theoretical) extreme case when all Web pages are dangling nodes, the Google matrix is diagonalizable of rank one.

COROLLARY 5.3 (rank-one Google matrix). *With the notation in section* 2 *and* (3.1)*, let* $G = eu^T$. *Let* $u_j \neq 0$ *be a nonzero element of* u. *Then* $X^{-1}GX = e_je_j^T$ *with*

$$X = I + ee_j^T - \frac{1}{v_j}e_ju^T$$

and

$$X^{-1} = I - e_je_j^T - eu^T + 2e_ju^T.$$

In particular, $\pi^T = e_j^TX^{-1} = u^T$.

Proof. Since $1 = u^Te \neq 0$, the Google matrix is diagonalizable, and the expression in Theorem 5.1 applies. □

Corollary 5.3 can also be derived from [34, Theorems 2.1, 2.3].

REFERENCES

[1] A. ARASU, J. NOVAK, A. TOMKINS, AND J. TOMLIN, *PageRank computation and the structure of the web: Experiments and algorithms*, in Proceedings of the Eleventh International World Wide Web Conference (WWW2002), ACM Press, New York, 2002. Available online at http://www2002.org/CDROM/poster/173.pdf.

[2] P. BERKHIN, *A survey on PageRank computing*, Internet Math., 2 (2005), pp. 73–120.

[3] A. BERMAN AND R. J. PLEMMONS, *Nonnegative Matrices in the Mathematical Sciences*, Classics Appl. Math. 9, SIAM, Philadelphia, 1994.

[4] M. BIANCHINI, M. GORI, AND F. SCARSELLI, *Inside PageRank*, ACM Transactions on Internet Technology, 5 (2005), pp. 92–128.

[5] C. BREZINSKI AND M. REDIVO-ZAGLIA, *The PageRank vector: Properties, computation, approximation, and acceleration*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 551–575.

[6] C. BREZINSKI, M. REDIVO-ZAGLIA, AND S. SERRA-CAPIZZANO, *Extrapolation methods for PageRank computations*, C. R. Math. Acad. Sci. Paris, 340 (2005), pp. 393–397.

[7] S. BRIN AND L. PAGE, *The anatomy of a large-scale hypertextual web search engine*, Comput. Networks and ISDN Systems, 30 (1998), pp. 107–117.

[8] A. Z. BRODER, R. LEMPEL, F. MAGHOUL, AND J. PEDERSEN, *Efficient PageRank approximation via graph aggregation*, in Proceedings of the Thirteenth International World Wide Web Conference (WWW2004), ACM Press, New York, 2004, pp. 484–485.

[9] T. DAYAR AND W. J. STEWART, *Quasi lumpability, lower-bounding coupling matrices, and nearly completely decomposable Markov chains*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 482–498.

[10] G. M. DEL CORSO, A. GULLÍ, AND F. ROMANI, *Fast PageRank computation via a sparse linear system*, Internet Math., 2 (2005), pp. 251–273. Available online at http://www.internetmathematics.org/volumes/2/3/DelCorso.pdf.

[11] N. Eiron, K. S. McCurley, and J. A. Tomlin, *Ranking the web frontier*, in Proceedings of the Thirteenth International World Wide Web Conference (WWW2004), ACM Press, New York, 2004, pp. 309–318.
[12] L. Eldén, *The Eigenvalues of the Google Matrix*, Technical report LiTH-MAT-R-04-01, Department of Mathematics, Linköping University, Linköping, Sweden, 2004.
[13] D. Gleich, L. Zhukov, and P. Berkhin, *Fast parallel PageRank: A Linear System Approach*, Technical report, Yahoo!, Sunnyvale, CA, 2004.
[14] G. H. Golub and C. Greif, *An Arnoldi-type algorithm for computing PageRank*, BIT, 46 (2006), pp. 759–771.
[15] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, Baltimore, 1996.
[16] A. Gulli and A. Signorini, *The indexable web is more than* 11.5 *billion pages*, in Proceedings of the Fourteenth International World Wide Web Conference (WWW2005), ACM Press, New York, 2005, pp. 902–903.
[17] L. Gurvits and J. Ledoux, *Markov property for a function of a Markov chain: A linear algebra approach*, Linear Algebra Appl., 404 (2005), pp. 85–117.
[18] Z. Gyöngyi, H. Garcia-Molina, and P. J., *Combating web spam with TrustRank*, in Proceedings of the Thirtieth VLDB Conference, ACM Press, New York, 2004, pp. 576–587.
[19] T. H. Haveliwala and S. D. Kamvar, *The Second Eigenvalue of the Google Matrix*, Technical report, Computer Science Department, Stanford University, Palo Alto, CA, 2003.
[20] T. H. Haveliwala, S. D. Kamvar, D. Klein, C. D. Manning, and G. H. Golub, *Computing PageRank Using Power Extrapolation*, Technical report 2003-45, Stanford University, Palo Alto, CA, 2003. Available online at http://dbpubs.stanford.edu/pub/2003-45.
[21] R. A. Horn and S. Serra-Capizzano, *A general setting for the parametric Google matrix*, Internet Math., to appear.
[22] I. C. F. Ipsen and S. Kirkland, *Convergence analysis of a PageRank updating algorithm by Langville and Meyer*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 952–967.
[23] I. C. F. Ipsen and R. S. Wills, *Mathematical properties and analysis of Google's PageRank*, Bol. Soc. Esp. Mat. Apl., 34 (2006), pp. 191–196.
[24] R. W. Jernigan and R. H. Baran, *Testing lumpability in Markov chains*, Statist. Probab. Lett., 64 (2003), pp. 17–23.
[25] S. D. Kamvar, T. H. Haveliwala, and G. H. Golub, *Adaptive methods for the computation of PageRank*, Linear Algebra Appl., 386 (2004), pp. 51–65.
[26] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, *Extrapolation methods for accelerating PageRank computations*, in Proceedings of the Twelfth International World Wide Web Conference (WWW2003), Toronto, ACM Press, New York, 2003, pp. 261–270.
[27] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*, Van Nostrand Co., Princeton, NJ, 1960.
[28] A. N. Langville and C. D. Meyer, *Deeper inside PageRank*, Internet Math., 1 (2004), pp. 335–380. Available online at http://www.internetmathematics.org/volumes/1/3/Langville.pdf.
[29] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, NJ, 2006.
[30] A. N. Langville and C. D. Meyer, *A reordering for the PageRank problem*, SIAM J. Sci. Comput., 27 (2006), pp. 2112–2120.
[31] A. N. Langville and C. D. Meyer, *Updating Markov chains with an eye on Google's PageRank*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 968–987.
[32] C. P. Lee, G. H. Golub, and S. A. Zenios, *A Fast Two-Stage Algorithm for Computing PageRank and Its Extensions*, Technical report, Stanford University, Palo Alto, CA, 2003.
[33] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, 1999. Available online at http://dbpubs.stanford.edu/pub/1999-66.
[34] S. Serra-Capizzano, *Jordan canonical form of the Google matrix: A potential contribution to the PageRank computation*, SIAM J. Matrix Anal. Appl., 27 (2005), pp. 305–312.
[35] R. S. Wills, *Google's PageRank: The math behind the search engine*, Math. Intelligencer, 28 (2006), pp. 6–11.