SIAM J. MATRIX ANAL. APPL.
Vol. 30, No. 4, pp. 1677–1696

ORDINAL RANKING FOR GOOGLE'S PAGERANK*

REBECCA S. WILLS† AND ILSE C. F. IPSEN‡

Abstract. We present computationally efficient criteria that can guarantee correct ordinal ranking of Google's PageRank scores when they are computed with the power method (ordinal ranking of a list consists of assigning an ordinal number to each item in the list). We discuss the tightness of the ranking criteria, and illustrate their effectiveness for top k and bucket ranking. We present a careful implementation of the power method, combined with a roundoff error analysis that is valid for matrix dimensions $n < 10^{14}$. To first order, the roundoff error depends neither on n nor on the iteration count, but only on the maximal number of inlinks and the dangling nodes. The applicability of our ranking criterion is limited by the roundoff error from a single matrix vector multiply. Numerical experiments suggest that our criteria can effectively rank the top PageRank scores. We also discuss how to implement ranking for extremely large practical problems, by curbing roundoff error, reducing the matrix dimension, and using faster converging methods.

Key words. ranking distance, power method, stochastic matrix, PageRank, Google matrix, ordinal rank, roundoff error

AMS subject classifications. 62F07, 65F15, 65F50, 65C40, 15A18, 15A51, 68P20

DOI. 10.1137/070698129

1. Introduction.

Google founders Larry Page and Sergey Brin developed the PageRank algorithm primarily for ranking Web pages. In addition to its continued use in many Google Web search tools [1], the PageRank algorithm reaches beyond the Web to many other applications involving directed graphs such as social networks and semantic networks [2, 9, 11, 14, 19, 30, 33, 37, 45], as well as genomics [34], and identifying sources of hospital infections [41]. In fact, Hilgers and Langville recently identified the PageRank algorithm as one of the five greatest applications of Markov Chains [44].

The Google PageRank vector π is the stationary distribution of a $n \times n$ stochastic matrix G

$$\pi^T G = \pi^T, \qquad \pi \geq 0, \qquad \|\pi\|_1 = 1. \tag{G}$$

Each component of π measures the importance of a web page [8, 36]. If $\pi_i > \pi_j$, then web page i has higher PageRank than web page j, and page i may be displayed ahead of page j among the search results.

The matrix G is a convex combination of two stochastic matrices, $G \equiv \alpha S + (1 - \alpha)\mathbf{1}v^T$. Here $0 < \alpha < 1$ is a scalar, which was originally set to .85 [8, section 2.1.1]; S is an $n \times n$ stochastic matrix; $\mathbf{1}$ is the column vector of all ones; and $v \geq 0$ is a column vector with $\|v\|_1 = 1$. Because $\alpha < 1$ and $\mathbf{1}$ is a right eigenvector of S, the eigenvalue one of G is algebraically simple [13, 20, 40], which implies that π is unique.

Row i of S contains the outgoing links from web page i to other pages, while column i contains the incoming links from other pages to page i. A web page without

*Received by the editors July 24, 2007; accepted for publication (in revised form) by N. J. Higham September 18, 2008; published electronically January 23, 2009.
http://www.siam.org/journals/simax/30-4/69812.html

†Department of Mathematics, Computer Science and Physics, Roanoke College, 221 College Lane, Salem, VA 24153 (wills@roanoke.edu, http://faculty.roanoke.edu/wills/).

‡Department of Mathematics, North Carolina State University, P.O. Box 8205, Raleigh, NC 27695-8205 (ipsen@ncsu.edu, http://www4.ncsu.edu/~ipsen/).

any outgoing links (such as a pdf, image, or audio file) is called a *dangling node*. Zero rows corresponding to dangling nodes are replaced by a *dangling node vector* that has nonnegative elements summing to one.

In their 1998 paper [36, section 2.6] Google founders Brin and Page computed PageRank with the power method.

Power method applied to G. Let $x^{(0)} \geq 0$ with $\|x^{(0)}\|_1 = 1$. Repeat

$$[x^{(k+1)}]^T = [x^{(k)}]^T G, \qquad k \geq 0, \tag{P}$$

until some termination criterion is satisfied. Although numerous, possibly faster, methods have been proposed since 1998 [4, 31, 32], the power method retains many advantages:

1. It is simple to implement, especially in a parallel computing environment [17].
2. It requires a minimum of storage.
3. It has a robust and predictable convergence behavior that is insensitive to changes in the matrix. The convergence rate depends only on α, and is not sensitive to the underlying Web graph represented by S, the personalization vector v, the dangling node vector, or the starting vector $x^{(0)}$ [17].
4. It is numerically stable. All operations are numerically well conditioned. If $1 - \alpha$ is precomputed, then no subtractions are necessary to compute the iterates of (P). There is no danger of overflow since, in exact arithmetic, $\|x^{(k)}\|_1 = 1$.

The power method (P) as well as most iterative methods for computing PageRank compare successive iterates, based on their geometric distance or ranking distance. Once such a distance is small enough, the most recent iterate is judged to be a sufficiently good approximation to π. But the question is: Is the induced ranking of the iterate correct?

Overview. In section 2 we answer the previous question with "no." In section 3 we present the main idea of our paper, a ranking criterion for the elements of an iterate $x^{(k)}$:

If $x_i^{(k)} > x_j^{(k)} + \beta$, then $\pi_i > \pi_j$.

Here β is an upper bound for the error $\|x^{(k)} - \pi\|_1$. We show how this criterion can be used for exact, top k, and bucket ranking. In section 4 we derive several classes of computationally efficient bounds β based on geometric distances between iterates. In section 5 we present a careful implementation of the power method and derive roundoff error bounds for the iterates $x^{(k)}$. Numerical experiments in section 6 demonstrate that our criterion can identify the ranking of the top PageRank scores. We conclude with a discussion of how to solve extremely large problems in section 7.

Notation. All matrices are $n \times n$ matrices, and all vectors are $n \times 1$ column vectors. The $n \times n$ identity matrix is I, with ith column e_i. The transpose of a vector v is v^T, the elements of v are v_i, and inequalities like $v \geq 0$ and $|v| \geq 0$ are to be interpreted componentwise. The one norm $\|\cdot\|_1$ is the maximal column sum, and the infinity norm $\|\cdot\|_\infty$ the maximal row sum. In particular, if x is a column vector, then

$$\|x\|_1 = \sum_i |x_i| = \|x^T\|_\infty, \qquad \|x\|_\infty = \max_i |x_i| = \|x^T\|_1.$$

2. Existing termination criteria do not produce correct rankings. The two most frequently mentioned termination criteria for PageRank computation are based on geometric distance and ranking distance [6, section 4.2.1].

Geometric distance. The traditional convergence criterion terminates the power method once some norm of the residual is sufficiently small. Since the power method iteration (P) contains no normalization, the residual equals the difference between successive iterates, $[x^{(k)}]^T - [x^{(k)}]^T G = [x^{(k)} - x^{(k+1)}]^T$. The power method (P) is terminated once $\|x^{(k)} - x^{(k+1)}\|$ is sufficiently small in the one, two, or infinity norms. The residual norm of (P) can be interpreted as a distance between two vectors, and has thus been classified as a *geometric distance* [6, section 4.2.1], in contrast to the *ranking distance* below.

Ranking distance. We specify the position of an element in an ordered list through its *ordinal rank*, which is defined below first for a single element and then for a whole vector (the ordinal rank bears no relation to the numerical rank).

DEFINITION 2.1 (ordinal rank). *Let* $x = \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix}^T$ *be a real vector, and* σ *a permutation that orders the elements of* x *in decreasing order,* $x_{\sigma(1)} \geq \dots \geq x_{\sigma(n)}$. *Then the ordinal rank of an individual element is* $\mathrm{Orank}(x_i) \equiv \sigma(i)$, $1 \leq i \leq n$, *and the ordinal rank of the whole vector is* $\mathrm{Orank}(x) \equiv \begin{pmatrix} \mathrm{Orank}(x_1) & \dots & \mathrm{Orank}(x_n) \end{pmatrix}$.

If the elements of x are distinct, and if $x_j = \max_i x_i$, then $\mathrm{Orank}(x_j) = 1$; and if $x_j = \min_i x_i$, then $\mathrm{Orank}(x_j) = n$. If x contains identical elements, then the ordinal ranking is not unique, because the permutation σ in Definition 2.1 is not unique and is not required to preserve the relative order of the elements (i.e., σ does not have to be stable in the sense of sorting). In contrast to other ranking schemes, no two items receive the same ordinal rank, even when they are equal. The concept of ordinal rank leads to the particular ranking distance below, which is known as *Kendall's* τ *distance* [6, section 4.2.2] [27, section 1.13].

DEFINITION 2.2 (ranking distance). *The ranking distance between real vectors* x *and* y, *each with distinct elements, is* $\sum_i \sum_j \delta_{xy}(i,j)$, *where*

$$\delta_{xy}(i,j) \equiv \begin{cases} 1 & \mathrm{Orank}(x_i) < \mathrm{Orank}(x_j) \text{ and } \mathrm{Orank}(y_i) > \mathrm{Orank}(y_j) \\ 0 & \text{otherwise} \end{cases}.$$

Ranking distances have been used as termination criteria in iterative methods for computing PageRank [3, 10, 25, 28, 38, 39, 46]. Because the matrix G is large, computing the ranking distance for entire vectors is too expensive. To reduce computation time, one can focus on the top k rankings [15, 25, 26]. Experiments in [26] suggest that termination based on the top k rankings tends to produce rankings that resemble those produced by a termination criterion based on the one norm of the residual.

