Updating PageRank with Iterative Aggregation

Amy N. Langville
N.C. State University
Mathematics Department
Raleigh, NC 27695-8205
anlangvi@unity.ncsu.edu

Carl D. Meyer
N.C. State University
Mathematics Department
Raleigh, NC 27695-8205
meyer@math.ncsu.edu

ABSTRACT

We present an algorithm for updating the PageRank vector [1]. Due to the scale of the web, Google only updates its famous PageRank vector on a monthly basis. However, the Web changes much more frequently. Drastically speeding the PageRank computation can lead to fresher, more accurate rankings of the webpages retrieved by search engines. It can also make the goal of real-time personalized rankings within reach. On two small subsets of the web, our algorithm updates PageRank using just 25% and 14%, respectively, of the time required by the original PageRank algorithm. Our algorithm uses iterative aggregation techniques [7, 8] to focus on the slow-converging states of the Markov chain. The most exciting feature of this algorithm is that it can be joined with other PageRank acceleration methods, such as the dangling node lumpability algorithm [6], quadratic extrapolation [4], and adaptive PageRank [3], to realize even greater speedups (potentially a factor of 60 or more speedup when all algorithms are combined).

Categories and Subject Descriptors

F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General

General Terms

Algorithms, Performance

Keywords

PageRank, updating, link analysis, power method, aggregation, disaggregation, Markov chains, stationary vector

1. INTRODUCTION

We have discovered a new technique that can drastically affect (and perhaps completely crack) one of the major bottlenecks associated with web-based information retrieval systems that are driven by eigenvector ranking schemes—the primary example is the PageRank mechanism that drives Google. The bottleneck is the need to update importance rankings of pages to account for the continual changes that occur in the web's structure when pages are added or deleted and links are created or destroyed. At last report, Google uses several days for this computation (because they use brute force and start from scratch each time an update is attempted). Consequently, updating can only be afforded every few weeks. Our solution harnesses the power of iterative aggregation principles for Markov chains to allow for much more frequent updates to the valuable ranking vectors.

WWW2004, May 17–22, 2004, New York, NY USA.
ACM 1-58113-912-8/04/0005.

2. THE UPDATING ALGORITHM

Our primary goal is to adapt the theory of exact [7] and approximate aggregation [8] to efficiently solve the updating problem. Suppose the Markov transition matrices and distributions at times t and $t+1$ are respectively given by

$$\mathbf{Q}_{m\times m} \text{ and } \boldsymbol{\Phi}^T = (\phi_1, \phi_2, \ldots, \phi_m) \text{ at } time = t$$
$$\mathbf{P}_{n\times n} \text{ and } \boldsymbol{\pi}^T = (\pi_1, \pi_2, \ldots, \pi_n) \text{ at } time = t+1.$$

Quantities $\mathbf{Q}$, $\mathbf{P}$, and $\boldsymbol{\Phi}^T$ are known while $\boldsymbol{\pi}^T$ is unknown, and $m \neq n$ because states may be added or deleted. Partition (and perhaps reorder) the state space $\mathcal{S} = G \cup \overline{G}$ for the chain at time $t+1$ so that $\mathbf{P}$ has the partitioned form

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

All newly added states go into G along with some of the preexisting states. The idea is to leave the g states in G unaggregated while the $n-g$ states in $\overline{G}$ are aggregated into a single superstate. Let $\{\phi_i\}_{i=g+1}^n$ be conformably ordered with $\{\pi_i\}_{i=g+1}^n$, and approximate (what, in the theory of stochastic complementation [7], is known as) the censored distribution $\mathbf{s}_2^T$ with

$$\mathbf{s}_2^T \approx \widetilde{\mathbf{s}}_2^T = \frac{(\phi_{g+1}, \ldots, \phi_n)}{\sum_{i=g+1}^n \phi_i},$$

so the exact aggregated transition matrix (of stochastic complementation) is approximated by the $(g+1)\times(g+1)$ matrix

$$\mathbf{A} \approx \widetilde{\mathbf{A}} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12}\mathbf{e} \\ \widetilde{\mathbf{s}}_2^T\mathbf{P}_{21} & 1 - \widetilde{\mathbf{s}}_2^T\mathbf{P}_{21}\mathbf{e} \end{pmatrix}$$

where $\mathbf{e}$ is the vector of all ones. If $\widetilde{\boldsymbol{\alpha}}^T = (\,\widetilde{\alpha}_1, \ldots, \widetilde{\alpha}_g, \widetilde{\alpha}_{g+1}\,)$ is the stationary distribution of $\widetilde{\mathbf{A}}$, then the aggregation theorem [7] yields an approximation tothe updated distribution,

$$\boldsymbol{\pi}^T \approx \widetilde{\boldsymbol{\pi}}^T = \left(\, \widetilde{\alpha}_1, \ldots, \widetilde{\alpha}_g \,|\, \widetilde{\alpha}_{g+1}\widetilde{\mathbf{s}}_2^T \,\right).$$

This approximation is further refined with a smoothing step $\widetilde{\boldsymbol{\pi}}^T\mathbf{P} = (\mathbf{x}^T \,|\, \mathbf{y}^T)$, where $(\mathbf{x}^T \,|\, \mathbf{y}^T)$ is a vector partitioned according to the G and $\overline{G}$ sets. Then the process is iterated by restarting the procedure with

$$\widetilde{\mathbf{s}}_2^T \longleftarrow \mathbf{y}^T/\mathbf{y}^T\mathbf{e}.$$

Interestingly, this procedure always converges to the PageRank vector.

