SIAM J. MATRIX ANAL. APPL.
Vol. 27, No. 4, pp. 968–987

UPDATING MARKOV CHAINS WITH AN EYE ON GOOGLE'S PAGERANK[*]

AMY N. LANGVILLE[†] AND CARL D. MEYER[‡]

Abstract. An iterative algorithm based on aggregation/disaggregation principles is presented for updating the stationary distribution of a finite homogeneous irreducible Markov chain. The focus is on large-scale problems of the kind that are characterized by Google's PageRank application, but the algorithm is shown to work well in general contexts. The algorithm is flexible in that it allows for changes to the transition probabilities as well as for the creation or deletion of states. In addition to establishing the rate of convergence, it is proven that the algorithm is globally convergent. Results of numerical experiments are presented.

Key words. Markov chains, updating, stationary vector, PageRank, stochastic complementation, aggregation/disaggregation, Google

AMS subject classifications. 60J10, 65C40, 15A51, 65F10, 65F15, 65F30, 65F50, 68P20, 68P10, 15A99, 15-04, 15A18, 15A06

DOI. 10.1137/040619028

1. Introduction.

Suppose that the stationary distribution vector

$$\boldsymbol{\phi}^T = (\phi_1, \phi_2, \ldots, \phi_m)$$

for an m-state homogeneous irreducible Markov chain with transition probability matrix $\mathbf{Q}_{m\times m}$ is known (by prior computation, such as solving $\boldsymbol{\phi}^T\mathbf{Q} = \boldsymbol{\phi}^T$, or by other means), but the chain requires updating by altering some of its transition probabilities or by adding or deleting some states. Suppose that the updated transition probability matrix $\mathbf{P}_{n\times n}$ is also irreducible. The updating problem is to compute the updated stationary distribution $\boldsymbol{\pi}^T = (\pi_1, \pi_2, \ldots, \pi_n)$ for $\mathbf{P}$ by somehow using the components in $\boldsymbol{\phi}^T$ to produce $\boldsymbol{\pi}^T$ with less effort than is required by working blind (i.e., by computing $\boldsymbol{\pi}^T$ without knowledge of $\boldsymbol{\phi}^T$).

When the updating involves only perturbing entries in $\mathbf{Q}$ (i.e., no states are added or deleted), the problem is referred to as an *element-updating problem.* If states need to be added or deleted, the problem is called a *state-updating problem.* The state-updating problem is clearly more difficult, and it generally includes the element-updating problem as a special case. Our purpose is to present a general-purpose algorithm that simultaneously handles both kinds of updating problems. But before presenting our algorithm, we review the shortcomings of some existing approaches to updating.

2. Restarting the power method.

Even for simple element updating, the restarted power method is not an overly effective technique. Suppose that the updating process calls for perturbing transition probabilities in $\mathbf{Q}$ to produce the updated matrix $\mathbf{P}$ (but no states are added or deleted), and suppose that it is known that

[*] Received by the editors November 16, 2004; accepted for publication (in revised form) by M. Benzi August 1, 2005; published electronically February 16, 2006. This research was supported in part by NSF grants CCR-ITR-0113121 and CCR-0318575.

http://www.siam.org/journals/simax/27-4/61902.html

[†] Department of Mathematics, College of Charleston, Charleston, SC 29424 (langvillea@cofc.edu).

[‡] Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205 (meyer@ncsu.edu).

the updated stationary distribution $\boldsymbol{\pi}^T$ for $\mathbf{P}$ is in some sense close to the original stationary distribution $\boldsymbol{\phi}^T$ for $\mathbf{Q}$. For example, this might occur if the perturbations to $\mathbf{Q}$ are small. It is intuitive that if $\boldsymbol{\phi}^T$ and $\boldsymbol{\pi}^T$ are close, then applying

$$\mathbf{x}_{j+1}^T = \mathbf{x}_j^T \mathbf{P} \quad \text{with} \quad \mathbf{x}_0^T = \boldsymbol{\phi}^T \tag{1}$$

should produce an accurate approximation to $\boldsymbol{\pi}^T$ in fewer iterations than are required when an arbitrary initial vector is used. To some extent this is true, but intuition generally overestimates the impact.

It is well known that if λ_2 is the subdominant eigenvalue of $\mathbf{P}$, and if λ_2 has index one (linear elementary divisors), then the asymptotic rate of convergence [29, p. 621] of (1) is

$$R = -\log_{10} |\lambda_2|. \tag{2}$$

For linear stationary iterative procedures the asymptotic rate of convergence R is an indication of the number of digits of accuracy that can be expected to be eventually gained on each iteration, and this is independent of the initial vector. For example, suppose that the entries of $\mathbf{P} - \mathbf{Q}$ are small enough to ensure that each component π_i agrees with ϕ_i in the first significant digit, and suppose that the goal is to compute the update $\boldsymbol{\pi}^T$ to twelve significant places by using (1). Since $\mathbf{x}_0^T = \boldsymbol{\phi}^T$ already has one correct significant digit, and since about $1/R$ iterations are required to gain each additional significant digit of accuracy, (1) requires about $11/R$ iterations, whereas starting from scratch with an initial vector containing no significant digits of accuracy requires about $12/R$ iterations. In other words, the effort is reduced by about 8% for each correct significant digit that can be built into $\mathbf{x}_0^T$. This dictates how much effort should be invested in determining a "good" initial vector. Of course, if one is willing to settle for fewer digits of accuracy or the number of agreeing digits is higher, then the savings could be more substantial.

To appreciate what this means concerning the effectiveness of using (1) as an updating technique, suppose, for example, that $|\lambda_2| = .85$, and suppose that the perturbations involved in updating $\mathbf{Q}$ to $\mathbf{P}$ are such that each component π_i agrees with ϕ_i in the first significant digit. If (1) is used to produce twelve significant digits of accuracy, then it follows from (2) that about 156 iterations are required. This is only about 16 fewer than are needed when starting blind with a random initial vector. Consequently, the power method is not an attractive approach to the element-updating problem even when changes are relatively small. While grossly ineffective, state updating can be accomplished by restarting the power method with an updated $\mathbf{Q}$ and an initial vector obtained by renormalizing results from prior computations after appropriate components are added or deleted. In other words, the restarted power method is not a viable technique for either element updating or state updating.

2.1. Faster converging states. There are times (e.g., see section 8) when there is a need to isolate the faster evolving states of a chain from the slower ones. This is especially true for chains such as Google's PageRank application (discussed later) in which there are a relatively small number of slower evolving components that drag the entire limiting process down. The asymptotic rate of convergence (2) of the power method is of little help here because it is by design a conservative measure that accounts for the overall evolution rate of the process, and hence must account for the slowest converging states.

To get a sense of what determines which components converge faster than others, let $\{1, \lambda_2, \ldots, \lambda_k\}$ be the *distinct* eigenvalues of $\mathbf{P}$, and suppose that $\mathbf{P}$ has a standard spectral decomposition [15], [29, p. 517]

$$\mathbf{P} = \sum_{i=1}^{k} \lambda_i \mathbf{G}_i = \mathbf{e}\boldsymbol{\pi}^T + \sum_{i=2}^{k} \lambda_i \mathbf{G}_i \quad \Longrightarrow \quad \mathbf{P}^n = \sum_{i=1}^{k} \lambda_i^n \mathbf{G}_i = \mathbf{e}\boldsymbol{\pi}^T + \sum_{i=2}^{k} \lambda_i^n \mathbf{G}_i,$$

where $\mathbf{e} = (1, 1, \ldots, 1)^T$, $\mathbf{G}_i$ is the ith spectral projector, and $1 > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_k|$. If λ_2 is a simple eigenvalue, then $\mathbf{G}_2 = \mathbf{u}_2\mathbf{v}_2^T/\mathbf{v}_2^T\mathbf{u}_2$, where $\mathbf{v}_2^T$ and $\mathbf{u}_2$ are left and right eigenvectors associated with λ_2, respectively, and

$$\mathbf{x}_n^T = \mathbf{x}_0^T \mathbf{P}^n x_0^T \mathbf{G}_i = \boldsymbol{\pi}^T + \xi \lambda_2^n \mathbf{v}_2^T + \sum_{i=3}^{k} \lambda_i^n \mathbf{x}_0^T \mathbf{G}_i, \tag{3}$$

where $\xi = \mathbf{x}_0^T\mathbf{u}_2/\mathbf{v}_2^T\mathbf{u}_2$. If we exclude the cases in which $|\lambda_2| = |\lambda_3|$, $\xi = 0$ (i.e., $\mathbf{x}_0 \perp \mathbf{u}_2$), and that components in $\xi\lambda_2^n\mathbf{v}_2^T$ are canceled out by corresponding components in the remainder of the sum (all of which are unlikely in practice), then it is clear from (3) that a given component in $\mathbf{x}_n^T$ will converge at a rate faster than that dictated by (2) if and only if the corresponding component in the left-hand eigenvector $\mathbf{v}_2^T$ is zero. In other words, *the positions of the zero (or near-zero) entries in* $\mathbf{v}_2^T$ *dictate the faster evolving components.* There is a special class of matrices for which $\mathbf{v}_2^T$ is known to have many components of very small magnitude. We postpone discussion of these until section 8.2.