Incorrect ranking. Simple examples, such as those described in [47, section 4.2] for the directed ring[1] graph with n vertices, illustrate that geometric and ranking distances between successive iterates of the power method (P) can fail to produce correct rankings. In addition, the examples demonstrate that (1) correct ranking can be achieved in some iteration and destroyed in the next, (2) a small residual norm does not guarantee correct ranking, (3) zero ranking distance between successive iterates does not guarantee correct ranking, and (4) successive iterates can be correctly ranked before the residual norm is small.

3. Ranking. Since geometric and ranking distances between successive iterates of the power method (P) do not ensure correct ranking, we consider instead the

[1] In a directed ring graph with n vertices, vertex i links to vertex $i+1$, $1 \leq i \leq n-1$, and vertex n links back to vertex 1.

ranking distance between an iterate $x^{(k)}$ and the desired PageRank vector π. We obtain information about the ranking distance from the geometric distance, and show how the resulting ranking criterion can be used to perform exact, top k, and bucket ranking.

We use a consequence of an inequality from [21, Corollary 2.4(a)] which relates the one norm and infinity norm of a vector whose elements sum to zero: If y is a column vector with $y^T\mathbf{1} = 0$, then

$$\|y\|_\infty \leq \|y\|_1/2. \tag{3.1}$$

The main idea of our paper is to gather information about relative ranking based on an approach by Kirkland [29, Corollaries 3.9-3.12], which we present below in a more general context.

THEOREM 3.1 (ranking criterion). *Let $x \geq 0$ with $\|x\|_1 = 1$ be an approximation to π in* (G)*, and $\beta \geq \|x - \pi\|_1$. If $x_i > x_j + \beta$, then $\pi_i > \pi_j$.*

Proof. If $x = \pi$, then $\beta = 0$ and the result holds trivially. Assuming $\beta > 0$ gives $(x_i - \pi_i) - (x_j - \pi_j) \leq |x_i - \pi_i| + |x_j - \pi_j| \leq 2\|x - \pi\|_\infty$. Since $(x - \pi)^T\mathbf{1} = 0$, (3.1) implies $\|x - \pi\|_\infty \leq \|x - \pi\|_1/2$. Hence $(x_i - \pi_i) - (x_j - \pi_j) \leq \|x - \pi\|_1 \leq \beta$, and $x_i - (x_j + \beta) \leq \pi_i - \pi_j$. Therefore, $0 < x_i - (x_j + \beta)$ implies $0 < \pi_i - \pi_j$. ∎

Consequently, if two elements of $x^{(k)}$ are well-separated (compared to the geometric distance between $x^{(k)}$ and π), then we can say something about the relative rankings of the corresponding PageRank scores. Because the ranking criterion in Theorem 3.1 applies only to well-separated elements, it can, in general, determine only a partial ranking of the PageRank scores.

Kirkland [29, section 3] expresses the quantity β in Theorem 3.1 as a function of the lengths of shortest cycles on which vertices i and j are situated in the graph of S. However, it is not clear how to efficiently compute shortest cycle lengths for all vertices.

Exact, top k, and bucket ranking. The bucket ranking criteria below are motivated by Google's Toolbar PageRank scores, which are integers from 0 (low) to 10 (high). Our ranking criteria determine a topological (or partial) order for the PageRank scores.

Let $x \geq 0$ with $\|x\|_1 = 1$ be an approximation to π in (G) and $\beta \geq \|x - \pi\|_1$. Let Q be a permutation that orders the elements of x in decreasing magnitude, i.e.,

$$\tilde{x} \equiv Qx = \begin{pmatrix} \tilde{x}_1 & \dots & \tilde{x}_n \end{pmatrix}^T, \qquad \tilde{x}_1 \geq \ldots \geq \tilde{x}_n, \qquad \tilde{\pi} \equiv Q\pi = \begin{pmatrix} \tilde{\pi}_1 & \dots & \tilde{\pi}_n \end{pmatrix}^T.$$

In contrast to the elements of $\tilde{x}$, those of $\tilde{\pi}$ are, in general, not ordered in decreasing magnitude.

First we show that if element k of $\tilde{x}$ is well separated from element $k + 1$, then elements $1, \ldots, k$ approximate the top k PageRank scores.

LEMMA 3.2 (top k). *If $\tilde{x}_k > \tilde{x}_{k+1} + \beta$, then* $\text{Orank}(\tilde{\pi}_i) \leq k$ *for $1 \leq i \leq k$, and* $\text{Orank}(\tilde{\pi}_j) \geq k + 1$ *for $k + 1 \leq j \leq n$.*

Proof. From $\tilde{x}_k > \tilde{x}_{k+1} + \beta$ follows $\tilde{\pi}_k > \tilde{\pi}_{k+1}$, according to Theorem 3.1. The descending ordering implies $\tilde{x}_k > \tilde{x}_{k+1} + \beta \geq \ldots \geq \tilde{x}_n + \beta$, so that $\tilde{\pi}_k > \tilde{\pi}_{k+1}, \ldots, \tilde{\pi}_n$. The descending ordering also implies $\tilde{x}_1 \geq \ldots \geq \tilde{x}_k > \tilde{x}_{k+1} + \beta$, so that $\tilde{\pi}_1, \ldots, \tilde{\pi}_k > \tilde{\pi}_{k+1}$. Combining the two sets of inequalities yields $\tilde{\pi}_1, \ldots, \tilde{\pi}_k > \tilde{\pi}_{k+1}, \ldots, \tilde{\pi}_n$. Therefore $\text{Orank}(\tilde{\pi}_i) \leq k$ for $1 \leq i \leq k$, and $\text{Orank}(\tilde{\pi}_j) \geq k + 1$ for $k + 1 \leq j \leq n$. ∎

Now we present a criterion for finding the "exact" rank (here "exact" does not refer to finite precision accuracy but to the fact that we can assign a number, rather

than an interval, to the rank). If element $k+1$ of $\tilde{x}$ is well separated from elements k and $k+2$, then element $k+1$ of $\tilde{x}$ approximates the $(k+1)$st PageRank score.

LEMMA 3.3 (exact rank). *If* $\tilde{x}_k > \tilde{x}_{k+1} + \beta$ *and* $\tilde{x}_{k+1} > \tilde{x}_{k+2} + \beta$, *then* $\mathrm{Orank}(\tilde{\pi}_{k+1}) = k+1$.

Proof. This follows from Lemma 3.2. Condition $\tilde{x}_k > \tilde{x}_{k+1}+\beta$ implies $\tilde{\pi}_1, \ldots, \tilde{\pi}_k > \tilde{\pi}_{k+1}$; hence $\mathrm{Orank}(\tilde{\pi}_{k+1}) \geq k+1$. Condition $\tilde{x}_{k+1} > \tilde{x}_{k+2} + \beta$ implies $\tilde{\pi}_{k+1} > \tilde{\pi}_{k+2}, \ldots, \tilde{\pi}_n$; hence $\mathrm{Orank}(\tilde{\pi}_{k+1}) \leq k+1$. ∎

Often it is not possible to determine the exact PageRank, but we can still try to assign PageRank scores to a "bucket". This is done in the next lemma, which represents an extension of Lemma 3.3 to intervals.

LEMMA 3.4 (bucket). *If* $\tilde{x}_k > \tilde{x}_{k+i} + \beta$ *and* $\tilde{x}_{k+i} > \tilde{x}_{k+i+j} + \beta$ *for* $i, j \geq 1$ *then* $k+1 \leq \mathrm{Orank}(\tilde{\pi}_{k+i}) \leq k+i+j-1$.

Proof. This follows from Lemma 3.2. The condition $\tilde{x}_k > \tilde{x}_{k+i} + \beta$ implies $\tilde{\pi}_1, \ldots, \tilde{\pi}_k > \tilde{\pi}_{k+i}$, so that $\mathrm{Orank}(\tilde{\pi}_{k+i}) \geq k+1$. The condition $\tilde{x}_{k+i} > \tilde{x}_{k+i+j} + \beta$ implies $\tilde{\pi}_{k+i} > \tilde{\pi}_{k+i+j}, \ldots, \tilde{\pi}_n$ so that $\mathrm{Orank}(\tilde{\pi}_{k+i}) \leq k+i+j-1$. ∎

Lemma 3.4 assigns PageRank score $\tilde{x}_{k+i}$ to a bucket that represents ranks $k+1, \ldots, k+i+j-1$. If $i = j = 1$ then the bucket consists of a single number and Lemma 3.4 reduces to Lemma 3.3. The top k ranking in Lemma 3.2 is a special case of bucket ranking with two buckets: One for ranks $1, \ldots, k$ and another one for ranks $k+1, \ldots, n$. In contrast to bucket sorting the buckets are not specified beforehand; they emerge during the execution of the power method (P).

4. Error bounds for the power method. We present four classes of computable bounds for the error $\|x^{(k)} - \pi\|_1$ and for the quantity β used in the ranking criteria in section 3. The four classes consist of the following types of bounds: simple (section 4.1), backward looking (section 4.2), forward looking (section 4.3), and two-level forward looking (section 4.4). We use the following facts for stochastic matrices S: $\|S^i\|_\infty = 1$, $i \geq 0$, and

$$\|S^i(I - \alpha^j S^j)^{-1}\|_\infty = \frac{1}{1-\alpha^j}, \qquad i \geq 0, \quad j \geq 1. \tag{4.1}$$

The last expression follows from the fact that $(I - \alpha^j S^j)^{-1}$ is nonnegative, and $\|(I - \alpha^j S^j)^{-1}\|_\infty = (1-\alpha^j)^{-1}$ [32, section 7.1].

First we justify why the ranking of the iterates in (P) can converge under a criterion like the one in Theorem 3.1. The inequalities below relate two corresponding components of two different iterates and show that the distance between the components changes less and less as the iterations progress.