Successful implementation of these ideas hinges on the ability to identify an optimal choice for the partition $\mathcal{S} = G \cup \overline{G}$, and this is a main facet of future research. Extensive testing seems to be the best way to produce practical heuristics for determining the appropriate partition for a given dataset. In fact, on several small datasets, we have experiments showing the numerical feasibility of the updating algorithm (see section 2.2).

2.1 Convergence

References [2, 5] prove that (1) this updating algorithm converges to the PageRank vector for all partitions $\mathcal{S} = G \cup \overline{G}$, and (2) there always exists a partition such that the convergence rate of the updating algorithm is strictly less than the convergence rate of the Google's power method.

3. RESULTS

We have experimented with a variety of datasets that were extracted as subsets of the Web. However, we describe just two case studies with typical characteristics and outcomes. `NCstate.dat` contains 10,000 pages obtained from a crawl that started with the NCSU homepage. This small web has $n = 10,000$ pages and $l = 101,118$ links. `California.dat` is a topical net of $n = 9664$ pages pertaining to the query topic of "California." It has $l = 16,150$ links.

Tables 1 and 2 compare the aggregation updating algorithm with Google's current method for updating PageRank, which is called full recomputation since the power method is started from scratch. We report the number of iterations and the total computation time required by each method.

Table 1: Comparison of updating methods on `NCstate.dat` ($n = 10,000$, $l = 101,118$)

	Iterative Aggregation		Full Recomputation	
$\lvert G\rvert$	Iterations	Time	Iterations	Time
500	160	16.64		
1000	51	6.47		
1500	33	4.57		
2000	21	3.64		
2500	**16**	**3.19**	**162**	**13.17**
3000	13	3.26		
5000	7	3.62		

Table 2: Comparison of updating methods on `California.dat` ($n = 9,664$, $l = 16,150$)

	Iterative Aggregation		Full Recomputation	
$\lvert G\rvert$	Iterations	Time	Iterations	Time
500	38	1.82		
1000	28	2.47		
1250	28	2.61		
1500	**14**	**1.42**	**176**	**9.63**
2000	13	1.57		
5000	10	1.65		

These tables show the speedups (as much as a factor of 10, on some other datasets) that are obtainable with this updating method. In effect, most of the work done by the updating algorithm is done on the very small aggregation matrix of size $|G| + 1$. For example, for `California.dat` with $|G| = 1500$, the aggregation algorithm converges in just 14 iterations and 1.42 seconds compared to the 176 iterations and 9.63 seconds required by the power method because most of the work was done on a $1,501 \times 1,501$ matrix, rather than a $9,664 \times 9,664$ matrix. Tables 1 and 2 also show the crucial role that $|G|$ plays in the speedups achieved, and thus, choosing $|G|$ becomes an issue. Examine Figure 1. The upper pane shows the norm of the residual vector for the updating algorithm applied to another dataset `abortion.dat` with $|G| = 50$, which creates a good choice for G and provides a factor of 6 speedup. The lower pane shows the norm of the residual vector for the same dataset with $|G| = 5$, which creates a bad partition and causes the updating algorithm to take nearly as much time as the notoriously slow power method. The slippery problem of choosing a good partition of the chain's states is an active area of research.

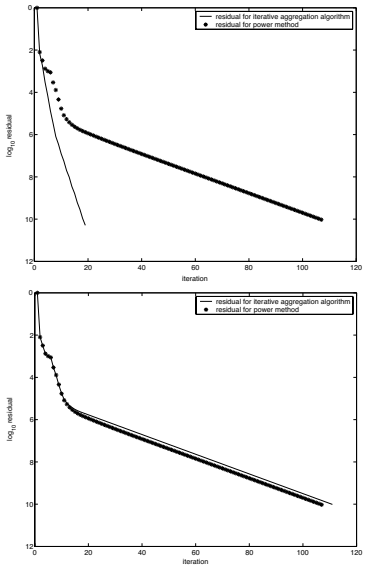

Figure 1: Norm of residual vector for `abortion.dat` for $|G| = 50$ (upper) and $|G| = 5$ (lower)

4. FUTURE WORK

By the WWW 2004 conference, we hope to have: (1) tested this updating algorithm on larger datasets, such as those used in [3, 4, 6], (2) made progress toward understanding and determining the partitioning of states into G and $\overline{G}$, and (3) created code that incorporates this updating algorithm with the other PageRank acceleration techniques.

5. REFERENCES

[1] 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, 1998.

[2] I. C. F. Ipsen and S. Kirkland. Convergence analysis of an improved PageRank algorithm. December 2003.

[3] S. D. Kamvar, T. H. Haveliwala, and G. H. Golub. Adaptive methods for the computation of pagerank. Technical report, Stanford University, 2003.

[4] S. D. Kamvar, T. H. Haveliwala, C. D. Manning, and G. H. Golub. Extrapolation methods for accelerating pagerank computations. Twelfth International World Wide Web Conference, 2003.

[5] A. N. Langville and C. D. Meyer. Updating the stationary vector of an irreducible Markov chain. Technical Report crsc02-tr33, N. C. State, Mathematics Dept., CRSC, 2002.

[6] C. P.-C. Lee, G. H. Golub, and S. A. Zenios. Partial state space aggregation based on lumpability and its application to pagerank. Technical report, Stanford University, 2003.

[7] C. D. Meyer. Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Review*, 31(2):240–272, 1989.

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