Inf Retrieval (2006) 9: 123–138
DOI 10.1007/s10791-006-7146-1

Efficient PageRank approximation via graph aggregation

A. Z. Broder* · **R. Lempel** · **F. Maghoul** · **J. Pedersen**

Received: 28 March 2004 / Revised: 30 September 2004 / Accepted: 28 January 2005

Abstract We present a framework for approximating random-walk based probability distributions over Web pages using graph aggregation. The basic idea is to partition the graph into classes of quasi-equivalent vertices, to project the page-based random walk to be approximated onto those classes, and to compute the stationary probability distribution of the resulting class-based random walk. From this distribution we can quickly reconstruct a distribution on pages. In particular, our framework can approximate the well-known PageRank distribution by setting the classes according to the set of pages on each Web host.

We experimented on a Web-graph containing over 1.4 billion pages and over 6.6 billion links from a crawl of the Web conducted by AltaVista in September 2003. We were able to produce a ranking that has Spearman rank-order correlation of 0.95 with respect to PageRank. The clock time required by a simplistic implementation of our method was less than half the time required by a highly optimized implementation of PageRank, implying that larger speedup factors are probably possible.

Keywords Web IR · Citation and link analysis

* Significant portions of the work presented here were done while A. Broder and R. Lempel were employed by the AltaVista corporation.

A. Z. Broder
IBM T.J. Watson Research Center
e-mail: abroder@us.ibm.com

R. Lempel (✉)
IBM Research Lab, Haifa, Israel
e-mail: rlempel@il.ibm.com

F. Maghoul · J. Pedersen
Yahoo! Inc.
e-mail: {fmaghoul; jpederse}@yahoo-inc.com

1. Introduction

Since the late nineties, Web search engines have started to rely more and more on off-page, Web-specific data such as link analysis, anchor-text, and click-through data. *Google* (`www.google.com`) was the first engine to use link analysis as a primary ranking factor and the now-defunct DirectHit[1] concentrated on click-through data. By now, all major engines exploit all these types of data. In particular, link analysis and anchor-text seem crucial for handling navigational queries (Broder, 2002).

One particular form of link-based ranking factors are *static* scores, which are query-independent importance scores that are assigned to all Web pages. The most famous algorithm for producing such scores is *PageRank*, devised by Brin and Page (1998) while developing the ranking module for the prototype of *Google.* The basic intuition behind PageRank, derived from classic IR concepts in bibliometrics, is that a web page (or a scientific paper) is "important" if many other pages (resp. other scientific papers) link to it (resp. cite it). But not all links or citations are equal: links from more important pages should count more. We are thus led to a recursive definition of importance, that can be formalized as the solution of a certain system of linear equations. Alternatively, PageRank can be described as the stationary probability distribution of a certain random walk on the Web graph, that is, the graph whose nodes are the Web pages, and whose directed edges are the links between them. We formalize this discussion in Section 2.1.

There is a large PageRank related literature, sometimes showing applications not directly related to ranking. Cho et al. (1998), for example, use PageRank to prioritize crawlers and Henzinger et al. (1999) use PageRank to measure the quality of a crawled corpus. The *voting model* proposed by Lifantsev (2000) can also be seen as a general and extensible variant of PageRank. Numerical properties of PageRank have been studied in Pandurangan et al. (2002), Ng et al. (2001), Lempel and Moran (2001), Lee (2002), Chien et al. (2002) and topic-sensitive and personalized versions of PageRank were described in Haveliwala, (2002), Haveliwala et al. (2003), Jeh and Widom (2003), Richardson and Domingos (2001). We discuss some of this literature in more detail in Section 2.1.

Modern search engines index billions of pages, interconnected by tens of billions of links. Computing PageRank on Web graphs of this scale requires considerable computational resources, both in terms of CPU cycles and in terms of random-access memory. Given the importance of PageRank in ranking search results, it is not surprising that there has been considerable interest in schemes that accelerate PageRank type computations (Haveliwala, 1999; Kamvar et al., 2003a,c; Abiteboul et al., 2003).