THEOREM 4.1 (component distances stabilize).

$$|x_i^{(k-1)} - x_j^{(k-1)}| - \alpha^{k-1}\gamma \leq |x_i^{(k)} - x_j^{(k)}| \leq |x_i^{(k-1)} - x_j^{(k-1)}| + \alpha^{k-1}\gamma, \qquad k \geq 1,$$

where $\gamma \equiv 2\|x^{(1)} - x^{(0)}\|_1$.

Proof. As in [7, Property 7] one shows $[x^{(k)}]^T = \alpha^k [x^{(0)}]^T S^k + (1-\alpha)\sum_{l=0}^{k-1} \alpha^l v^T S^l$. Hence the difference between two components equals

$$\begin{aligned} x_i^{(k)} - x_j^{(k)} &= \alpha^k \left[x^{(0)}\right]^T S^k (e_i - e_j) + (1-\alpha)\sum_{l=0}^{k-1} \alpha^l v^T S^l (e_i - e_j) \\ &= \left(x_i^{(k-1)} - x_j^{(k-1)}\right) + \alpha^{k-1}\left[x^{(1)} - x^{(0)}\right]^T S^{k-1}(e_i - e_j). \end{aligned}$$

The Hölder inequality and (4.1) imply

$$\left|[x^{(1)} - x^{(0)}]^T S^{k-1}(e_i - e_j)\right| \leq \|x^{(1)} - x^{(0)}\|_1 \|S^{k-1}(e_i - e_j)\|_\infty \leq 2\|x^{(1)} - x^{(0)}\|_1. \quad \square$$

The idea behind the bounds for $\|x^{(k)} - \pi\|_1$ is to extend the residual, which is a difference of successive iterates $[x^{(k)}]^T - [x^{(k)}]^T G = [x^{(k)} - x^{(k+1)}]^T$, to differences between nonsuccessive iterates $[x^{(k-j)} - x^{(k)}]^T$. The derivations in subsequent sections are based on the following recursions.

LEMMA 4.2 (recursions).

$$\begin{aligned} \text{Error :} \qquad & [x^{(k+1)} - \pi]^T = \alpha[x^{(k)} - \pi]^T S, \qquad k \geq 0 \\ \text{Iterate difference :} \quad & [x^{(k-j+1)} - x^{(k+1)}]^T = \alpha[x^{(k-j)} - x^{(k)}]^T S, \qquad 1 \leq j \leq k. \end{aligned}$$

Proof. The recursions follow from $[x^{(k+1)}]^T = [x^{(k)}]^T G$, $G = \alpha S + (1-\alpha)\mathbf{1}v^T$, and $[x^{(k)} - \pi]^T \mathbf{1} = 0$.

Note that the statements also follow from the properties of splitting methods [43, section 3.6]. This is because π is the solution of the linear system $\pi^T(I - \alpha S) = (1-\alpha)v^T$, and the power method $[x^{(k+1)}]^T = [x^{(k)}]^T G$ is mathematically equivalent to a splitting method $[x^{(k+1)}]^T M = [x^{(k)}]^T N + (1-\alpha)v^T$, where $M = I$ and $N = \alpha S$. $\square$

The second recursion in Lemma 4.2 is an extension of the one derived for $j = 1$ in [29, Proof of Corollary 3.11].

4.1. Simple bound. The recursions in Lemma 4.2 lead immediately to a simple normwise error bound that depends only on α and k.

THEOREM 4.3 (simple bound). $\|x^{(k)} - \pi\|_1 \leq \alpha^k \|x^{(0)} - \pi\|_1 \leq 2\alpha^k$, $k \geq 0$.

Proof. Lemma 4.2 implies $[x^{(k)} - \pi]^T = \alpha^k [x^{(0)} - \pi]^T S^k$, and (4.1) implies

$$\|x^{(k)} - \pi\|_1 = \|[x^{(k)} - \pi]^T\|_\infty \leq \alpha^k \|[x^{(0)} - \pi]^T\|_\infty \|S^k\|_\infty = \alpha^k \|x^{(0)} - \pi\|_1.$$

The second upper bound follows from the triangle inequality and $\|x^{(0)}\|_\infty = \|\pi\|_\infty = 1$. $\square$

The bounds in Theorem 4.3 first appeared in [5, Theorem 5.1] and [24, section 4]. In the special case $x^{(0)} = v$, when the starting vector equals the personalization vector, the power of α can be increased by one, $\|x^{(k)} - \pi\|_1 \leq \alpha^{k+1}\|v - S^T\pi\|_1 \leq 2\alpha^{k+1}$; see also [7, Property 9]. The simple bound in Theorem 4.3 can be used as a ranking criterion in the power method (P) as follows.

COROLLARY 4.4 (ranking with the simple bound). *If* $x_i^{(k)} > x_j^{(k)} + 2\alpha^k$, $k \geq 1$, *then* $\pi_i > \pi_j$.

Proof. Follows from Theorem 4.3 and setting $\beta = 2\alpha^k$ in Theorem 3.1. $\square$

4.2. Backward-looking bounds. Backward-looking bounds are constructed from previous iterates.

THEOREM 4.5 (backward-looking bounds). $\|x^{(k)} - \pi\|_1 \leq \frac{\alpha^j}{1-\alpha^j}\|x^{(k-j)} - x^{(k)}\|_1$, $1 \leq j \leq k$.

Proof. Lemma 4.2 implies

$$\begin{aligned} \alpha^j [x^{(k-j)} - x^{(k)}]^T S^j &= [x^{(k)} - x^{(k+j)}]^T = [x^{(k)} - \pi]^T - [x^{(k+j)} - \pi]^T \\ &= [x^{(k)} - \pi]^T - \alpha^j [x^{(k)} - \pi]^T S^j = [x^{(k)} - \pi]^T (I - \alpha^j S^j). \end{aligned}$$

Hence $\left[x^{(k)} - \pi\right]^T = \alpha^j \left[x^{(k-j)} - x^{(k)}\right]^T S^j (I - \alpha^j S^j)^{-1}$. Take norms

$$\|x^{(k)} - \pi\|_1 = \|[x^{(k)} - \pi]^T\|_\infty \leq \alpha^j \|[x^{(k-j)} - x^{(k)}]^T\|_\infty \|S^j (I - \alpha^j S^j)^{-1}\|_\infty$$

and use (4.1). $\square$

The bound for $j = 1$ in Theorem 4.5 was already derived in [29, (3.1)] for general approximations, not limited to just power method iterates. Note that for a fixed step length j, a full backward look is not possible in early iterations as long as $k < j$. Experiments indicate that no bound is always the tightest. We can distinguish two types of backward-looking bounds: Those where j is fixed and those where j is a function of k. Among the fixed step bounds, the $j = 1$ bound is preferable for several reasons: It tends to be competitive in the long-term since $\|x^{(k)} - \pi\|_1 \leq \alpha\|x^{(k-1)} - \pi\|_1 \leq \alpha^2\|x^{(k-2)} - \pi\|_1 \leq \ldots$. It takes effect immediately starting with iteration 1 – in contrast to other bounds which require a startup of j iterations before the full backward look is possible. It performs well in our experiments in later iterations. At last, it keeps storage requirements low, because bounds with fixed j need to store j or more vectors. Among the bounds where j is a function of k, the simplest one is $j = k$,

$$\|x^{(k)} - \pi\|_1 \leq \frac{\alpha^k}{1-\alpha^k}\|x^{(0)} - x^{(k)}\|_1.$$

This bound has low storage requirements as well, it is effective immediately, starting with iteration 1, and in our experiments it tends to do well in early iterations. However, the $j = k$ bound does not do as well in later iterations. This is because it depends on the initial error $\|x^{(0)} - \pi\|_1$ which remains constant throughout the iterations, while for bounds with a fixed step length j, the error $\|x^{(k-j)} - \pi\|_1$ approaches zero as k increases. Applied to the power method (P), the backward-looking bounds in Theorem 4.5 can be used for ranking as follows.

COROLLARY 4.6 (ranking with backward-looking bounds). *If* $x_i^{(k)} > x_j^{(k)} + \frac{\alpha^l}{1-\alpha^l}\|x^{(k-l)} - x^{(k)}\|_1$, $1 \leq l \leq k$, *then* $\pi_i > \pi_j$.

4.3. Forward-looking bounds. Forward-looking bounds are constructed from future iterates. The derivation is similar to the one in Theorem 4.5.

THEOREM 4.7 (forward-looking bounds). $\|x^{(k)} - \pi\|_1 \leq \frac{\|x^{(k+j)} - x^{(k)}\|_1}{1-\alpha^j}$, $k \geq 0$, $j \geq 1$.

The forward-looking bound for $j = 1$ was derived in [7, Property 12]. Looking farther ahead can lead to better estimates for the error at the current iteration k. This is to be expected because future iterates can be more accurate. Comparing the bounds in Theorem 4.7 for $j = 1$ and $j > 1$ shows that looking several steps ahead can result in tighter bounds than just looking a single step ahead,

$$\frac{\|x^{(k+j)} - x^{(k)}\|_1}{1-\alpha^j} \leq \frac{\|x^{(k+1)} - x^{(k)}\|_1}{1-\alpha}, \qquad k \geq 0, \quad j \geq 1.$$

Applied to the power method (P), the forward-looking bounds in Theorem 4.7 can be used for ranking as follows.

COROLLARY 4.8 (ranking with forward-looking bounds). *If* $x_i^{(k)} > x_j^{(k)} + \frac{\|x^{(k+l)} - x^{(k)}\|_1}{1-\alpha^l}$, $k \geq 0$, $l \geq 1$, *then* $\pi_i > \pi_j$.