3. Exact updating.

The purpose of this section is to show that element updating can, in fact, always be accomplished by means of a simple and direct formula that is guaranteed to return *exact* results (in exact arithmetic), even when perturbations are allowed to be large. However, you may not want to pay the computational cost to obtain exact results when significant updating is needed.

Consider perturbing some transition probabilities in $\mathbf{Q}$ (irreducible and stochastic) to produce an updated matrix $\mathbf{P}$ (also irreducible and stochastic) without adding or deleting states, but no longer constrain the perturbations to be small. Instead of considering perturbations that are small in magnitude, consider perturbations that affect only a small number of states.

The problem is cast in terms of updating $\mathbf{Q}$ one row at a time. The idea is similar to application of the Sherman–Morrison formula [29, p. 124] for updating a solution to a nonsingular linear system, but the techniques must be adapted to the singular matrix $\mathbf{A} = \mathbf{I} - \mathbf{Q}$. The mechanism for doing this is by means of the group inverse $\mathbf{A}^\#$ for $\mathbf{A}$, which is the unique matrix satisfying the three equations $\mathbf{A}\mathbf{A}^\#\mathbf{A} = \mathbf{A}$, $\mathbf{A}^\#\mathbf{A}\mathbf{A}^\# = \mathbf{A}^\#$, and $\mathbf{A}\mathbf{A}^\# = \mathbf{A}^\#\mathbf{A}$. This matrix is often involved in questions concerning Markov chains; see [6, 23, 29] for some general background and [6, 9, 11, 14, 23, 25, 27, 28, 31, 32, 38] for Markov chain applications. The precise formula to perform exact updating is as follows.

THEOREM 3.1. *Let* $\mathbf{Q}$ *be the transition probability matrix of an irreducible Markov chain and suppose that the ith row* $\mathbf{q}^T$ *of* $\mathbf{Q}$ *is updated to produce* $\mathbf{p}^T = \mathbf{q}^T - \boldsymbol{\delta}^T$, *the ith row of* $\mathbf{P}$, *which is also the transition matrix of an irreducible chain. If* $\boldsymbol{\phi}^T$ *and* $\boldsymbol{\pi}^T$ *denote the stationary probability distributions of* $\mathbf{Q}$ *and* $\mathbf{P}$, *respectively, and if* $\mathbf{A} = \mathbf{I} - \mathbf{Q}$, *then* $\boldsymbol{\pi}^T = \boldsymbol{\phi}^T - \boldsymbol{\epsilon}^T$, *where*

$$\boldsymbol{\epsilon}^T = \left[\frac{\phi_i}{1 + \boldsymbol{\delta}^T\mathbf{A}^\#_{*i}}\right] \boldsymbol{\delta}^T\mathbf{A}^\# \qquad (\mathbf{A}^\#_{*i} = \text{the } i\text{th column of } \mathbf{A}^\#). \tag{4}$$

To handle multiple row updates to $\mathbf{Q}$*, this formula must be sequentially applied one row at a time, which means that the group inverse must be sequentially updated. The formula for updating* $(\mathbf{I}-\mathbf{Q})^{\#}$ *to* $(\mathbf{I}-\mathbf{P})^{\#}$ *is as follows:*

$$(5) \qquad (\mathbf{I}-\mathbf{P})^{\#} = \mathbf{A}^{\#} + \mathbf{e}\boldsymbol{\epsilon}^T\left[\mathbf{A}^{\#} - \gamma\mathbf{I}\right] - \frac{\mathbf{A}^{\#}_{*i}\boldsymbol{\epsilon}^T}{\phi_i}, \quad \textit{where} \quad \gamma = \frac{\boldsymbol{\epsilon}^T\mathbf{A}^{\#}_{*i}}{\phi_i}.$$

Since exact updating is not the primary focus of this article, the formal proof of Theorem 3.1 is omitted, but the interested reader can find the details that constitute a proof in [31].

While Theorem 3.1 provides theoretical answers to the element updating problem, it is not computationally satisfying, especially if more than just one or two rows are involved. If every row needs to be touched, then using formulas (4) and (5) requires $O(n^3)$ floating point operations, which is comparable to the cost of starting from scratch.

Other updating formulas exist [9, 12, 16, 20, 36], but all are variations of the same rank-one updating idea involving a Sherman–Morrison [13, 15], [29, p. 124] type of formula, and all are $O(n^3)$ algorithms for a general update. Moreover, all of these rank-one updating techniques apply only to the simpler element-updating problem and are not easily adapted to handle more complicated state-updating problems. The bottom line is that while exact element-updating formulas might be useful when only a row or two need to be changed and no states are added or deleted, they are not practical for making more general updates.

4. Approximate updating using approximate aggregation.

If, instead of aiming for the exact value of the updated stationary distribution, one is willing to settle for an approximation, then the door opens wider. For example, an approximation approach based on state-lumping has been suggested in [7] to estimate Google's PageRank vector. State-lumping is part of a well-known class of methods known as *approximate aggregation techniques* [37] that have been used in the past to estimate stationary distributions of nearly uncoupled chains. Even though it produces only estimates of $\boldsymbol{\pi}^T$, approximate aggregation can handle both element updating as well as state updating and is computationally cheap.

The idea behind the application of approximate aggregation to perform updating is to use the previously known distribution

$$\boldsymbol{\phi}^T = (\phi_1, \phi_2, \ldots, \phi_m)$$

together with the updated transition probabilities in $\mathbf{P}$ to build an aggregated Markov chain having a transition probability matrix $\mathbf{A}$ that is smaller in size than $\mathbf{P}$. The stationary distribution $\boldsymbol{\alpha}^T$ of $\mathbf{A}$ is used to generate an estimate of the true updated distribution $\boldsymbol{\pi}^T$ as outlined below.

The state space $\mathcal{S}$ of the updated Markov chain is first partitioned as $\mathcal{S} = G \cup \overline{G}$, where G is the subset of states whose stationary probabilities are likely to be most affected by the updates (newly added states are automatically included in G, and deleted states are accounted for by changing affected transition probabilities to zero). The complement $\overline{G}$ naturally contains all other states. The intuition is that the effect of perturbations involving only a few states in large sparse chains (such as those in Google's PageRank application) is primarily local, and most stationary probabilities are not significantly affected. Deriving good methods for determining G is a pivotal issue, and this is discussed in detail in section 8.

Partitioning the states of the updated chain as $\mathcal{S} = G \cup \overline{G}$ induces a partition (and reordering) of the updated transition matrix

$$\mathbf{P}_{n\times n} = \begin{array}{c c} & \begin{array}{cc} G & \overline{G} \end{array} \\ \begin{array}{c} G \\ \overline{G} \end{array} & \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{pmatrix} \end{array}, \tag{6}$$

where $\mathbf{P}_{11}$ is $g \times g$ with $g = |G|$ being the cardinality of G and $\mathbf{P}_{22}$ is $(n-g)\times(n-g)$. Similarly, the associated stationary distribution

$$\boldsymbol{\pi}^T = (\pi_1, \ldots, \pi_g \,|\, \pi_{g+1}, \ldots, \pi_n) = \left(\pi_1, \ldots, \pi_g \,|\, \overline{\boldsymbol{\pi}}^T\right)$$

is reordered and partitioned in an associated manner.

The stationary probabilities from the original distribution $\boldsymbol{\phi}^T$ that correspond to the states in $\overline{G}$ are placed in a row vector $\overline{\boldsymbol{\phi}}^T$, and the states in $\overline{G}$ are lumped into one superstate to create a smaller aggregated Markov chain whose transition matrix is the $(g+1)\times(g+1)$ matrix given by

$$\widetilde{\mathbf{A}} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12}\mathbf{e} \\ \tilde{\mathbf{s}}^T\mathbf{P}_{21} & 1 - \tilde{\mathbf{s}}^T\mathbf{P}_{21}\mathbf{e} \end{pmatrix}, \text{ where } \tilde{\mathbf{s}}^T = \frac{\overline{\boldsymbol{\phi}}^T}{\overline{\boldsymbol{\phi}}^T\mathbf{e}} \text{ ($\mathbf{e}$ is a column of ones).} \tag{7}$$

The approximation procedure in [7] computes the stationary distribution

$$\widetilde{\boldsymbol{\alpha}}^T = (\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_g, \tilde{\alpha}_{g+1})$$

for $\widetilde{\mathbf{A}}$ and uses the first g components in $\widetilde{\boldsymbol{\alpha}}^T$ along with those in $\overline{\boldsymbol{\phi}}^T$ to create an approximation $\widetilde{\boldsymbol{\pi}}^T$ to the true updated distribution $\boldsymbol{\pi}^T$ by setting

$$\widetilde{\boldsymbol{\pi}}^T = \left(\tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_g \,|\, \overline{\boldsymbol{\phi}}^T\right). \tag{8}$$