To some extent our work follows in this vein. We use a framework for computing PageRank-like probability distributions for Web pages based on graph aggregation. The basic idea is to partition the graph into *classes* of quasi-equivalent vertices and to compute the stationary probability distribution of a biased random walk on classes. From this distribution we can reconstruct a distribution on pages. More concretely, in the context of this paper, the classes used were the sets of pages on a given host. The random walk associated with PageRank is decomposed into intra-host and inter-host steps. In our algorithm, inter-host steps are governed by the hyperlinks connecting pages of different hosts; intra-host steps are not necessarily driven by the underlying graph, but instead are governed by a fixed distribution. Thus we obtain only an approximation of PageRank rather than the exact PageRank

[1] Acquired by Ask Jeeves, www. ask.com

values. This is related to the ideas presented in (Kamvar et al., 2003b), and Section 3 expands on the similarities and differences between our approach and that of Kamvar et al. (2003b).

As with any new proposed static ranking, one must examine the appropriateness of the scores to the task at hand—achieving high quality Web search. We tackle this evaluation task by statistically comparing the scores produced by our approach with those produced by PageRank, demonstrating that we are able to construct a good approximation of PageRank. In our experiments, based on a Web-graph containing over 1.4 billion pages and over 6.6 billion links from a crawl of the Web conducted by AltaVista, we are able to produce a ranking that has Spearman rank-order correlation (Snedecor and Cochran, 1989) of 0.95 with respect to PageRank.

In terms of use for ranking of Web search results, we do not know whether the (small) differences between our model and the original PageRank are for better or for worse. Our model is much less sensitive to changes in the internal organization of hosts, which might be an advantage.

The algorithm spends most of its running time finding a stationary distribution on the Web host graph, that is, the graph that has an edge from host h_1 to host h_2 whenever there is a link from some page on h_1 to some page in h_2. Nevertheless, it ultimately still derives static scores with page granularity, that is, pages on a given host will generally be assigned different scores. Since the Web's host graph is significantly smaller than the Web's page graph (by factors of 20 and beyond in the number of edges), the algorithm scales well. The clock time required by a simplistic implementation of our method was less than half the time required by a highly optimized implementation of PageRank. We expect that refined implementations will be able to yield speed-up factors that are closer to the ratio between the sizes of the Web's page and host graphs. Furthermore, optimization techniques for PageRank such as (Haveliwala, 1999; Kamvar et al., 2003c) might be able to speed up our approach as well.

The rest of this paper is organized as follows: Section 2 starts by reviewing the PageRank algorithm, and then moves on to survey previous work on optimizing PageRank's computation process, and discussions of "flavored" PageRank variants. It recounts alternative static rank approaches that have been proposed, and reports on some studies of the host-graph of the Web. Section 3 presents the details of our scheme, and compares its complexity with that of the original PageRank algorithm. Section 4 covers our experiments with a specific graph-aggregated PageRank approximation. We concretely define the approximation flavor, provide performance figures and compare the resulting score vector with the PageRank score vector. Section 5 concludes, and points out directions for future research.

2. Related work

2.1. PageRank and variations

PageRank (Brin and Page, 1998) is an important part of the ranking function of the Google search engine. The PageRank of a page p is the probability of visiting p in a random walk of the entire Web, where the set of states of the random walk is the set of Web pages, and each random step is of one of the following two types:

Browsing step: from the given state/page q, choose an outgoing link of q uniformly at random, and follow that link to the destination page.
Teleportation: choose a Web page uniformly at random, and jump to it.

PageRank chooses a parameter d, $0 < d < 1$; each state transition is a browsing step with probability d, or a teleportation step with probability $1 - d$.[2]

PageRank requires teleportations (jumps to random Web pages) since the Markov chain that is implied by the link-structure of the Web is separable rather than ergodic. In particular, a large-scale study of the Web graph (Broder et al., 2000) suggests that while there is a *core* of pages which forms a strongly connected component in the graph, a majority of Web pages have directed paths of links either to the core or from it, but not both. However, incorporating random jumps introduces a (small) probability of transition from any page a to any page b, even in absence of a Web link $a \to b$, thus giving rise to an ergodic Markov chain that has a well-defined stationary distribution (Gallager, 1996). Furthermore, the set of PageRank scores obey the following formula (where page p has incoming links from $q_1, \ldots, q_k$, and N is the total number of Web pages):

$$\text{PageRank}(p) = \frac{1-d}{N} + d\left(\sum_{i=1}^{k} \frac{\text{PageRank}(q_i)}{\text{out degree of } q_i}\right)$$

The above set of equations is easily written in matrix form, with the stationary distribution (the vector of PageRank scores) simply being the principal eigenvector of the corresponding stochastic matrix. The PageRank scores are typically computed by applying the Power method for approximating the principal eigenvector of a matrix (Jennings, 1977). The method involves repeated multiplications of an arbitrary initial vector by the matrix in question, until the iterations converge to a fixed vector.