4.4. Two-level, forward-looking bounds. Another type of forward bound looks forward in two stages.

THEOREM 4.9 (two-level, forward-looking bounds).

$$\|x^{(k)} - \pi\|_1 \leq \|x^{(k+j)} - x^{(k)}\|_1 + \frac{\|x^{(k+j+i)} - x^{(k+j)}\|_1}{1-\alpha^i}, \qquad k \geq 0, \quad j, i \geq 1.$$

COROLLARY 4.10 (ranking with two-level, forward-looking bounds). *If*

$$x_i^{(k)} > x_j^{(k)} + \|x^{(k+l)} - x^{(k)}\|_1 + \frac{\|x^{(k+l+h)} - x^{(k+l)}\|_1}{1-\alpha^h}, \qquad k \geq 0, \quad l,h \geq 1,$$

then $\pi_i > \pi_j$.

5. Finite precision computation. We present error bounds for perturbed power method iterates in section 5.1, a floating point implementation of the power method in section 5.2, and bounds for ranking in floating point arithmetic in section 5.3.

5.1. Perturbation bounds. In a finite precision context, the ranking criterion in Theorem 3.1 must be applied to perturbed power method iterates $\hat{x}^{(k)}$. That is, if $\hat{x}_i^{(k)} > \hat{x}_j^{(k)} + \|\hat{x}^{(k)} - \pi\|_1$, then $\pi_i > \pi_j$. We assume that the perturbed iterates are nonnegative, have unit norm, and incur an error during each iteration. The error bounds for $\|\hat{x}^{(k)} - \pi\|_1$ below are simple and easy to compute.

THEOREM 5.1 (finite precision error bounds). *Let* $\hat{x}^{(0)} = x^{(0)}$, *and* $[\hat{x}^{(k+1)}]^T = [\hat{x}^{(k)}]^T G + g_k^T$, $k \geq 0$, *be such that* $\hat{x}^{(k)} \geq 0$ *and* $\|\hat{x}^{(k)}\|_1 = 1$, $k \geq 0$.

1. *Simple bound:*

$$\|\hat{x}^{(k)} - \pi\|_1 \leq 2\alpha^k + \frac{1-\alpha^k}{1-\alpha} \max_{0 \leq i \leq k-1} \|g_{k-i}\|_1, \qquad k \geq 1.$$

2. *Backward-looking bounds:*

$$\|\hat{x}^{(k)} - \pi\|_1 \leq \frac{\alpha^j}{1-\alpha^j}\|\hat{x}^{(k-j)} - \hat{x}^{(k)}\|_1 + \frac{1-\alpha^j}{1-\alpha} \max_{0 \leq i \leq j-1} \|g_{k-i}\|_1, \qquad 0 \leq j < k.$$

3. *Forward-looking bounds:*

$$\|\hat{x}^{(k)} - \pi\|_1 \leq \frac{\|\hat{x}^{(k+j)} - \hat{x}^{(k)}\|_1}{1-\alpha^j} + \frac{1-\alpha^j}{1-\alpha} \max_{1 \leq i \leq j} \|g_{k+i}\|_1, \qquad k \geq 0, \quad j \geq 1.$$

4. *Two-level, forward-looking bounds:*

$$\|\hat{x}^{(k)} - \pi\|_1 \leq \|\hat{x}^{(k+j)} - \hat{x}^{(k)}\|_1 + \frac{\|\hat{x}^{(k+j+i)} - \hat{x}^{(k+j)}\|_1}{1-\alpha^i} + \frac{1-\alpha^i}{1-\alpha} \max_{1 \leq l \leq i} \|g_{k+j+l}\|_1,$$

where $k \geq 0$, $j,i \geq 1$.

Proof. First we derive an expression for the absolute error in the perturbed iterates. From $\hat{x}^{(k)} \geq 0$, $x^{(k)} \geq 0$, $\|\hat{x}^{(k)}\|_1 = \|x^{(k)}\|_1 = 1$ follows $g_k^T \mathbf{1} = f_k^T \mathbf{1} = 0$. With $f_0 \equiv 0$ this implies

$$\hat{x}^{(k)} = x^{(k)} + f_k, \qquad f_k^T \equiv \alpha^j f_{k-j}^T S^j + \sum_{i=0}^{j-1} \alpha^i g_{k-i}^T S^i, \qquad 1 \leq j \leq k. \tag{5.1}$$

We use (5.1) to derive each of the four finite precision bounds.

1. Simple bound: This follows from $[\hat{x}^{(k)} - \pi]^T = [x^{(k)} - \pi]^T + f_k^T$ and (5.1).
2. Backward-looking bounds: As in the proof of Theorem 4.5 one shows

$$[\hat{x}^{(k)} - \pi]^T (I - \alpha^j S^j) = \alpha^j [\hat{x}^{(k-j)} - \hat{x}^{(k)}]^T S^j + f_k^T - \alpha^j f_{k-j}^T S^j.$$

Then (5.1) implies $f_k^T - \alpha^j f_{k-j}^T S^j = \sum_{i=0}^{j-1} \alpha^i g_{k-i}^T S^i$.

3. Forward-looking bounds: As in the proof of Theorem 4.7 one shows

$$[\hat{x}^{(k)} - \pi]^T (I - \alpha^j S^j) = [\hat{x}^{(k)} - \hat{x}^{(k+j)}]^T + f_{k+j}^T - \alpha^j f_k^T S^j.$$

From (5.1) follows $f_{k+j}^T - \alpha^j f_k^T S^j = \sum_{i=0}^{j-1} \alpha^i g_{k+j-i}^T S^i$.

4. Two-level, forward-looking bounds: As in the proof of Theorem 4.9, one shows that $[\pi - \hat{x}^{(k)}]^T$ is equal to

$$[\hat{x}^{(k+j)} - \hat{x}^{(k)}]^T - [\hat{x}^{(k+j)} - \hat{x}^{(k+j+i)}]^T (I - \alpha^i S^i)^{-1} - f_{k+j}^T - [f_{k+j+i} - f_{k+j}]^T (I - \alpha^i S^i)^{-1}.$$

Hence (5.1) implies for the error term

$$-f_{k+j}^T - [f_{k+j+i} - f_{k+j}]^T (I - \alpha^i S^i)^{-1} = -\sum_{l=0}^{i-1} \alpha^l g_{k+j+i-l}^T S^l (I - \alpha^i S^i)^{-1}. \qquad \square$$

The term g_k takes care of finite precision errors incurred in iteration k, including those from matrix vector multiplication, as well as explicit normalization of the iterates if necessary. Theorem 5.1 shows that the bounds are affected only by the error in a single iteration, and do not suffer from accumulation of errors.

5.2. Power method implementation. We discuss the implementation of the power method in floating point arithmetic.

As already mentioned in section 1, the matrix S is derived from the webgraph and zero rows corresponding to dangling nodes (i.e., web pages without outlinks) are modified to ensure that S is stochastic. Computationally, though, it is more efficient to keep the webgraph part separated from the dangling node fix so that one can take advantage of the latter's low rank [32, section 8.1]. It turns out that this separation also limits accumulation of roundoff error in a matrix vector multiplication with S. Therefore it is necessary to discuss the construction of S in more detail.

The web graph is represented by a $n \times n$ *substochastic* matrix H. That is, the elements of H are nonnegative, and each row is either zero, or else its elements sum to one. The zero rows correspond to dangling nodes, which are web pages without outlinks. To obtain the stochastic matrix S, one can replace each zero row by the same dangling node vector w^T, where w is a column vector with $w \geq 0$ and $\|w\|_1 = 1$. The resulting Google matrix is $G = \alpha S + (1 - \alpha)\mathbf{1}v^T$, where $S = H + dw^T$ and d is a column vector of zeros and ones. An element of d is equal to 1 if the corresponding row in H is zero; otherwise this element of d is equal to zero. The following floating point implementation of the power method (P) exploits the fact that dw^T and $\mathbf{1}v^T$ have rank one.

Floating point implementation of (P). Let $\hat{x}^{(0)} \geq 0$ with $\|\hat{x}^{(0)}\|_1 = 1$, and $\alpha_1 \equiv 1 - \alpha$. Repeat

$$\text{(FP)} \qquad [y^{(k+1)}]^T := \text{fl}\left(\alpha([\hat{x}^{(k)}]^T H + ([\hat{x}^{(k)}]^T d)w^T) + \alpha_1 v^T\right)$$
$$\hat{x}^{(k+1)} := \text{fl}\left(y^{(k+1)}/\|y^{(k+1)}\|_1\right)$$

until some termination criterion is satisfied.

The explicit normalization of the iterates in (FP) is necessary to ensure that iterate norms remain close to unity in a finite precision environment. Figure 5.1 illustrates why this is necessary. The norms of the unnormalized iterates $y^{(k)}$ deviate much further from 1 than the norms of the normalized iterates $\hat{x}^{(k)}$. The ratios

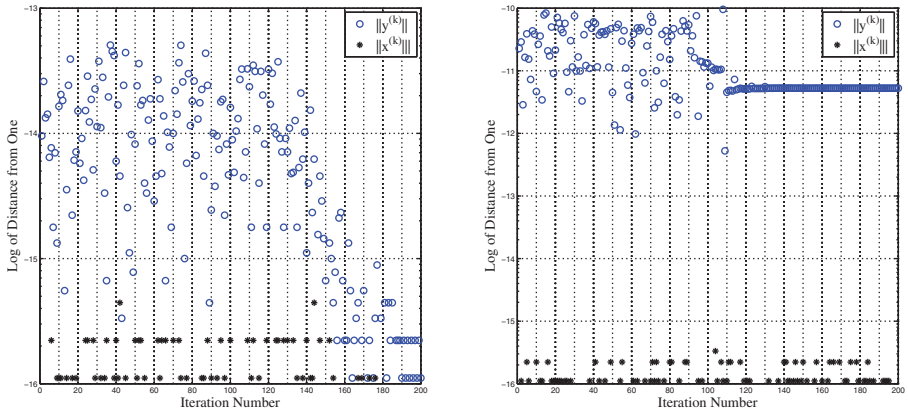

FIG. 5.1. $|1 - \|\hat{x}^{(k)}\|_1|$ *and* $|1 - \|y^{(k)}\|_1|$ *for iterates of the Power Method* (FP) *applied to the matrices* G_S *(left) and* G_L *(right) in section* 6.