In other words, use

$$\pi_i \approx \tilde{\pi}_i = \begin{cases} \tilde{\alpha}_i & \text{if state } i \text{ belongs to } G, \\ \phi_i & \text{if state } i \text{ belongs to } \overline{G}. \end{cases} \tag{9}$$

The reason that (9) can produce good estimates is because it can be demonstrated (see section 6) that when there is absolutely no change in the stationary probabilities that correspond to states in $\overline{G}$ (i.e., when $\overline{\boldsymbol{\phi}}^T = \overline{\boldsymbol{\pi}}^T$), then

$$\tilde{\alpha}_i = \begin{cases} \pi_i & \text{for } 1 \le i \le g, \\ \overline{\boldsymbol{\pi}}^T\mathbf{e} & \text{for } i = g+1. \end{cases}$$

Therefore, when there is only a small change in the stationary probabilities that correspond to states in $\overline{G}$ (i.e., when $\overline{\boldsymbol{\phi}}^T \approx \overline{\boldsymbol{\pi}}^T$), it is reasonable to expect that $\tilde{\alpha}_i \approx \pi_i$ for $1 \le i \le g$. This along with $\overline{\boldsymbol{\phi}}^T\mathbf{e} \approx \overline{\boldsymbol{\pi}}^T\mathbf{e}$ also ensures that $\widetilde{\boldsymbol{\pi}}^T\mathbf{e} \approx 1$, and thus the approximation (8) can be close enough to a probability vector to avoid the need for renormalization. The accuracy of this approximation scheme along with other theoretical details is discussed in sections 5 and 6.

5. Exact aggregation. The technique described in section 4 is simply one particular way to approximate the results of *exact* aggregation that is developed in [26] and is briefly outlined below. For an irreducible n-state Markov chain whose state space has been partitioned into k disjoint groups $\mathcal{S} = G_1 \cup G_2 \cup \cdots \cup G_k$, the associated transition probability matrix assumes the block-partitioned form

$$\mathbf{P}_{n\times n} = \begin{array}{c} \\ G_1 \\ G_2 \\ \vdots \\ G_k \end{array} \!\!\begin{array}{c} \begin{array}{cccc} G_1 & G_2 & \cdots & G_k \end{array} \\ \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \cdots & \mathbf{P}_{1k} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \cdots & \mathbf{P}_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{k1} & \mathbf{P}_{k2} & \cdots & \mathbf{P}_{kk} \end{pmatrix} \end{array} \quad \text{(with square diagonal blocks).} \tag{10}$$

This *parent* Markov chain defined by $\mathbf{P}$ induces k smaller Markov chains, called *censored chains*, as follows. The *censored Markov chain* associated with a group of states G_i is defined to be the Markov process that records the location of the parent chain only when the parent chain visits states in G_i. Visits to states outside of G_i are ignored. The transition probability matrix for the ith censored chain is the ith *stochastic complement* [26]

$$\mathbf{S}_i = \mathbf{P}_{ii} + \mathbf{P}_{i\star}(\mathbf{I} - \mathbf{P}_i^{\star})^{-1}\mathbf{P}_{\star i}, \tag{11}$$

in which $\mathbf{P}_{i\star}$ and $\mathbf{P}_{\star i}$ are, respectively, the ith row and the ith column of blocks with $\mathbf{P}_{ii}$ removed, and $\mathbf{P}_i^{\star}$ is the principal submatrix of $\mathbf{P}$ obtained by deleting the ith row and ith column of blocks. For example, if the partition consists of just two groups $\mathcal{S} = G_1 \cup G_2$, then there are only two censored chains, and their respective transition matrices are the two stochastic complements

$$\mathbf{S}_1 = \mathbf{P}_{11} + \mathbf{P}_{12}(\mathbf{I} - \mathbf{P}_{22})^{-1}\mathbf{P}_{21} \quad \text{and} \quad \mathbf{S}_2 = \mathbf{P}_{22} + \mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1}\mathbf{P}_{12}.$$

If the stationary distribution for $\mathbf{P}$ is $\boldsymbol{\pi}^T = (\boldsymbol{\pi}_1^T \,|\, \boldsymbol{\pi}_2^T \,|\, \cdots \,|\, \boldsymbol{\pi}_k^T)$ (partitioned conformably with $\mathbf{P}$), then the *ith censored distribution* (the stationary distribution for $\mathbf{S}_i$) is known [26] to be equal to

$$\mathbf{s}_i^T = \frac{\boldsymbol{\pi}_i^T}{\boldsymbol{\pi}_i^T\mathbf{e}} \qquad (\mathbf{e} \text{ is an appropriately sized column of ones}). \tag{12}$$

For primitive chains [29, p. 693] (also known as aperiodic or regular chains [20, 36, 35]), the jth component of $\mathbf{s}_i^T$ is the limiting conditional probability of being in the jth state of group G_i given that the process is somewhere in G_i.

To compress each group G_i into a single state in order to create a small k-state aggregated chain, squeeze the parent transition matrix $\mathbf{P}$ down to the *aggregated transition matrix* (sometimes called the *coupling matrix*) by setting

$$\mathbf{A}_{k\times k} = \begin{pmatrix} \mathbf{s}_1^T\mathbf{P}_{11}\mathbf{e} & \cdots & \mathbf{s}_1^T\mathbf{P}_{1k}\mathbf{e} \\ \vdots & \ddots & \vdots \\ \mathbf{s}_k^T\mathbf{P}_{k1}\mathbf{e} & \cdots & \mathbf{s}_k^T\mathbf{P}_{kk}\mathbf{e} \end{pmatrix}. \tag{13}$$

If $\mathbf{P}$ is stochastic and irreducible, then so is $\mathbf{A}$ [26]. For primitive chains, transitions between states in the aggregated chain defined by $\mathbf{A}$ correspond to transitions between groups G_i in the unaggregated parent chain when the parent chain is in equilibrium.

The remarkable feature surrounding this aggregation idea is that it allows a parent chain to be decomposed into k small censored chains that can be independently solved, and the resulting censored distributions $\mathbf{s}_i^T$ can be combined with the stationary distribution of $\mathbf{A}$ to construct the parent stationary distribution $\boldsymbol{\pi}^T$. This is the exact aggregation theorem.

Theorem 5.1 (exact aggregation [26]). *If* $\mathbf{P}$ *is the block-partitioned transition probability matrix* (10) *for an irreducible n-state Markov chain whose stationary probability distribution is*

$$\boldsymbol{\pi}^T = \boldsymbol{\pi}_1^T \mid \boldsymbol{\pi}_2^T \mid \cdots \mid \boldsymbol{\pi}_k^T \quad \textit{(partitioned conformably with } \mathbf{P}\textit{)},$$

and if $\alpha^T = (\alpha_1, \alpha_2, \ldots, \alpha_k)$ *is the stationary distribution for the aggregated chain defined by the matrix* $\mathbf{A}_{k\times k}$ *in* (13)*, then* $\alpha_i = \boldsymbol{\pi}_i^T\mathbf{e}$*, and the stationary distribution for* $\mathbf{P}$ *is*

$$\boldsymbol{\pi}^T = \left(\alpha_1\mathbf{s}_1^T \mid \alpha_2\mathbf{s}_2^T \mid \cdots \mid \alpha_k\mathbf{s}_k^T\right),$$

where $\mathbf{s}_i^T$ *is the censored distribution associated with the stochastic complement* $\mathbf{S}_i$ *in* (11)*.*

6. Exact versus approximate aggregation. While exact aggregation as presented in Theorem 5.1 is elegant, it is an inefficient numerical procedure for computing $\boldsymbol{\pi}^T$ because costly inversions are embedded in the stochastic complements (11) that are required to produce the censored distributions $\mathbf{s}_i^T$. Consequently, it is common to attempt to somehow approximate the censored distributions, and there are at least two methods for doing so.

1. Sometimes the stochastic complements $\mathbf{S}_i$ are first estimated, and then the distributions of these estimates are computed to provide approximate censored distributions, which in turn leads to an approximate aggregated transition matrix that is used to produce an approximation to $\boldsymbol{\pi}^T$ by employing Theorem 5.1.
2. The other approach is to bypass the stochastic complements altogether and somehow estimate the censored distributions $\mathbf{s}_i^T$ directly. This is the essence of the approximation scheme described in section 4.