2.1.1. Numerical properties of PageRank

Pandurangan et al. (2002) have studied the distribution of PageRank scores across several independent Web subgraphs, and noted that it follows a power-law. Furthermore, the exponent of the distribution was found to be 2.1—similar to the observed power-law exponent of the distribution of in-degrees of Web pages (Barabasi and Albert, 1999; Kleinberg et al., 1999; Broder et al., 2000). Despite the similarity in distributions, they showed (on two small Web subgraphs) that PageRank is not highly correlated with indegree. On the other hand, Upstill et al. (2003) found log indegree to be highly correlated with the PageRank values reported by Google's toolbar. Thus, the measure of correlation between PageRank and in-degree (or functions thereof) on the Web at large is unclear; it is, however, possible to construct artificial graphs where most nodes of high in-degree have lower PageRanks than most nodes with low-indegree (Lempel and Moran, 2001).

Several researchers have examined the numerical stability of the PageRank scores in terms of their sensitivity to perturbations of the underlying graph. Ng et al. (2001) examined the L_1-change to the scores when modifying the outgoing links of a set P of pages. They bounded this change by a linear function of the aggregate PageRanks of all pages in P. Lee (Lee, 2002) argued that an algorithm is stable if the L_1 change in its score vector following perturbations is bounded by a linear function of the sum of the scores of the perturbed nodes. For that definition, he showed that PageRank is stable for all graphs. Neither of these works examined how the perturbations affect the rankings that are induced by the score vectors.

[2]Note that this implicitly assumes that each Web page has at least one outgoing link. This is not true in general, and so most discussions of PageRank assume that transitions from pages with no outgoing links are teleportations with probability 1 (as if those pages actually linked to all pages).

That aspect was looked upon by Chien et al. (2002), who showed that when a link is added to a graph, (1) the PageRank of the node receiving the link rises, and (2) the same node's rank cannot decrease with respect to its rank prior to the change. However, in Lempel and Moran (2001) is was shown that by changing the destination of a single link in an arbitrarily large graph, the relative rankings of many pairs of nodes (about a quarter of all pairs) may flip.

2.1.2. Towards personalized PageRank

Brin and Page, when presenting PageRank, noted that it is possible to obtain topic-oriented flavors of PageRank by biasing the random jumps of the algorithm to favor topical pages as destinations, rather than jumping uniformly at random over all Web pages (Brin and Page, 1998). This idea was expanded by Haveliwala (2002), where 16 precomputed topic-sensitive flavors of PageRank (corresponding to the 16 top-level categories of the ODP[3]) were used to improve rankings of queries in real-time. Each PageRank flavor was computed by distributing its random jumps uniformly across the pages belonging to the corresponding ODP top-level category. Then, at runtime, a linear combination of those flavors was used for ranking, where the combination's coefficients are determined by the similarity of the query (and additional context, if available) to the contents of the pages in each category. Jeh and Widom (2003) discuss personalized PageRank flavors where random jump configurations are more flexible. They assume that there is a set H of pages (which may contain several thousand pages), such that each personalized flavor may choose any distribution of random jumps over H. See Haveliwala et al. (2003) for additional details on the above approaches.

Richardson and Domingos (2001) proposed computing a PageRank flavor for each term in the lexicon of a search engine's index. For each such term, their formulation biases both the random jump probabilities and the link-following steps towards pages whose relevance score with respect to the term is high. They report gains of about 20% in search quality.

2.1.3. Accelerations of PageRank

As mentioned in the Introduction, Indices of modern search engines contain billions of pages, interconnected by tens of billions of links. Performing PageRank computations on data of this scale requires considerable resources, both in terms of CPU cycles and in terms of random-access memory. Clock-wall times for PageRank computations on large graphs can reach many hours (certainly in single-machine settings). When considering that the topic-induced or personalized PageRank flavors (discussed in the previous section) require that a search engine perform such computations multiple times, the need for speedy implementations becomes critical. Consequently, several papers have described methods for accelerating PageRank computations.