$|1 - \|y^{(k)}\|_1|$ can approach 10^{-13} for the small matrix G_S and hover around 10^{-11} for the larger matrix G_L. The analysis in the proof of Theorem 5.2 explains this: In IEEE double precision the unit roundoff is $\epsilon \approx 10^{-16}$, so that the error in the matrix vector multiplication $[\hat{x}^{(k)}]^T G_L$ is about $m\epsilon \approx 4 \cdot 10^{-11}$, where $m = 168,685$ is the maximal number of nonzero elements in any column of H_L. For the smaller matrix G_S, the computation of $x^T d$ dominates, leading to an error of $\|d\|_1 \epsilon \approx 7 \cdot 10^{-12}$, since d has 2,861 elements equal to one. In contrast, the norms of the normalized iterates are almost perfect. In all iterations k the deviation $|\|\hat{x}^{(k)}\|_1 - 1|$ is either ϵ, $\epsilon/2$, or 0. Note that Figure 5.1 shows merely the effect of a single missing normalization; the accumulated damage from failing to normalize over many iterations is much worse.

5.3. Floating point bounds. We bound the roundoff error g_k incurred in iteration k of the power method (FP). Existing roundoff error bounds for the power method and stationary iterative methods [22, sections 17, 18], [42] do not seem to be applicable here, because they require knowledge of the condition number of a diagonalizing transformation, or assume that the spectral radius of the iteration matrix is strictly less than one.

We assume the standard model for the elementary floating point arithmetic operations with unit roundoff ϵ [22, section 2.2]. If a and b are floating point numbers, then

$$\mathrm{fl}(a \text{ op } b) = (a \text{ op } b)(1 + \delta), \qquad |\delta| \leq \epsilon, \quad \text{op} = +, -, *, /.$$

We exploit the fact that the iterations in (FP) execute no subtractions, and that all operations are well conditioned. The norms are computed by *compensated summation* [22, section 4.3], [35] so that the dominant part of the roundoff error $\|g_k\|_1$ does not depend on the matrix dimension n, but only on the sparsity of the matrix. The analysis below holds for matrices of order $n < 10^{14}$, in IEEE double precision with unit roundoff $\epsilon \equiv 10^{-16}$.

THEOREM 5.2 (floating point bounds). *Assume that* $n\epsilon < .01$, *and*

1. *The scalars* α *and* $\alpha_1 \equiv 1 - \alpha$, *and the elements of* H, v, w, *and* $\hat{x}^{(0)} = x^{(0)}$ *are floating point numbers.*
2. *The iterates* $\hat{x}^{(k)}$ *are computed according to* (FP).
3. *The norms* $\|y^{(k+1)}\|_1$ *in* (FP) *are computed by compensated summation.*

4. *H has at most m nonzeros per column, and $\|d\|_1$ zero rows.*
Then

$$\|g_{k+1}\|_1 \le \frac{2\epsilon(3.03 + c\alpha M)}{1 - \epsilon(3.03 + c\alpha M)} + \mathcal{O}(n\epsilon^2), \qquad k \ge 0$$

where $M \equiv \max\{m, \|d\|_1 + 1\}$, and $c \equiv 1.01(1 + 3.03\epsilon)$.

Proof. Abbreviate $x \equiv \hat{x}^{(k)}$, $y \equiv y^{(k+1)}$, and $\delta \equiv \|d\|_1 + 1$.

1. Since H has at most m nonzero elements per column, the elements of S and x are nonnegative, and $m\epsilon \le .01$ we obtain with [22, (3.2) and Lemma 3.4] $\mathrm{fl}(x^T H) = x^T H + h_1^T$, where $|h_1^T| \le 1.01m\epsilon\,(x^T H)$.

2. Similarly, since α and the elements of d and x are nonnegative, we obtain for the second summand in $\alpha x^T S$, $\mathrm{fl}(\alpha x^T d) = \alpha x^T d + h_2$, where $|h_2| \le 1.01\delta\epsilon\,(\alpha x^T d)$.

3. For the computation of $y^T = \mathrm{fl}(x^T G)$ abbreviate $z^T \equiv \alpha\,\mathrm{fl}(x^T H) + \mathrm{fl}(\alpha x^T d) w^T + \alpha_1 v^T$ so that $y^T = \mathrm{fl}(z^T) = z^T + h_3^T$, where $|h_3^T| \le 3.03\epsilon\, z^T$. From $z^T = x^T G + \alpha h_1^T + h_2 w^T$ follows

$$\begin{aligned} |h_3^T| &\le 3.03\epsilon\, \left(x^T G + 1.01m\epsilon\,(x^T H) + 1.01\delta\epsilon\,(\alpha x^T d w^T)\right) \\ &\le 3.03\epsilon\, \left(x^T G + 1.01M\epsilon\,(\alpha x^T S)\right). \end{aligned}$$

In order to express y^T in terms of $x^T G$, write $y^T = z^T + h_3^T = x^T G + \alpha h_1^T + h_2 w^T + h_3^T$. Then $y^T = x^T G + h_4^T$, where $h_4^T \equiv \alpha h_1^T + h_2 w^T + h_3^T$. Hence

$$\begin{aligned} |h_4^T| &\le 1.01m\epsilon\,(\alpha x^T H) + 1.01\delta\epsilon\,(\alpha x^T d w^T) + 3.03\epsilon\, \left(x^T G + 1.01M\epsilon\,(\alpha x^T S)\right) \\ &\le 1.01M\epsilon\,(\alpha x^T S) + 3.03\epsilon\, \left(x^T G + 1.01M\epsilon\,(\alpha x^T S)\right) \\ &\le \epsilon\left(3.03\, x^T G + 1.01M(1 + 3.03\epsilon)\,(\alpha x^T S)\right). \end{aligned}$$

4. Now comes the computation of $\|y\|_1$. Since $n\epsilon < .01$, and the additions in $\|y\|_1$ involve only nonnegative numbers, a compensated summation gives [22, (4.8)] $\eta \equiv \mathrm{fl}(\|y\|_1) = \|y\|_1(1 + h_5)$, where $|h_5| \le 2\epsilon + \mathcal{O}(n\epsilon^2)$. It is more convenient to write instead $\eta = \|y\|_1/(1 + h_6)$, where $|h_6| \le 2\epsilon + \mathcal{O}(n\epsilon^2)$.

5. A final division completes the normalization,

$$\hat{x}^{(k+1)} = \mathrm{fl}\left(\frac{y}{\eta}\right) = \frac{y^T}{\eta} + h_7^T, \qquad \text{where} \quad |h_7^T| \le \epsilon\,\frac{y^T}{\eta}.$$

To express $\hat{x}^{(k+1)}$ in terms of $x^T G$ write

$$\hat{x}^{(k+1)} = \frac{y^T}{\|y\|_1} + \frac{y^T}{\|y\|_1} h_6 + h_7^T = \frac{x^T G + h_4}{\|y\|_1} + \frac{y^T}{\|y\|_1} h_6 + h_7^T = x^T G + g_{k+1}^T,$$

where

$$g_{k+1}^T = \frac{(1 - \|y\|_1) x^T G + h_4^T}{\|y\|_1} + \frac{y^T}{\|y\|_1} h_6 + h_7^T.$$

Apply $\|y\|_1 \ge 1 - \|h_4\|_1$ twice to get

$$\|g_{k+1}\|_1 \le \frac{2\|h_4\|_1}{1 - \|h_4\|_1} + |h_6| + \|h_7\|_1.$$

From $\|h_7\|_1 \le \epsilon(1 + |h_6|)$ and $|h_6| \le 2\epsilon + \mathcal{O}(n\epsilon^2)$ follows

$$\|g_{k+1}\|_1 \le \frac{2\|h_4\|_1}{1 - \|h_4\|_1} + 3\epsilon + \mathcal{O}(n\epsilon^2).$$

At last use $\|h_4\|_1 \le \epsilon(3.03 + c\alpha M)$. ∎

TABLE 5.1
Range of parameter values for experiments in section 6.

Unit roundoff	$\epsilon \approx 10^{-16}$
Damping factor	$\alpha = .85$
Dimension of H, S, G	$n \leq 4 \cdot 10^6$
Max # nonzeros in columns of H	$m \leq 2 \cdot 10^5$
Max # iterations	$k \leq 200$

Theorem 5.2 implies that the roundoff error in an iteration of the power method (FP) is bounded approximately by

$$\|g_k\|_1 \lessapprox \frac{2\alpha M\epsilon}{1-\alpha M\epsilon} \leq 4\alpha M\epsilon \qquad \text{if} \quad \alpha M\epsilon \leq 1/2.$$

Because we assume use of compensated summation, the roundoff error $\|g_k\|_1$ does not depend, to first order, on the matrix dimension n. It also does not depend on the iteration count k. The roundoff error is more or less constant, and the same for all iterations. It represents the error caused by a single matrix vector multiply, and is determined, for the most part, by the maximal number of nonzeros m in any column of H (i.e., the maximal number of inlinks into any web page) and the number of dangling nodes $\|d\|_1$, whichever is larger. The discussion relating to Figure 5.1 in section 5.2 indicates that the bounds in Theorem 5.2 are realistic, and not too pessimistic.