To understand the application of the second approach given above, consider the updated transition matrix $\mathbf{P}$ given in (6) to be partitioned into $g+1$ levels in which the first g diagonal blocks are just 1×1, and the lower right-hand block is the $(n-g) \times (n-g)$ matrix $\mathbf{P}_{22}$ associated with the states in $\overline{G}$. In other words, to fit the context of the Theorem 5.1, the partition in (6) is viewed as

$$\mathbf{P} = \begin{array}{cc} & \begin{array}{cc} G & \overline{G} \end{array} \\ \begin{array}{c} G \\ \overline{G} \end{array} & \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{pmatrix} \end{array} = \left(\begin{array}{c|c|c|c} p_{11} & \cdots & p_{1g} & \mathbf{P}_{1\star} \\ \hline \vdots & \ddots & \vdots & \vdots \\ \hline p_{g1} & \cdots & p_{gg} & \mathbf{P}_{g\star} \\ \hline \mathbf{P}_{\star 1} & \cdots & \mathbf{P}_{\star g} & \mathbf{P}_{22} \end{array}\right), \tag{14}$$

where

$$\mathbf{P}_{11} = \begin{pmatrix} p_{11} & \cdots & p_{1g} \\ \vdots & \ddots & \vdots \\ p_{g1} & \cdots & p_{gg} \end{pmatrix}, \quad \mathbf{P}_{12} = \begin{pmatrix} \mathbf{P}_{1\star} \\ \vdots \\ \mathbf{P}_{g\star} \end{pmatrix}, \quad \text{and} \quad \mathbf{P}_{21} = (\mathbf{P}_{\star 1} \cdots \mathbf{P}_{\star g}).$$

Since the first g diagonal blocks in the partition (14) have size 1×1 (they are the scalars p_{ii}), it is evident that their corresponding stochastic complements are $\mathbf{S}_i = 1$ (because they are 1×1 stochastic matrices), and thus the censored distributions are $\mathbf{s}_i{}^T = 1$ for $1 \leq i \leq g$. This means that the *exact* aggregated transition matrix (13) associated with the partition (14) is

$$\mathbf{A} = \left(\begin{array}{c|c|c|c} p_{11} & \cdots & p_{1g} & \mathbf{P}_{1\star}\mathbf{e} \\ \hline \vdots & \ddots & \vdots & \vdots \\ \hline p_{g1} & \cdots & p_{gg} & \mathbf{P}_{g\star}\mathbf{e} \\ \hline \mathbf{s}^T\mathbf{P}_{\star 1} & \cdots & \mathbf{s}^T\mathbf{P}_{\star g} & \mathbf{s}^T\mathbf{P}_{22}\mathbf{e} \end{array}\right)_{(g+1)\times(g+1)} \tag{15}$$

$$= \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12}\mathbf{e} \\ \mathbf{s}^T\mathbf{P}_{21} & \mathbf{s}^T\mathbf{P}_{22}\mathbf{e} \end{pmatrix} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12}\mathbf{e} \\ \mathbf{s}^T\mathbf{P}_{21} & 1 - \mathbf{s}^T\mathbf{P}_{21}\mathbf{e} \end{pmatrix},$$

where $\mathbf{s}^T$ is the censored distribution derived from the only significant stochastic complement

$$\mathbf{S} = \mathbf{P}_{22} + \mathbf{P}_{21}\big(\mathbf{I} - \mathbf{P}_{11}\big)^{-1}\mathbf{P}_{12}.$$

If the stationary distribution for $\mathbf{A}$ is

$$\boldsymbol{\alpha}^T = \big(\alpha_1, \dots, \alpha_g, \alpha_{g+1}\big),$$

then exact aggregation (Theorem 5.1) ensures that the *exact* stationary distribution for $\mathbf{P}$ is

$$\boldsymbol{\pi}^T = (\pi_1, \dots, \pi_g \,|\, \pi_{g+1}, \dots, \pi_n) = \big(\pi_1, \dots, \pi_g \,|\, \overline{\boldsymbol{\pi}}^T\big) = \big(\alpha_1, \dots, \alpha_g \,|\, \alpha_{g+1}\mathbf{s}^T\big). \tag{16}$$

It is a fundamental issue to describe just how well the estimate $\widetilde{\boldsymbol{\pi}}^T$ given in (8) approximates the exact distribution $\boldsymbol{\pi}^T$ given in (16). Obviously, the degree to which $\widetilde{\pi}_i \approx \pi_i$ for $i > g$ (i.e., the degree to which $\overline{\boldsymbol{\phi}}^T \approx \overline{\boldsymbol{\pi}}^T$) depends on the degree to which the partition $\mathcal{S} = G \cup \overline{G}$ can be adequately constructed. While it is somewhat intuitive that this should also affect the degree to which $\widetilde{\pi}_i$ approximates π_i for $i \leq g$, it is not clear, at least on the surface, just how good this latter approximation is expected to be. The analysis is as follows.

Instead of using the exact censored distribution $\mathbf{s}^T$ to build the exact aggregated matrix $\mathbf{A}$ in (15), the vector $\tilde{\mathbf{s}}^T = \overline{\boldsymbol{\phi}}^T / \overline{\boldsymbol{\phi}}^T\mathbf{e}$ is used to approximate $\mathbf{s}^T$ in order to construct the approximate aggregate $\widetilde{\mathbf{A}}$ given in (7). The magnitude of

$$\boldsymbol{\delta}^T = \mathbf{s}^T - \tilde{\mathbf{s}}^T = \frac{\overline{\boldsymbol{\pi}}^T}{\overline{\boldsymbol{\pi}}^T\mathbf{e}} - \frac{\overline{\boldsymbol{\phi}}^T}{\overline{\boldsymbol{\phi}}^T\mathbf{e}}$$

and the magnitude of

$$\mathbf{E} = \mathbf{A} - \widetilde{\mathbf{A}} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{\delta}^T\mathbf{P}_{21} & -\boldsymbol{\delta}^T\mathbf{P}_{21}\mathbf{e} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\delta}^T \end{pmatrix} \mathbf{P}_{21}\big(\mathbf{I} \,|\, -\mathbf{e}\big) \tag{17}$$

are clearly of the same order. This suggests that if the partition $\mathcal{S} = G \cup \overline{G}$ can be adequately constructed so as to ensure that the magnitude of $\boldsymbol{\delta}^T$ is small, then $\widetilde{\mathbf{A}}$ is close to $\mathbf{A}$, so their respective stationary distributions $\widetilde{\boldsymbol{\alpha}}^T$ and $\boldsymbol{\alpha}^T$ should be close, thus ensuring that $\widetilde{\pi}_i$ and π_i are close for $i \leq g$.

However, Markov chains can exhibit sensitivities to small perturbations when a subdominant eigenvalue of the transition probability matrix is close to 1 [28, 30] or when there are large mean first passage times in the chain [8], and several measures of "condition" have been developed to gauge these situations [9, 11, 14, 18, 24].

The point being made here is that unless the degree to which $\mathbf{A}$ is well conditioned is established, the degree of the approximation in (8) is in doubt regardless of how close $\overline{\boldsymbol{\phi}}^T$ is to $\overline{\boldsymbol{\pi}}^T$. This may seem to be a criticism of the idea behind the approximation (8), but, to the contrary, the purpose of this article is to argue that this is a good idea because it can be viewed as the first step in an *iterative* aggregation scheme that performs remarkably well. The following sections are dedicated to developing an iterative aggregation approach to updating stationary probabilities.

7. Updating with iterative aggregation.

Iterative aggregation is an algorithm for solving nearly uncoupled (sometimes called nearly completely decomposable) Markov chains, and it is discussed in detail in [38]. Iterative aggregation is not a general-purpose technique, and it usually does not work for chains that are not nearly uncoupled. However, the ideas can be adapted to the updating problem, and these variations work extremely well, even when applied to Markov chains that are not nearly uncoupled. This is in part due to the fact that the approximate aggregation matrix (7) differs from the exact aggregation matrix (15) in only one row. Our iterative aggregation updating algorithm is described below.

Assume that the stationary distribution

$$\boldsymbol{\phi}^T = (\phi_1, \phi_2, \ldots, \phi_m)$$

for some irreducible Markov chain $\mathcal{C}$ is already known, perhaps from prior computations, and suppose that $\mathcal{C}$ needs to be updated. As in earlier sections, let the transition probability matrix and stationary distribution for the updated chain be denoted by $\mathbf{P}$ and

$$\boldsymbol{\pi}^T = (\pi_1, \pi_2, \ldots, \pi_n),$$

respectively. The updated matrix $\mathbf{P}$ is assumed to be irreducible. In applications such as computing Google's PageRank [3], irreducibility is guaranteed because small positive values are added to each entry resulting in $\mathbf{P} > \mathbf{0}$. It is important to note that m is not necessarily equal to n because the updating process allows for the addition or deletion of states as well as the alteration of transition probabilities.