Haveliwala shows how PageRank computations can be adapted to run on machines with limited RAM (Haveliwala, 1999). The performance gains are due to RAM-aware partitioning of the score vectors and connectivity data that are used during the Power iterations. Certain hardware configurations are shown where speedup factors reach three orders of magnitude. Chen et al. (2002) addressed RAM limitations by applying techniques of out-of-core graph algorithms to PageRank's Power iterations. Their approach resulted in significant performance gains as the ratio between the size of the data and the available RAM grew.

[3] The Open Directory Project, http://www.dmoz.org/

Kamvar et al. (2003b) note the locality of reference present in Web links: most links connect pages of the same host, with many of those connecting pages that are close to each other in terms of the directory structure of the host (the path part of the URL). Furthermore, when inter-host links exist, they are usually present between multiple pairs of pages on the two hosts. These observations allow for efficient software representations of the Web graph, reducing I/O and paging costs, and enabling parallelization of the PageRank computations with reduced communication overhead. The authors also report that when starting the Power method's iterations from an initial vector that is based on intra-host PageRank scores, the number of iterations required until convergence to the global PageRank scores is (empirically) halved. Overall, a speedup factor of 2–3 in PageRank computation is obtained in this work. See Section 3 for more on the differences between Kamvar et al. (2003b) and this paper.

Kamvar et al. (2003c) accelerate PageRank by adapting known algebraic methods for accelerating convergence of general linear sequences, to Power iterations over stochastic matrices. Speedup factors of up to 3 are reported. In Kamvar et al. (2003a), the authors observe that most entries in the PageRank vector converge quickly, with relatively few entries requiring many more power iterations to converge. This leads them to an algorithm that identifies (while iterating) the values that have converged and avoids recalculating them (and their contributions to scores of other pages) in subsequent iterations. Speedup factors of nearly 30% are achieved.

2.2. Alternatives to PageRank

So far we have discussed either accelerations of the basic PageRank algorithm itself, or alterations of the random jump behavior of the algorithm that produce different flavors of PageRank. In either case, the basic random walk model remained the same. This section discusses this model more critically, and surveys some alternatives (or major deviations) suggested in the literature.

While little is known about the exact use of PageRank by Google, it is widely believed that the simple model described in Section 2.1 does not give rise to the scores that are actually used by the engine. The practical implementation of PageRank, for example, might adopt a common practice in many link analysis algorithms, and follow internal links (links connecting pages within a site) with different probabilities than external links (links connecting pages of different sites). Such differentiation essentially changes the semantics of PageRank's browsing step. In particular, it was noted in Amitay et al. (2003) that PageRank may not be the most appropriate browsing model for many sites. One scenario discussed there is of a surfer, visiting the home page of a search engine. Realistically, the surfer is much more likely to submit a query and continue to browse the search results (essentially performing a random jump), than to follow the link from the engine's home page to its "about" page. As defined, PageRank will assign a significant fraction of the home page's score to the "about" page, while in practice, relatively few surfers visit that page. Another scenario considers surfers at the top level of some Web hierarchy, e.g. Yahoo! (`www.yahoo.com`). Such surfers will either enter a query in a search box (resulting in a random jump, as argued above) or will patiently browse the directory structure to one of the category pages. Today's hierarchies contain thousands of categories, and the layout of the sites is such that browsing paths to some categories are quite long. In PageRank, the probability of sustaining a long browsing sequence decreases exponentially, because of the random jumps that are performed probabilistically at each step. Thus, the contribution of the PageRank of the hierarchy's home page to a category page decreases exponentially in the category's depth. The category pages are essentially punished because of the fact that the directory is well organized.

Tsoi et al. (2003) study the problem of assigning static scores, close as possible to normal PageRank scores, that satisfy certain linear constraints. Such constraints can specify minimal score values for certain pages, enforce orderings between scores of pages, etc. They propose an algorithm based on quadratic programming, and then propose some dimensionality reduction schemes on the scale of the data so as to enable practical implementations of the algorithm. In Tomlin (2003), the author suggested a ranking model based on network flows through the Web graph. PageRank is actually a special case of this more general model, where the flow of probabilities through the network is conserved. He heuristically compared the quality of the rankings produced by PageRank with those produced by two other variants of this model, one of which seemed to produce rankings of equal or higher quality than those produced by PageRank. Abiteboul et al. (2003) propose an algorithm that is able to compute static scores in an online, incremental fashion while continually scanning the underlying graph (e.g. while crawling the Web). In their approach, called *OPIC*, the connectivity matrix of the graph need not be stored—it is sufficient to only consider the incident (or outgoing) edges of the current node being visited. Furthermore, no constraints are imposed on the scanning order of the graph—they merely require that each node will be visited infinitely often. Also presented is an adaptation of OPIC that is suitable for scoring changing graphs, which is of special interest on the ever-changing Web.