The only error we did not capture effectively in Theorem 5.2 consists of the higher order effects $\mathcal{O}(n\epsilon^2)$ in the compensated summation. Higher order effects can be completely avoided with *doubly compensated summation* [22, Algorithm 4.3] for applications of PageRank for matrices with dimensions not exceeding $n \leq 2^{13} = 8192$.

COROLLARY 5.3 (floating point version of error bounds). *With the assumptions in Theorem* 5.2*, the bounds in Theorem* 5.1 *hold with*

$$\|g_k\|_1 \leq \frac{2\epsilon(3.03 + c\alpha M)}{1-\epsilon(3.03+c\alpha M)} + \mathcal{O}(n\epsilon^2), \qquad k \geq 1,$$

where $M \equiv \max\{m, \|d\|_1 + 1\}$ *and* $c \equiv 1.01(1 + 3.03\epsilon)$.

Corollary 5.3 implies that the floating point error in the bounds is independent of the iteration count. The error is caused essentially by the matrix vector multiply and can be assumed to be constant. Moreover, the contribution of the floating point error to the different types of bounds is essentially the same, so that all bounds incur more or less the same floating point error.

We examine the ramifications of the above analysis when the power method (FP) is applied to the data matrices in section 6, whose parameter ranges are listed in Table 5.1. The simple bound in Theorem 4.3 and the roundoff error bound in Theorem 5.2 amount to $2\alpha^k \geq 10^{-14}$ and $\|g_k\|_1 \leq 4 \cdot 10^{-11}$. The roundoff error dominates the ranking bounds in later iterations, so that the bounds remain essentially constant from then on. Since the iterates can still change, though, one could continue the power method (FP) as long as the ranking criteria in Theorem 5.1 collect new ranking information. Note that the ranking criteria do not care whether the errors are due to finite termination or roundoff. For illustration purposes we execute 200 iterations of the power method in the experiments in section 6. A suitable termination criterion would stop the iterations once $\log(2\alpha^k) < \log(\|g_k\|_1)$.

TABLE 6.1
Properties of the data matrices (n = matrix dimension, m = maximal number of nonzeros in any column, $M = \max\{m, \text{dangling nodes} + 1\}$, and g = roundoff error).

Matrix	n	Nonzeros	m	Dangling nodes		M	g
H_S	9,914	36,854	340	2,861	29%	2,862	10^{-12}
H_L	3,148,440	39,383,235	168,685	91,462	3%	168,685	10^{-10}

The higher order effects $\mathcal{O}(n\epsilon^2)$ from the compensated summation are not likely to be of any consequence for the experiments in section 6 because $n\epsilon^2 \leq 10^{-25}$, which is negligible compared to $2\alpha^k \geq 10^{-14}$.

6. Numerical experiments. We present numerical experiments on data matrices from web crawls to compare the finite precision error bounds in section 5 and assess the performance of the ranking criterion. We describe the data matrices in section 6.1, and compare the bounds with respect to tightness in section 6.2 and with regard to ranking performance in section 6.3.

6.1. Data matrices. We present numerical experiments with two matrices that are obtained from web crawls and available on David Gleich's web page [16].

The properties of the two matrices are listed in Table 6.1. The small matrix H_S of dimension 9,914 represents a 2001 crawl [16, Webbase subgraph cs.stanford.edu], while the larger matrix H_L of dimension 3,148,440 represents a 2006 crawl [16, Wikipedia 2006-11-04]. Although the matrix H_S is small and dates from an older crawl, its larger percentage of dangling nodes is more representative of web graphs than that of H_L.

We choose the most popular values for the parameters of the Google matrix: $\alpha = .85$ for the amplification factor; and the uniform vector for personalization, dangling node, and starting vectors, $x^{(0)} = v = w = \frac{1}{n}\mathbf{1}$. The two data matrices for our experiments are

$$G_S \equiv \alpha(H_S + dw^T) + (1-\alpha)\mathbf{1}v^T, \qquad G_L \equiv \alpha(H_L + dw^T) + (1-\alpha)\mathbf{1}v^T.$$

The quantity g in Table 6.1 denotes the roundoff error from Corollary 5.3,

$$g \equiv \frac{2\epsilon(3.03 + c\alpha M)}{1 - \epsilon(3.03 + c\alpha M)}, \qquad M \equiv \max\{m, \|d\|_1 + 1\}, \qquad c \equiv 1.01(1 + 3.03\epsilon).$$

Note that the number of dangling nodes in G_S exceeds the maximal number of inlinks. Hence the dominant amplification factor M for the roundoff error of G_S is determined by the number of dangling nodes; see Theorem 5.2. This may reflect more what happens in practice when the nondangling nodes can be outnumbered by the dangling nodes, especially when applied to web graphs, since dangling nodes are part of the ever increasing web frontier. All experiments are performed in Matlab. We did not compute the norms with compensated summation because Matlab's accuracy appears to be sufficient for small problems with $n \leq 10^7$.

6.2. Tightness of the bounds. We compare bounds for the error $\|\hat{x}^{(k)} - \pi\|_1$ in iteration k. Since the roundoff error is essentially the same for all bounds and constant in each iteration, see Corollary 5.3, it suffices to compare the exact bounds. We assume that storage for 3 iterates is available, and that no overwriting takes place, so that successive iterates $\hat{x}^{(k-1)}$ and $\hat{x}^{(k)}$ require different storage locations. Among the resulting seven bounds below we included the k-step backward bound to illustrate the behavior of a bound where j is a function of k.

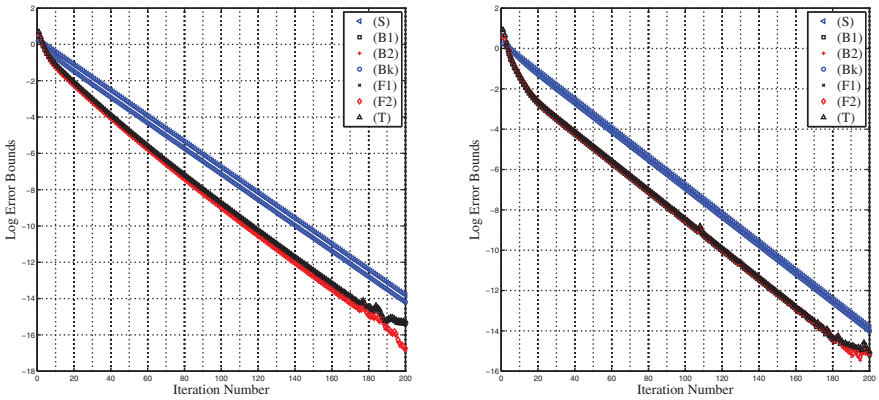

FIG. 6.1. *Error bounds for power method* (FP) *applied to the matrices* G_S *(left) and* G_L *(right).*

(S) Simple bound: $2\alpha^k$

(B1) Backward looking 1-step bound: $\frac{\alpha}{1-\alpha}\|\hat{x}^{(k-1)} - \hat{x}^{(k)}\|_1$

(B2) Backward looking 2-step bound: $\frac{\alpha^2}{1-\alpha^2}\|\hat{x}^{(k-2)} - \hat{x}^{(k)}\|_1$

(Bk) Backward looking k-step bound: $\frac{\alpha^k}{1-\alpha^k}\|x^{(0)} - \hat{x}^{(k)}\|_1$

(F1) Forward looking 1-step bound: $\frac{\|\hat{x}^{(k+1)} - \hat{x}^{(k)}\|_1}{1-\alpha}$

(F2) Forward looking 2-step bound: $\frac{\|\hat{x}^{(k+2)} - \hat{x}^{(k)}\|_1}{1-\alpha^2}$

(T) Two level 1-1-step bound: $\|\hat{x}^{(k+1)} - \hat{x}^{(k)}\|_1 + \frac{\|\hat{x}^{(k+2)} - \hat{x}^{(k+1)}\|_1}{1-\alpha}$

Figure 6.1 shows the above bounds for the matrices G_S and G_L. The bounds fall into two groups. The first group, consisting of (S) and (Bk), is less tight than the second group, which comprises the remaining bounds. There is little difference among the bounds in the second group. The straight lines in the context of the vertical logarithmic axis suggest that the geometric distances between iterates decrease at the same rate.

6.3. Ranking performance. Due to the lack of difference among the bounds in the competitive second group, we choose only (B1) for ranking, because it is the cheapest. The floating point version of the corresponding ranking criterion from Theorem 5.1 is as follows: If $\hat{x}_i^{(k)} > \hat{x}_j^{(k)} + \beta_B$, then $\pi_i > \pi_j$, where

$$\text{(B1-FP)} \qquad \beta_B \equiv \frac{\alpha}{1-\alpha}\|\hat{x}^{(k-1)} - \hat{x}^{(k)}\|_1 + g.$$

Applicability. Let Q_j be a permutation that orders the elements of $\hat{x}^{(j)}$ in decreasing order. That is, $\tilde{x}^{(j)} \equiv Q_j\hat{x}^{(j)}$ where $\tilde{x}_1^{(j)} \geq \ldots \geq \tilde{x}_n^{(j)}$. We count the number of pairs to which the criterion (B1-FP) applies. That is, we count the number of distinct pairs for which $\tilde{x}_i^{(j)} > \tilde{x}_{i+1}^{(j)} + \beta_B$ in iterations $1, \ldots, j$. Figure 6.2 shows this number for each iteration with the matrix G_S (due to memory limitations we were not able to collect this information for the large matrix G_L in every iteration). The line in the upper half represents the number of pairs of identical elements $\tilde{x}_i^{(j)} = \tilde{x}_{i+1}^{(j)}$ in each iterate. For instance, if $\tilde{x}_1^{(j)} = \tilde{x}_2^{(j)} = \tilde{x}_3^{(j)}$, then we count the two pairs $(1, 2)$ and $(2, 3)$. Since the ranking criterion cannot be applied to $\tilde{x}_1$ and $\tilde{x}_2$, the number of identical element pairs puts a natural limit on the performance of any ranking criterion that does not rely on additional criteria.