The Iterative Aggregation Updating Algorithm

Initialization

i. Partition the states of the updated chain as $\mathcal{S} = G \cup \overline{G}$ and reorder $\mathbf{P}$ as described in (6)

ii. $\overline{\boldsymbol{\phi}}^T \longleftarrow$ the components from $\boldsymbol{\phi}^T$ that correspond to the states in $\overline{G}$

iii. $\mathbf{s}^T \longleftarrow \overline{\boldsymbol{\phi}}^T/(\overline{\boldsymbol{\phi}}^T\mathbf{e})$ (an initial approximate censored distribution)

Iterate until convergence

1. $\mathbf{A} \longleftarrow \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12}\mathbf{e} \\ \mathbf{s}^T\mathbf{P}_{21} & 1 - \mathbf{s}^T\mathbf{P}_{21}\mathbf{e} \end{pmatrix}_{(g+1)\times(g+1)} \qquad (\, g = |G|\,)$
2. $\boldsymbol{\alpha}^T \longleftarrow (\alpha_1, \alpha_2, \ldots, \alpha_g, \alpha_{g+1})$ (the stationary distribution of $\mathbf{A}$)
3. $\boldsymbol{\chi}^T \longleftarrow \big(\alpha_1, \alpha_2, \ldots, \alpha_g \,|\, \alpha_{g+1}\mathbf{s}^T\big)$
4. $\boldsymbol{\psi}^T \longleftarrow \boldsymbol{\chi}^T\mathbf{P}$ (see note following the algorithm)
5. If $\|\boldsymbol{\psi}^T - \boldsymbol{\chi}^T\| < \tau$ for a given tolerance τ, then quit—else $\mathbf{s}^T \longleftarrow \boldsymbol{\psi}^T/\boldsymbol{\psi}^T\mathbf{e}$ and go to step 1

Note concerning step 4. Step 4 is necessary because the vector $\boldsymbol{\chi}^T$ generated in step 3 is a fixed point in the sense that if step 4 is omitted and the process is restarted using $\boldsymbol{\chi}^T$ instead of $\boldsymbol{\psi}^T$, then the same $\boldsymbol{\chi}^T$ is simply reproduced at step 3 on each subsequent iteration. Step 4 has two purposes—it moves the iterate off the fixed point while simultaneously contributing to the convergence process. That is, the $\boldsymbol{\psi}^T$ resulting from step 4 can be used to restart the algorithm as well as produce a better approximation because applying a power step makes small progress toward the stationary solution. In the past, some authors [38] have used Gauss–Seidel in place of the power method at step 4.

While precise rates of convergence for general iterative aggregation algorithms are difficult to articulate, the specialized nature of our iterative aggregation updating algorithm allows us to easily establish its rate of convergence. The following theorem shows that this rate is directly dependent on how fast the powers of the one significant stochastic complement $\mathbf{S} = \mathbf{P}_{22} + \mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1}\mathbf{P}_{12}$ converge. In other words, since $\mathbf{S}$ is an irreducible stochastic matrix, the rate of convergence is completely dictated by the magnitude and Jordan structure of the largest subdominant eigenvalue of $\mathbf{S}$.

THEOREM 7.1. *The iterative aggregation updating algorithm defined above converges to the stationary distribution $\boldsymbol{\pi}^T$ of $\mathbf{P}$ for all partitions $\mathcal{S} = G \cup \overline{G}$. The rate at which the iterates converge to $\boldsymbol{\pi}^T$ is exactly the rate at which the powers $\mathbf{S}^n$ converge, which is governed by the magnitude and Jordan structure of largest subdominant eigenvalue $\lambda_2(\mathbf{S})$ of $\mathbf{S}$. If $\lambda_2(\mathbf{S})$ is real and simple, then the asymptotic rate of convergence is $R = -\log_{10}|\lambda_2(\mathbf{S})|$.*

Proof. For any initial probability vector $\mathbf{s}^T(0)$, let $\mathbf{A}(n)$, $\boldsymbol{\alpha}^T(n)$, $\boldsymbol{\chi}^T(n)$, $\boldsymbol{\psi}^T(n)$, and $\mathbf{s}^T(n)$ denote the respective results from steps 1–5 after n iterations of the iterative aggregation updating algorithm. A few straightforward calculations reveal that

$$\begin{aligned}
\mathbf{A}(n+1) &= \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12}\mathbf{e} \\ \mathbf{s}^T(n)\mathbf{P}_{21} & 1 - \mathbf{s}^T(n)\mathbf{P}_{21}\mathbf{e} \end{pmatrix}, \\
\boldsymbol{\alpha}^T(n+1) &= \beta_{n+1}\left(\mathbf{s}^T(n)\mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1} \,|\, 1\right), \\
&\qquad\qquad \text{where } \ \beta_{n+1} = \big(1 + \mathbf{s}^T(n)\mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1}\mathbf{e}\big)^{-1} \qquad [23, \text{p. } 458], \\
\boldsymbol{\chi}^T(n+1) &= \beta_{n+1}\left(\mathbf{s}^T(n)\mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1} \,|\, \mathbf{s}^T(n)\right), \\
\boldsymbol{\psi}^T(n+1) &= \beta_{n+1}\left(\mathbf{s}^T(n)\mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1} \,|\, \mathbf{s}^T(n)\mathbf{S}\right), \\
\mathbf{s}^T(n+1) &= \mathbf{s}^T(n)\mathbf{S} = \mathbf{s}^T(0)\mathbf{S}^n.
\end{aligned}$$

This makes it clear that $\mathbf{s}^T(n) \to \mathbf{s}^T$ (the censored distribution associated with $\mathbf{S}$) independent of the initial value $\mathbf{s}^T(0)$ and that the rate of convergence is exactly the rate at which $\mathbf{S}^n \to \mathbf{e}\mathbf{s}^T$. As $\mathbf{s}^T(n) \to \mathbf{s}^T$, we have

$$\beta_n \to \beta = \big(1 + \mathbf{s}^T\mathbf{P}_{21}(\mathbf{I} - \mathbf{P}_{11})^{-1}\mathbf{e}\big)^{-1}$$

and

$$\boldsymbol{\psi}^T(n) \to \beta\left(\mathbf{s}^T\mathbf{P}_{21}(\mathbf{I}-\mathbf{P}_{11})^{-1}\,|\,\mathbf{s}^T\right) = \boldsymbol{\pi}^T. \qquad \square$$

8. Determining the partition.

The iterative aggregation updating algorithm is globally convergent, and it never requires more iterations than the power method to attain a given level of convergence [17]. However, iterative aggregation clearly requires more work per iteration than the power method. One iteration of iterative aggregation requires forming the aggregation matrix, solving for its stationary vector, and executing one power iteration. The key to realizing an improvement in iterative aggregation over the power method rests in properly choosing the partition $\mathcal{S} = G \cup \overline{G}$. As Theorem 7.1 shows, good partitions are precisely those that yield a stochastic complement $\mathbf{S} = \mathbf{P}_{22} + \mathbf{P}_{21}(\mathbf{I}-\mathbf{P}_{11})^{-1}\mathbf{P}_{12}$ whose subdominant eigenvalue $\lambda_2(\mathbf{S})$ is small in magnitude.

Experience indicates that as $|G| = g$ (the size of $\mathbf{P}_{11}$) becomes larger, iterative aggregation tends to converge in fewer iterations. But as g becomes larger, each iteration requires more work, so the trick is to strike an acceptable balance. A small g that significantly reduces $|\lambda_2(\mathbf{S})|$ is the ideal situation.

Even for moderately sized problems there is an extremely large number of possible partitions, but there are some useful heuristics that can help guide the choice of G that will produce reasonably good results. For example, a relatively simple approach is to take G to be the set of all states "near" the updates, where "near" might be measured in a graph theoretic sense or else by transient flow (i.e., using the magnitude of entries of $\mathbf{x}_{j+1}^T = \mathbf{x}_j^T\mathbf{P}$ after j iterations, where j is small, say 5 or 10). In the absence of any other information, this naive strategy is at least a good place to start. However, there are usually additional options that lead to even better "G-sets," and some of these are described below.

8.1. Partitioning by differing time scales.

In most applications involving irreducible aperiodic Markov chains the components of the nth step distribution vector do not converge at a uniform rate, and consequently most iterative techniques, including the power method, often spend the majority of the time in resolving a small number of components—the slow ones. The slow converging components can be isolated either by monitoring the process for a few iterations or by theoretical means such as those described in section 2.1. For the PageRank problem, Kamvar et al. [19] have shown experimentally that a trend is set in the first few iterations, so that one can classify a state as slow converging or fast converging after just 10 iterations. If the states corresponding to the slower converging components are placed in G while the faster converging states are lumped into $\overline{G}$, then the iterative aggregation algorithm concentrates its effort on resolving the smaller number of slow converging states.

In loose terms, the effect of steps 1–3 in the iterative aggregation algorithm is essentially to make progress toward achieving an equilibrium (or steady state) for a smaller chain consisting of just the "slow states" in G together with one additional lumped state that accounts for all "fast states" in $\overline{G}$. The power iteration in step 4 moves the entire process ahead on a global basis, so if the slow states in G are substantially resolved by the relatively cheaper steps 1–3, then not many of the more costly global power steps are required to push the entire chain toward its global equilibrium. This is the essence of the original Simon–Ando idea first proposed in 1961 and explained and analyzed in [26, 37]. As $g = |G|$ becomes smaller relative to n, steps 1–3 become significantly cheaper to execute, and the process converges quite rapidly in both iteration count and wall-clock time. Examples and reports on experiments are given in section 8.3.

In some applications the slow states are particularly easy to identify because they are the ones having the larger stationary probabilities. This is a particularly nice state of affairs for the updating problem because we have the stationary probabilities from the prior period at our disposal, and thus all we have to do to construct a good G-set is to include the states with prior large stationary probabilities and throw in the states that were added or updated along with a few of their nearest neighbors. Clearly, this is an advantage only when there are just a few "large" states. However, it turns out that this is a characteristic feature of Google's PageRank application and other scale-free networks with power-law distributions. This is explained in the next section.

8.2. Scale-free networks and Google's PageRank.

As discussed in [1, 2, 5, 10], the link structure of the World Wide Web constitutes a *scale-free network.* This means that the number of nodes $n(l)$ having l edges (possibly directed) is proportional to l^{-k}, where k is a constant that does not change as the network expands (hence the term "scale-free"). In other words, the distribution of nodal degrees obeys a *power-law distribution* in the sense that

$$P[\deg(N) = d] \propto \frac{1}{d^k} \qquad \text{for some } k > 1 \qquad (\propto \text{ means "proportional to"}).$$

For example, studies [1, 2, 5, 10] have shown that for the World Wide Web the parameter for the indegree power-law distribution is $k \approx 2.1$, while the outdegree distribution has $k \approx 2.7$. The growth rate of the World Wide Web is tremendous, and the rapidly accumulating wealth of information contained therein is staggering, so the scale-free nature of the Web becomes important if the Web is to be harnessed.

The vast amount of knowledge would, for the most part, be inaccessible if it were not for Web search engines, and Google is the reigning champion in the search engine business. At the heart of Google is its innovative PageRank concept [3, 4], which is a process for assigning to each Web page a value that determines the order of presentation in reply to a query. PageRanks, in their purest form, are simply stationary probabilities for a particular Markov chain (described below). However, in practice, Google tweaks the mathematical PageRanks with proprietary "metrics" to create the final values that are used when matching a user's query. Notwithstanding the tweaking, the mathematics is the fundamental component of PageRank, and this is what we focus on.

In the remainder of this article, we consider PageRank to be the stationary probability distribution vector $\boldsymbol{\pi}^T$ for an irreducible aperiodic Markov chain whose transition probability matrix has the form

$$\mathbf{P}_{n\times n} = \alpha\mathbf{S} + (1-\alpha)\mathbf{e}\mathbf{v}^T,$$

in which $\mathbf{S} = \mathbf{H} + \mathbf{E}$ is a stochastic matrix involving the Web's raw hyperlink matrix $\mathbf{H}$ defined by

$$h_{ij} = \begin{cases} 1/(\text{total \# outlinks from page } \mathcal{P}_i) & \text{if } \mathcal{P}_i \text{ contains a link to } \mathcal{P}_j, \\ 0 & \text{otherwise} \end{cases}$$

and a modification matrix $\mathbf{E}$ that accounts for "dangling nodes" (pages with no outlinks). Vector $\mathbf{e}$ is a column of ones, $\mathbf{v}^T$ is a "personalization" probability vector that gives Google flexibility to perform customization, and $0 < \alpha < 1$ is Google's parameter that models a Web surfer's propensity to deviate from the underlying link structure; i.e., with probability $(1-\alpha)$ a Web surfer requests a "random" Web page. For more details concerning these and other features of PageRank, see [21, 22].

Of course, the scale is enormous—currently $n = O(10^9)$—and a recent MATLAB publication [33] characterized PageRank as "the world's largest matrix computation" with execution times measured in days. In building a search engine for a linked database in which the link structure is static (or nearly so), a large computational cost to determine PageRank might be tolerable because once $\boldsymbol{\pi}^T$ is determined the search engine can use it repeatedly. However, the World Wide Web is a dynamic network in which pages and links between them are being added and deleted almost continuously, so the problem of updating $\boldsymbol{\pi}^T$ is important and substantial. At the 2002 national SIAM meeting in Philadelphia (which is the last public disclosure that the authors are aware of) Google's Director of Technology Craig Silverstein went on record as saying that at that time it had no more effective way to deal with the global updating of PageRank other than by starting from scratch every three or four weeks. Furthermore, the PageRank vectors from prior periods were not used to determine PageRank for a current period. Google has, no doubt, made progress on this problem since 2002, but they are not talking. While local updating for popular sites seems to be more frequent, observation suggests that the time between complete global updates may still be an issue.

8.3. Experiments with power-law distributions. The scale-free nature of the Web translates into a power-law distribution of PageRanks—experiments described in [10, 34] indicate that PageRank has a power-law distribution whose parameter is $k \approx 2.1$. In other words, there are relatively very few pages that have a significant PageRank, while the overwhelming majority of pages have a nearly negligible PageRank.

Consequently, when PageRanks are plotted in order of decreasing magnitude, the resulting graph has a pronounced "L-shape" with an extremely sharp bend. It is this characteristic "L-shape" of PageRank distributions that reveals a near optimal partition $\mathcal{S} = G \cup \overline{G}$ for the iterative aggregation updating algorithm described in section 7, hereafter referred to as IAD.

To illustrate this point, we report on experiments derived from Web crawls pertaining to six specialized topics:

Topic	#Pages	#Links
Movies	451	713
MathWorks	517	13,531
Censorship	562	736
Abortion	1,693	4,325
Genetics	2,952	6,485
California	9,664	16,150

When the PageRanks for these datasets are plotted in order of decreasing magnitude, the characteristic L-shapes are apparent. A horizontal axis in Figure 1 represents components of a PageRank vector in order of decreasing magnitude, and on the vertical axis are the corresponding magnitudes of the PageRank.

In an attempt to discern characteristics of good partitions, several experiments were performed on these six datasets. After the initial PageRanks were computed,

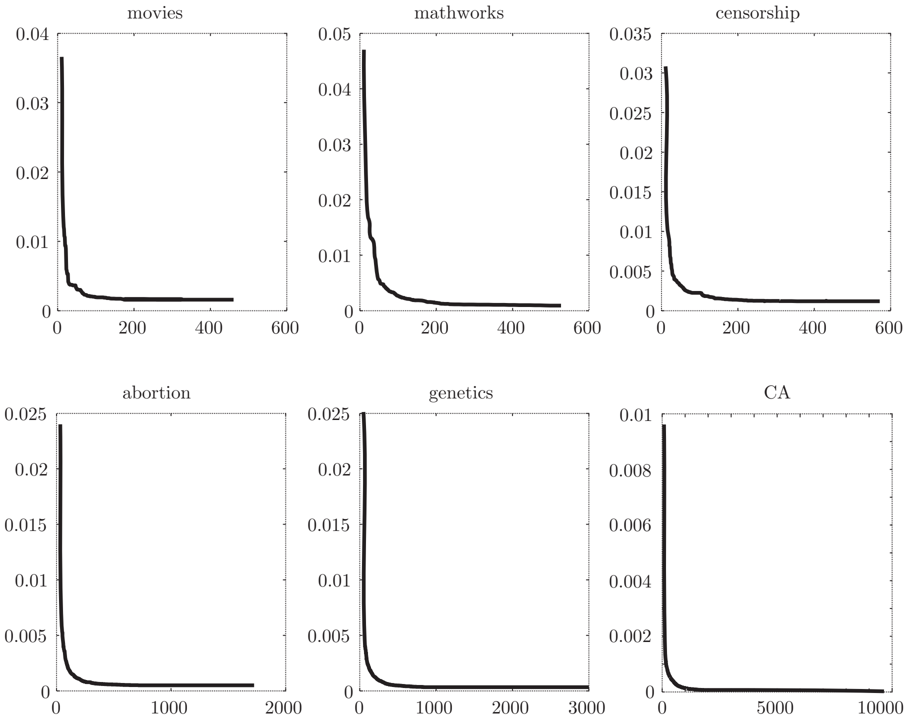

FIG. 1. *PageRanks for six datasets.*

each dataset was updated by adding some new pages and deleting some old ones, in addition to adding some new links and deleting some old links. Except for the larger California dataset, thirty new pages were added and twenty old ones were deleted, while fifty new links were created and twenty old ones were removed. For California, thirty new pages were added, three hundred old ones were removed, two hundred new links were added, and fifty old links were removed.[1] For each dataset, several partitions $\mathcal{S} = G \cup \overline{G}$ were constructed by first placing new states and the states that were altered (along with their nearest neighbors) into G, and then additional states were successively added to G in order of the magnitude of prior PageRanks. The IAD of section 7 was executed in MATLAB for each trial partition. Termination was always when the residual 1-norm dropped below 10^{-10}. The experimental results concerning iteration counts as well as total execution times (in seconds) are reported in Tables 1 and 2.