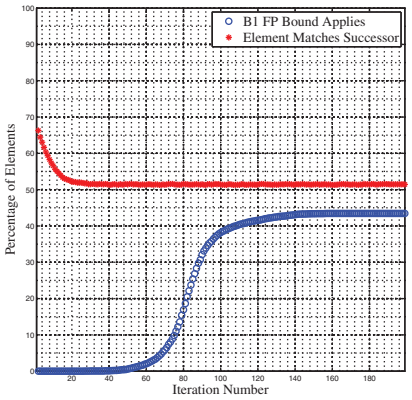

FIG. 6.2. *Applicability of ranking criterion* (B1-FP) *applied to the small matrix* G_S.

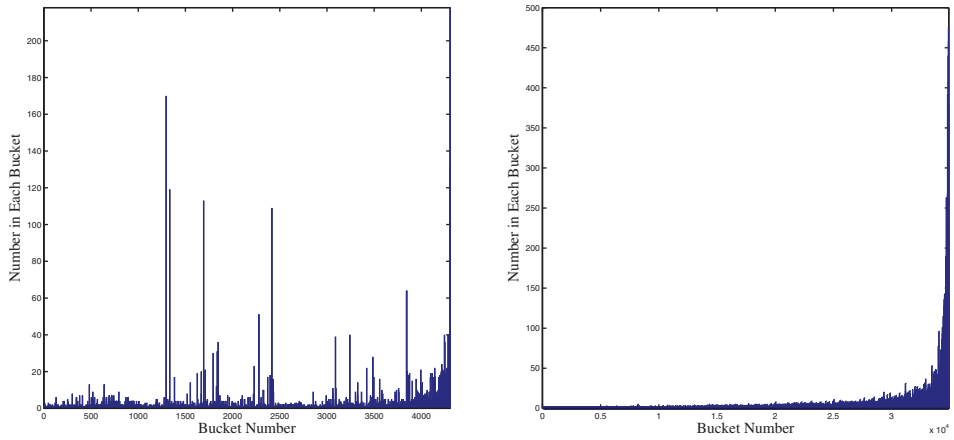

FIG. 6.3. *Buckets for ranking criterion* (B1-FP) *applied to iteration* 200 *with matrices* G_S *(left) and* G_L *(right).*

Figure 6.2 suggests that over 50% of the elements in most iterates are identical to another element. After about 100 iterations, criterion (B1-FP) applies to more than 40% of the elements. This means that only less than 10% of the elements remain unranked. The collection of new ranking information seems to level off after about 140 iterations. We can explain this as follows. The simple ranking bound $2\alpha^k$ is dominated by the roundoff error $\|g_k\|_1$ after about 170 iterations. Since the (B1) bound is tighter than the simple bound, the roundoff error dominates earlier. This is another justification for a termination criterion of the type already mentioned in section 5.3: Terminate the power method (FP) as soon as $\beta_B \approx g$.

Bucket ranking. Figure 6.3 gives an idea of how many different elements can be ranked and how big the buckets are. The histograms refer to the last iteration and depict the number of elements per bucket. To prevent distortion of the vertical axis and to assure better visibility for the top-ranked buckets, we omit the rightmost buckets (a single bucket in case of G_S and many buckets in case of G_L), which are the largest and contain the smallest elements.

Table 6.2 gives detailed information about the number and size of the buckets. The number of buckets represents *distinct* ranks that have been identified. For both

TABLE 6.2
Number and size of buckets for the matrices G_S (top) and G_L (bottom). The last two columns list the number of elements in the first and last buckets, respectively.

n	# Buckets	First bucket	Last bucket	
9,914	4,307	1	699	7%
3,148,440	34,911	1	2,996,646	95%

TABLE 6.3
Ranking information for the matrices G_S (top) and G_L (bottom). The second column lists the number of elements with exact rank, the third column the number of exactly ranked elements among the top 100 elements, and the last column lists the lowest rank that could be distinguished.

n	Exact ranking		Exact top 100	Lowest rank
9,914	3,177	32%	79	9,215
3,148,440	24,120	0.76%	100	151,794

matrices the ranking criterion (B1-FP) isolates the highest ranked PageRank score, because the first bucket contains only a single element. For the small matrix G_S, 7% of the smallest elements cannot be ranked, while for the large matrix G_L this number increases to 95%.

Table 6.3 gives detailed information about the ranking. It shows how many elements are ranked exactly, how many elements among the top 100 are exactly ranked, and the lowest rank that could be identified. Since the lowest identified rank for the matrix G_S is 9,215, the smallest bucket contains $n - 9,215 = 699$ elements, as shown in Table 6.2. The preceding information illustrates that the ranking criteria are able to identify the PageRanks of the top-ranked elements.

7. Extremely large matrices. As stated in Theorem 3.1, the main idea of our paper is the following ranking criterion: Let $x \geq 0$ with $\|x\|_1 = 1$ be any approximation to the PageRank vector π, and $\beta \geq \|x - \pi\|_1$.

(C) $$\text{If } x_i > x_j + \beta, \text{ then } \pi_i > \pi_j.$$

In the preceding sections we discussed the performance of (C) for matrices of dimension $n \leq 4 \cdot 10^6$, and for many applications of PageRank this is sufficient. However, the indexed web comprises hundreds of billions of web pages. Below are several suggestions for how to apply the criterion to matrices of extreme dimension.

7.1. Curbing roundoff error. The subsequent discussions are based on the roundoff error analysis in Theorem 5.2, which is valid for matrix dimensions $n < 10^{14}$ in IEEE double precision. Our experiments suggest that these roundoff error bounds are accurate and not at all pessimistic.

1. Computation of iterate norms. To prevent first-order dependence of the roundoff error on n, the iterates must be normalized on a regular basis, and the norms computed with compensated summation. However, even with compensated summation the higher order terms of the roundoff error can reach $n\epsilon^2 = 10^{-18}$ for $n = 10^{14}$ in IEEE double precision, which is large enough to be of concern for the ranking bound (C). Doubly compensated summation, or cascaded compensated summation [35, Algorithm 4.8], may be able to reduce higher order effects.

2. Matrix vector multiplications. Theorem 5.2 shows that with accurate computation of the iterate norms, the roundoff error is mainly due to a single matrix vector multiplication. In particular, it is determined by the maximal number of nonzeros in any column of the web matrix H (maximal number of inlinks) and the number of

dangling nodes (web pages without outlinks), whichever is larger. Since the dangling nodes are part of the increasing web frontier, they can easily outnumber the inlinks [12, section 2] and contribute substantially to the roundoff error. In section 7.2 we indicate how to reduce the influence of the dangling nodes.

3. Termination criteria. In later iterations the ranking bound β in (C) is dominated by roundoff error. Once this happens no new ranking information seems to be available. One may want to terminate the power method as soon as β is on the order of the roundoff error. The bound β is computed from geometric differences between iterates, that is, expressions of the form $\|\hat{x}^{(k+j)} - \hat{x}^{(k)}\|_1$. Catastrophic cancellation may damage the accuracy of these norms. This can be circumvented by resorting to the simple bound $2\alpha^k$ in Theorem 4.3. However, this bound is the least tight among all the bounds and requires the most iterations. A practical approach might be to just iterate until β is on the order of the roundoff error, so that accuracy in the computation of $\|\hat{x}^{(k+j)} - \hat{x}^{(k)}\|_1$ becomes less important, and collect ranking information only in the very last iteration.

7.2. Reducing matrix dimension. There are at least two advantages in reducing the matrix dimension: Faster computation and smaller roundoff error. Two easy approaches involve eliminating unreferenced pages (pages without inlinks) and dangling nodes (pages without outlinks). After such a reduction, the computation time depends only on the number of nondangling referenced pages, and the roundoff error depends only on the maximal number of inlinks to nondangling nodes.

1. Unreferenced pages. Suppose, as is likely in applications of PageRank to web graphs, that the dangling nodes outnumber the unreferenced pages, and that we have reordered the web matrix H so that the unreferenced pages are numbered last,

$$QHQ^T = \begin{pmatrix} H_1 & H_2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

where the diagonal blocks are square and Q is a permutation matrix. If one sets the trailing block of the dangling node vector equal to zero $w^TQ^T = \begin{pmatrix} w_1^T & w_2^T & 0 \end{pmatrix}$ so that the dangling nodes do not add new inlinks to the unreferenced pages, then the last block column of S is also zero. If we partition the personalization vector conformally, $v^TQ^T = \begin{pmatrix} v_1^T & v_2^T & v_3^T \end{pmatrix}$. Then $\pi^T = \pi^T G$ implies that the PageRank of the unreferenced pages is simply $(1-\alpha)v_3$. Furthermore, setting $v_3 = 0$ forces the PageRank of the unreferenced pages to be zero, so that they automatically receive the lowest ranking. Therefore, by keeping the dangling node vector positions associated with unreferenced pages zero, one can compute the PageRank of the remaining pages from the smaller matrix $\left(\begin{smallmatrix} H_1 & H_2 \\ 0 & 0 \end{smallmatrix}\right)$.

2. Dangling nodes. One can further reduce the matrix dimension by lumping all dangling nodes into a single node. The resulting lumped matrix $S_l \equiv \left(\begin{smallmatrix} H_1 & H_2\mathbf{1} \\ w_1^T & w_2^T\mathbf{1} \end{smallmatrix}\right)$ is stochastic and its dimension is equal to one plus the number of nondangling nodes [23]. One can rank the nondangling nodes by applying the power method (FP) and the ranking criterion (C) to $G_l \equiv \alpha S_l + (1-\alpha)\mathbf{1}v_l^T$, where $v_l^T \equiv \begin{pmatrix} v_1^T & v_2^T\mathbf{1} \end{pmatrix}$. The PageRanks of the dangling nodes can be recovered with a single matrix vector multiplication [23].