For the purpose of baseline comparisons, the iteration counts and execution times for the power method are given at the bottom of each table along with the relative improvement afforded by our IAD when the best observed G-partition was employed. A star ($\star$) in a table indicates that there was no experiment at the indicated size $g = |G|$ because g exceeds the number of pages in the dataset.

8.4. Experimental conclusions. While these experiments are small when compared to the scale of the entire World Wide Web, they nevertheless reveal some interesting patterns that are summarized below.

[1] Different values here had little effect on the performance of the algorithm.

Table 1
Experimental results for Movies, Censorship, and MathWorks.

	Movies ($g_{opt} \approx 50$)		MathWorks ($g_{opt} \approx 50$)		Censorship ($g_{opt} \approx 100$)	
$g = \lvert G\rvert$	Iterations	Time	Iterations	Time	Iterations	Time
5	23	.018	66	.362	41	.062
10	13	.017	64	.329	41	.046
15	13	.016	64	.332	40	.045
20	13	.015	50	.295	20	.024
25	12	.016	46	.291	20	.025
50	12	.015*	19	.207*	13	.019
100	10	.016	18	.224	9	.017*
200	9	.018	16	.284	9	.019
300	9	.021	11	.265	9	.022
400	7	.021	8	.278	9	.026
Power method	22	.017	69	.255	42	.031
*Improvement		11.8%		18.8%		45.2%

Table 2
Experimental results for Abortion, Genetics, and California.

	Abortion ($g_{opt} \approx 250$)		Genetics ($g_{opt} \approx 250$)		California ($g_{opt} \approx 2000$)	
$g = \lvert G\rvert$	Iterations	Time	Iterations	Time	Iterations	Time
10	165	.773	163	2.16	170	7.75
50	58	.256	19	.483	75	3.56
100	14	.159	19	.456	57	3.75
250	13	.140*	17	.276*	51	2.59
500	7	.199	9	.313	34	2.01
1000	7	.194	8	.319	19	1.03
2000	$\star$	$\star$	6	.393	10	.997*
3000	$\star$	$\star$	$\star$	$\star$	7	1.17
4000	$\star$	$\star$	$\star$	$\star$	7	1.22
5000	$\star$	$\star$	$\star$	$\star$	7	1.56
Power method	168	.449	165	1.45	176	5.87
*Improvement		68.8%		81%		83%

- G-sets can always exist for which the iterative aggregation technique provides a significant improvement over the power method.
- The improvements become more pronounced as the size of the datasets increases.
- Good G-sets can be constructed by including states affected by updated information along with a few states that are associated with the largest stationary probabilities from the preupdated distribution.
- As $g = |G|$ increases, the performance of IAD (as measured by execution time) improves up to some point, but increasing g beyond this point degrades the

performance of IAD. How to determine a priori a nearly optimal size for G is discussed in section 8.5.

- Finally, when IAD is used as an updating technique, the fact that updates might change the problem size is of little or no consequence. This is an extremely important feature because dealing with issues caused by adding or deleting states is generally a major problem for most updating applications.

8.5. Near optimal G-sets and L-curves. As observed above, IAD performs well by using a G-set that includes states affected by updated information along with states that are associated with the largest stationary probabilities from the preupdated distribution. But since the performance of IAD is dependent on the size of such a G, it is important to have a mechanism that gauges how many of the largest states from the preupdated distribution should be included in G. By examining the approximate values of g_{opt} given in Tables 1 and 2, it is clear that there is somewhat of a pattern that relates g_{opt} to the shape of the L-curve for the power-law distribution. To see this, superimpose the respective values of g_{opt} given in Tables 1 and 2 on the L-curves given in Figure 1. The results are shown in Figure 2.

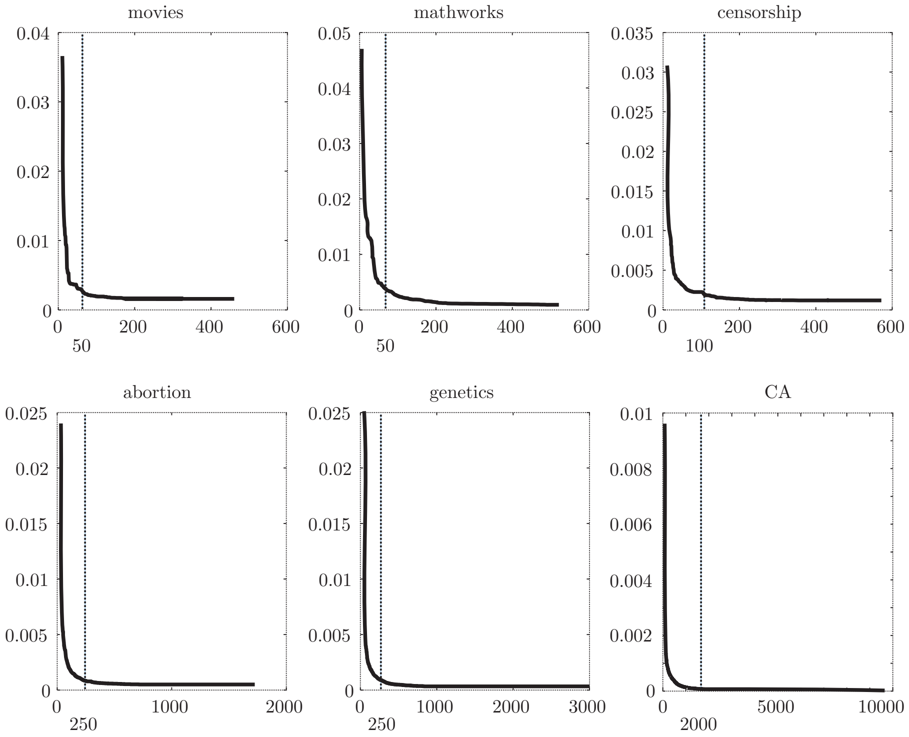

FIG. 2. *Location of* g_{opt}.

These graphs suggest that when the stationary probabilities of a Markov chain have a power-law distribution, the size of the G-set used in IAD is nearly optimal around a point that is just to the right of the pronounced bend in the L-curve. In other words, an apparent method for constructing a reasonably good partition $\mathcal{S} = G \cup \overline{G}$ for IAD updating is as follows.

1. First put all new states and states with altered links (perhaps along with some nearest neighbors) into G.

2. Add other states that remain after the update in order of the magnitude of their prior stationary probabilities up to the point where these stationary probabilities level off.

Of course, there is some subjectiveness to this strategy. However, the leveling-off point is relatively easy to discern in distributions having a sharply defined bend in the L-curve, and only distributions that gradually die away or do not conform to a power-law distribution are problematic. For example, there is more uncertainty in choosing the leveling-off point in the three smaller test cases (movies, censorship, and MathWorks) than in the three larger ones (abortion, genetics, and California), but being somewhere in the ballpark is generally good enough even when the bend is not so sharply defined. The results in Tables 1 and 2 indicate how much variation around g_{opt} can be tolerated without seriously affecting IAD performance, and in all cases there is a fair amount of leeway.

If, when ordered by magnitude, the stationary probabilities

$$\pi(1) \geq \pi(2) \geq \cdots \geq \pi(n)$$

for an irreducible Markov chain conform to a power-law distribution so that there are constants $\alpha > 0$ and $k > 0$ such that $\pi(i) \approx \alpha i^{-k}$, then the "leveling-off point" i_{level} can be taken to be the smallest value for which $|d\pi(i)/di| \approx \epsilon$ for some user-defined tolerance ϵ. That is, $i_{level} \approx (k\alpha/\epsilon)^{1/k+1}$. This provides a rough estimate of g_{opt}, but empirical evidence suggests that better estimates require a scaling factor $\sigma(n)$ that accounts for the size of the chain; i.e.,

$$g_{opt} \approx \sigma(n) \left(\frac{k\alpha}{\epsilon}\right)^{1/k+1} \approx \sigma(n) \left(\frac{k\pi(1)}{\epsilon}\right)^{1/k+1}.$$

If this is the case, and if the observations from [10, 34] are correct in the sense that PageRanks for the entire World Wide Web conform to a power-law distribution with parameter $k = 2.109$, then we should have that

$$g_{opt} \approx \sigma(n) \left[\frac{2.109\pi(1)}{\epsilon}\right]^{1/3.109}.$$

However, validating this conclusion with a specific scaling factor is beyond the scope of our data and computational resources.

8.6. The drop-off point. Our experiments also indicate that the number of IAD iterations required is a nonlinear function of the size of the G-set. As $g = |G|$ increases, as described in section 8.3, there is initially a sharp drop in the number of IAD iterations required. After the sharp drop there is a more moderate and steady decrease in the iteration count as a function of g. But an even more interesting feature of this phenomenon is that *the sharp drop in iteration count occurs at a point that is more or less independent of the number of nodes or links that are updated.*

For example, the graphs in Figure 3 plot the number of IAD iterations against $g = |G|$ for the MathWorks dataset used in section 8.3 when the updating involves 10, 50, and 100 nodes and links. Regardless of the number of updates, all graphs in Figure 3 exhibit roughly the same shape, with a drop-off point around 40 or 50, which is also about the same value for g_{opt} that was determined from the experiments in Table 1. This phenomenon was generally observed in our other experiments as well.