7.3. Ranking with faster converging methods. The ranking criterion (C) is not tied to any computational method. To apply it to a method other than the power method, one first needs rigorous bounds on the forward error $\|x - \pi\|_1$ that also

take into account roundoff error. This may be hard to do for methods with involved decision processes and intricate and possibly worse conditioned matrix operations.

Instead it may be easier to compute an approximation to PageRank with a fast converging method, such as a Krylov space method [17, 18], and then use this approximation as a restart for a single power method iteration to rank the nondangling nodes. Here is a more detailed description.

1. Apply a fast method to compute an approximation z to the PageRank of the lumped matrix G_l, and terminate when the residual norm is less than g_l. Here

$$g_l \equiv \frac{2\epsilon(3.03 + c\alpha m_l)}{1 - \epsilon(3.03 + c\alpha m_l)}, \qquad c \equiv 1.01(1 + 3.03\epsilon)$$

is the roundoff error in a single iteration of the power method, and m_l is the maximal number of nonzeros in any column of S_l.

2. Execute a single iteration of the power method (FP) with $\hat{x}^{(0)} := z/\|z\|_1$ as the starting vector. That is, $[y^{(1)}]^T := [\hat{x}^{(0)}]^T G_l$, $\hat{x}^{(1)} := y^{(1)}/\|y^{(1)}\|_1$, and compute $\|z\|_1$ and $\|y^{(1)}\|_1$ by a cacaded compensated summation method [35] or by doubly compensated summation [22, Algorithm 4.3].

3. Determine the ranking bound (B1-FP) from section 6.3, $\beta_l := \|\hat{x}^{(1)} - \hat{x}^{(0)}\|_1 + g_l$, where $\|\hat{x}^{(1)} - \hat{x}^{(0)}\|_1$ is computed by doubly compensated summation. Use the ranking criterion: If $\hat{x}_i^{(1)} > \hat{x}_j^{(1)} + \beta_l$, then $\pi_i > \pi_j$, and construct buckets according to the rules in section 3.

Acknowledgments. We thank Steve Kirkland for many helpful discussions, and David Gleich for sharing his data so generously and patiently. We are also grateful to Nick Higham for suggesting the use of compensated summation.

REFERENCES

[1] *http://www.google.com/technology/*. Our Search: Google Technology.

[2] R. Andersen, F. R. K. Chung, and K. J. Lang, *Local partitioning for directed graphs using PageRank*, in Algorithms and Models for the Web-Graph, Lecture Notes Comput. Sci. 4863, Springer Verlag, New York, 2007, pp. 166–178.

[3] R. Baeza-Yates, P. Boldi, and C. Castillo, *Generalizing PageRank: Damping functions for link-based ranking algorithms*, in Proc. ACM SIGIR, Seattle, WA, August 2006, ACM Press, pp. 308–315.

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

[5] M. Bianchini, M. Gori, and F. Scarselli, *Inside PageRank*, ACM Trans. Internat. Tech., 5 (2005), pp. 92–128.

[6] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas, *Link analysis ranking: Algorithms, theory, and experiments*, ACM Trans. Internat. Tech., 5 (2005), pp. 231–297.

[7] C. Brezinski and M. Redivo-Zaglia, *The PageRank vector: Properties, computation, approximation, and acceleration*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 551–575.

[8] S. Brin and L. Page, *The anatomy of a large-scale hypertextual Web search engine*, Comput. Networks ISDN Syst., 30 (1998), pp. 107–117.

[9] P. Chen, H. Xie, S. Maslov, and S. Redner, *Finding scientific gems with Google's PageRank algorithm*, J. Infornmet., 1 (2007), pp. 8–15.

[10] J. V. Davis and I. S. Dhillon, *Estimating the global PageRank of Web communities*, in KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM Press, New York, 2006, pp. 116–125.

[11] R. Dellavalle, L. Schilling, M. Rodriguez, H. V. de Sompel, and J. Bollen, *Refining dermatology journal impact factors using PageRank*, J. Am. Acad. Dermatol., 57 (2007), pp. 116–119.

[12] N. Eiron, K. S. McCurley, and J. A. Tomlin, *Ranking the Web frontier*, in WWW '04: Proceedings of the 13th Conference on World Wide Web, ACM Press, New York, 2004, pp. 309–318.

[13] L. Eldén, *A note on the eigenvalues of the Google matrix*, Technical report, LiTH-MAT-R-04-01, Department of Mathematics, Linköping University, 2004.
[14] A. Esuli and F. Sebastiani, *PageRanking WordNet synsets: An application to opinion mining*, in Proceedings of ACL 2007, Prague, CZ, 2007, pp. 25–27.
[15] R. Fagin, R. Kumar, and D. Sivakumar, *Comparing top* k *lists*, SIAM J. Discrete Math., 17 (2003), pp. 425–431.
[16] D. Gleich, *http://www.stanford.edu/~dgleich/data/*. Datasets, graphs and more.
[17] D. Gleich, L. Zhukov, and P. Berkhin, *Fast parallel PageRank: A linear system approach*, Technical report, Yahoo! Inc., 2004.
[18] G. H. Golub and C. Greif, *An Arnoldi-type algorithm for computing PageRank*, BIT, 46 (2006), pp. 759–771.
[19] T. L. Griffiths, M. Steyvers, and A. Firl, *Google and the mind: Predicting fluency with PageRank*, Psychological Science, 18 (2007), pp. 1069–1076.
[20] T. H. Haveliwala and S. D. Kamvar, *The second eigenvalue of the Google matrix*, Technical report 2003-20, Stanford University, 2003.
[21] M. Haviv and L. V. D. Heyden, *Perturbation bounds for the stationary probabilities of a finite Markov chain*, Adv. Appl. Prob., 16 (1984), pp. 804–818.
[22] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[23] I. C. F. Ipsen and T. M. Selee, *PageRank computation, with special attention to dangling nodes*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 1281–1296.
[24] 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.
[25] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, *Exploiting the block structure of the Web for computing PageRank*, Technical report 2003-17, Stanford University, 2003.
[26] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub, *Extrapolation methods for accelerating PageRank computations*, in The Twelfth International World Wide Web Conference, ACM Press, New York, 2003, pp. 261–270.
[27] S. M. G. Kendall, *Rank Correlation Methods*, Charles Griffin & Company Limited, 1975.
[28] A. Khalil and Y. Liu, *Experiments with PageRank computation*, Technical report 603, Computer Science Department at Indiana University, 2004.
[29] S. J. Kirkland, *Conditioning of the entries in the stationary vector of a Google-type matrix*, Linear Algebra Appl., 418 (2006), pp. 665–681.
[30] S. M. Kirsch, M. Gnasa, and A. B. Cremers, *Beyond the Web: Retrieval in social information spaces*, in ECIR, 2006, pp. 84–95.
[31] A. N. Langville and C. D. Meyer, *A survey of eigenvector methods for Web information retrieval*, SIAM Rev., 47 (2005), pp. 135–161.
[32] A. N. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings*, Princeton University Press, Princeton, NJ, 2006.
[33] R. Mihalcea, P. Tarau, and E. Figa, *PageRank on semantic networks, with application to word sense disambiguation*, in COLING '04: Proceedings of the 20th International Conference on Computational Linguistics, Morristown, NJ, 2004, Association for Computational Linguistics.
[34] J. L. Morrison, R. Breitling, D. J. Higham, and D. R. Gilbert, *GeneRank: Using search engine technology for the analysis of microarray experiments*, BMC Bioinformatics, 6 (2005), p. 233.
[35] T. Ogita, S. M. Rump, and S. Oishi, *Accurate sum and dot product*, SIAM J. Sci. Comput., 26 (2005), pp. 1955–1988.
[36] L. Page, S. Brin, R. Motwani, and T. Winograd, *The PageRank citation ranking: Bringing order to the Web*, Technical report, Stanford University, 1998.
[37] S. P. Ponzetto and M. Strube, *Deriving a large scale taxonomy from Wikipedia*, in AAAI '07, 2007, pp. 1440–1445.
[38] F. Qiu and J. Cho, *Automatic identification of user interest for personalized search*, in WWW '06: Proc. 15th International Conference on World Wide Web, ACM Press, New York, 2006, pp. 727–736.
[39] T. Sarlós, A. A. Benczúr, K. Csalogány, D. Fogaras, and B. Rácz, *To randomize or not to randomize: Space optimal summaries for hyperlink analysis*, in WWW '06: Proc. 15th International Conference on World Wide Web, ACM Press, New York, 2006, pp. 297–306.
[40] 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.
[41] T. Simonite, *Google tool could search out hospital superbugs*, 30 January 2008.

[42] G. W. Stewart, *On the powers of a matrix with perturbations*, Numer. Math., 96 (2003), pp. 363–376.
[43] R. S. Varga, *Matrix Iterative Analysis*, Prentice Hall, Englewood Cliffs, NJ, 1962.
[44] P. von Hilgers and A. N. Langville, *The five greatest applications of Markov Chains*, in Proceedings of the Markov Anniversary Meeting, Boston Press, Boston, MA, 2006.
[45] J. Wang, J. Liu, and C. Wang, *Keyword extraction based on PageRank*, in PAKDD, 2007, pp. 857–864.
[46] Y. Wang and D. J. DeWitt, *Computing PageRank in a distributed Internet search system*, in Proc. 30th VLDB Conference, 2004.
[47] R. S. Wills, *When rank trumps precision: Using the power method to compute Google's PageRank*, http://www.lib.ncsu.edu/theses/available/etd-06122007-173712/, 2007.