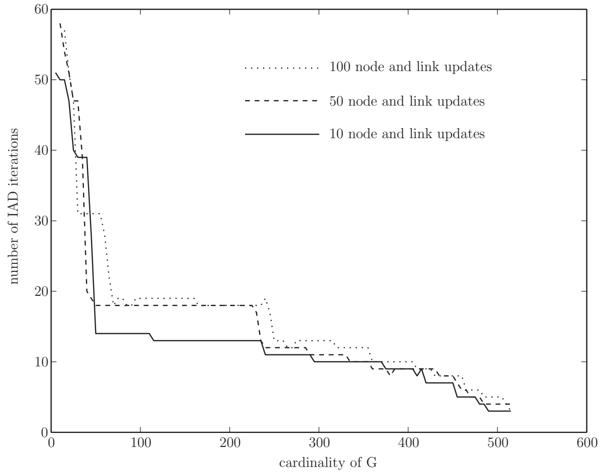

Fig. 3. *Drop-off points.*

Consequently, this suggests that the value of g_{opt} can alternately be characterized by saying that g_{opt} *is approximately equal to the drop-off point in the iteration count.*

Theorem 7.1 established that the asymptotic rate of convergence of IAD is governed by the subdominant eigenvalue λ_2 of the one significant stochastic complement $\mathbf{S}$. Therefore, the drop-off point in the iteration count should be explainable in terms of a drop-off in the magnitude of λ_2 as $g = |G|$ is increased. Indeed, this is corroborated by our experiments. For example, by again using the MathWorks data as a typical case, it is seen in Table 3 that as g increases, the value of $|\lambda_2|$ decreases rapidly until g is somewhere around 40 or 50, after which there is only a slow decrease. And this more or less agrees with the value of g_{opt} that was observed from the data in Table 1.

Table 3
$|\lambda_2|$ *as a function of* $g = |G|$.

g	10	20	30	35	38	40	45	50	100
$\lvert\lambda_2\rvert$	.7206	.6891	.6610	.6054	.4431	.4018	.4012	.4005	.3857

9. Conclusions and future work. An algorithm for updating the stationary vector of a Markov chain subject to changes in the number of states as well as changes to the transition probabilities has been introduced and analyzed. This IAD updating algorithm exploits the old stationary vector to create the new stationary vector, and numerical experiments suggest it is quite effective. It is superior to the power method in terms of both iteration count as well as total execution time. The IAD approach offers room for even greater improvements. For example, the extrapolation technique introduced in [19] can be employed in conjunction with the IAD introduced in this article to further accelerate the updating process. Preliminary experiments indicate that marrying IAD to extrapolation has remarkable promise—results will be reported in a separate article. Finally, we have demonstrated the applicability of IAD to updating Google's PageRank vector.

Acknowledgments. We thank Ronny Lempel for generously supplying the three datasets on movies, abortion, and genetics, and we thank Cleve Moler for sharing his MathWorks dataset as well as other Web-crawling M-files.

REFERENCES

[1] A.-L. Barabasi, *Linked: The New Science of Networks*, Plume, New York, 2003.

[2] A.-L. Barabasi, R. Albert, and H. Jeong, *Scale-free characteristics of random networks: The topology of the World-Wide Web*, Phys. A, 281 (2000), pp. 69–77.

[3] S. Brin and L. Page, *The anatomy of a large-scale hypertextual web search engine*, Comput. Networks ISDN Systems, 33 (1998), pp. 107–117.

[4] S. Brin, L. Page, R. Motwami, and T. Winograd, *The PageRank Citation Ranking: Bringing Order to the Web*, Technical report, Computer Science Department, Stanford University, Stanford, CA, 1998.

[5] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, *Graph structure in the web*, in Proceedings of the Ninth International World Wide Web Conference, 2000; available online at http://www9.org/w9cdrom/160/160.html.

[6] S. Campbell and C. D. Meyer, *Generalized Inverses of Linear Transformations*, Pitman, San Francisco, 1979.

[7] S. Chien, C. Dwork, R. Kumar, and D. Sivakumar, *Towards exploiting link evolution*, in Proceedings of the Workshop on Algorithms and Models for the Web Graph, 2001.

[8] G. E. Cho and C. D. Meyer, *Markov chain sensitivity measured by mean first passage times*, Linear Algebra Appl., 313 (2000), pp. 21–28.

[9] G. E. Cho and C. D. Meyer, *Comparison of perturbation bounds for the stationary distribution of a Markov chain*, Linear Algebra Appl., 335 (2001), pp. 137–150.

[10] D. Donato, L. Laura, S. Leonardi, and S. Millozzi, *Large scale properties of the webgraph*, Eur. Phys. J. B, 38 (2004), pp. 239–243.

[11] R. E. Funderlic and C. D. Meyer, *Sensitivity of the stationary distribution vector for an ergodic Markov chain*, Linear Algebra Appl., 76 (1986), pp. 1–17.

[12] R. E. Funderlic and R. J. Plemmons, *Updating LU factorizations for computing stationary distributions*, SIAM J. Alg. Disc. Meth., 7 (1986), pp. 30–42.

[13] G. H. Golub and C. F. V. Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1996.

[14] G. H. Golub and C. D. Meyer, *Using the QR factorization and group inversion to compute, differentiate, and estimate the sensitivity of stationary probabilities for Markov chains*, SIAM J. Alg. Disc. Meth., 7 (1986), pp. 273–281.

[15] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1990.

[16] J. J. Hunter, *Stationary distributions of perturbed Markov chains*, Linear Algebra Appl., 82 (1986), pp. 201–214.

[17] I. C. F. Ipsen and S. Kirkland, *Convergence analysis of an improved PageRank algorithm*, Technical report CRSC-TR04-02, North Carolina State University, Raleigh, NC, 2004.

[18] I. C. F. Ipsen and C. D. Meyer, *Uniform stability of Markov chains*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 1061–1074.

[19] 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, 2003; available online at http://www2003.org/cdrom/index.html.

[20] J. G. Kemeny and L. J. Snell, *Finite Markov Chains*, Van Nostrand, New York, 1960.

[21] A. N. Langville and C. D. Meyer, *A survey of eigenvector methods for Web information retrieval*, SIAM Rev., 47 (2005), pp. 135–161.

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

[23] C. D. Meyer Jr., *The role of the group generalized inverse in the theory of finite Markov chains*, SIAM Rev., 17 (1975), pp. 443–464.

[24] C. D. Meyer Jr., *The condition of a finite Markov chain and perturbation bounds for the limiting probabilities*, SIAM J. Alg. Disc. Meth., 1 (1980), pp. 273–283.

[25] C. D. Meyer, *Analysis of finite Markov chains by group inversion techniques*, in Recent Applications of Generalized Inverses, Res. Notes in Math. 66, S. L. Campbell, ed., Pitman, Boston, London, 1982, pp. 50–81.

[26] C. D. Meyer, *Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems*, SIAM Rev., 31 (1989), pp. 240–272.

[27] C. D. Meyer, *The character of a finite Markov chain*, in Linear Algebra, Markov Chains, and Queueing Models, IMA Vol. Math. Appl. 48, C. D. Meyer and R. J. Plemmons, eds., Springer-Verlag, New York, 1993, pp. 47–58.

[28] C. D. Meyer, *Sensitivity of the stationary distribution of a Markov chain*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 715–728.

[29] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.

[30] C. D. Meyer and R. J. Plemmons, *Linear Algebra, Markov Chains, and Queueing Models*, Springer-Verlag, New York, 1993.

[31] C. D. Meyer and J. M. Shoaf, *Updating finite Markov chains by using techniques of group matrix inversion*, J. Statist. Comput. Simulation, 11 (1980), pp. 163–181.

[32] C. D. Meyer and G. W. Stewart, *Derivatives and perturbations of eigenvectors*, SIAM J. Numer. Anal., 25 (1988), pp. 679–691.

[33] C. Moler, *The world's largest matrix computation*, MATLAB News and Notes, October (2002), pp. 12–13.

[34] G. Pandurangan, P. Raghavan, and E. Upfal, *Using PageRank to characterize Web structure*, in Computing and Combinatorics, Lecture Notes in Comput. Sci. 2387, Springer-Verlag, Berlin, 2002.

[35] E. Seneta, *Nonnegative Matrices and Markov Chains*, Springer-Verlag, New York, 1981.

[36] E. Seneta, *Sensitivity analysis, ergodicity coefficients, and rank-one updates for finite Markov chains*, in Numerical Solution of Markov Chains, W. J. Stewart, ed., Dekker, New York, 1991, pp. 121–129.

[37] H. A. Simon and A. Ando, *Aggregation of variables in dynamic systems*, Econometrica, 29 (1961), pp. 111–138.

[38] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.