2.3. The host-graph of the web

Our algorithm, described in the next section, makes use of the fact that the number of hosts on the Web, and the number of links between pages of different hosts, are significantly lower than the corresponding numbers of pages and interconnecting links.

Ruhl et al. (2001) studied the host-graph induced by a large snapshot (604 M pages) of the Web from August 2000. The nodes of the hostgraph represented Web hosts, and a directed link between two hosts existed if any page from the source host linked to any page of the destination host. They report that the number of hosts in that snapshot was merely 10.37 million. Furthermore, the 5540 million links between Web pages induced less than 263 million links (below 5% of the original number) in the host-graph. Kamvar et al. (2003b) report that in a crawl containing 70 million pages with over 600 million interconnecting links, more than 93% of the links connected same-host pages.

Our experimental setup, presented in Section 4, exhibits similar ratios: we used a Web-graph containing over 1446 million pages with almost 6650 million links. The number of unique hosts in this graph was about 31 million (just over 2% of the number of pages), with 241 million host-to-host edges (3.6% of the number of page-to-page links).

3. Stationary distributions based on graph aggregation

Let T be a random walk on a graph with n nodes. T will denote both the random walk and the stochastic matrix that governs it. Let the n nodes be partitioned into m classes $H_1, \ldots, H_m$. From T and the m classes of nodes we develop an alternative random walk, T', whose stationary distribution can be calculated more efficiently. In T', a state transition departing from a node $x \in H_i$ consists of the following two-stage process:

- Move to some node $y \in H_i$ according to a distribution π_i. Note that π_i depends on the class, but not on the particular vertex x in the class.
- From y move on according to T.

Note that in general, this alternative walk is not mathematically equivalent to the original random walk T. Furthermore, for many choices of T, $H_1, \ldots, H_m$ and $\pi_1, \ldots, \pi_m$, the two random walks may result in very different stationary distributions.

For the purposes of this paper, we concentrate on the case where T denotes PageRank and the partitioning of Web pages is according to their host. Hence, a class H_i contains all the nodes (pages) on a given host, and only those nodes. The choice of partitioning Web pages by hosts is quite natural, as this partitioning reflects the ownership and themes of the content. Furthermore, intra-host linkage patterns are more regular and predictable than inter-host linkage patterns, due to the prevalent use of templates in Web sites (Bar-Yossef and Rajagopalan, 2002). Thus, representing intra-host random browsing by some host-specific distribution may enable our alternative random walk to not stray very far from PageRank.

In what follows, we show that the stationary distribution of T' can be derived from the principal eigenvector of a $m \times m$ stochastic matrix (recall that m is the number of classes), which is generally less expensive than computing the principal eigenvector of the $n \times n$ stochastic matrix T.

Throughout this section, the following notations are used:

$T = [t_{i,j}]$ = the original page-to-page probability transition matrix. We assume that T is irreducible and aperiodic, and so T is ergodic and has a well-defined stationary distribution, namely its principal eigenvector. Note that the standard PageRank transition matrix satisfies these conditions.
n = the number of pages; thus, T has size $n \times n$.
$H = \{H_1, \ldots, H_m\}$ is the set of all hosts represented in T.
$m = |H|$ = the number of hosts.
$h(p)$ = the host of page p. We use $h(p)$ to denote both the host name and the set of pages on it; thus, $p \in h(p)$.
π_h = a positive probability distribution on pages with support limited to the pages of host h, that is $\pi_h(p) = 0$ for all $p \notin h(p)$ while $\pi_h(p) > 0$ for all $p \in h(p)$.

As explained earlier, we replace the original random walk T, by a new random walk T'. Using the notations above, and assuming that T' is at page p, a transition (step) consists of two parts:

1. Jump to a page in $q \in h(p)$, chosen at random according to the distribution $\pi_{h(p)}$.
2. Perform a transition out of q according to the q'th row of T, that is according to the behavior of the original random walk T when leaving page q.

There are several plausible models for the distributions π_h:

a. π_h could be uniform over the pages of host h.
b. π_h could be proportional to $\deg(q)$ for $q \in h$ and 0 elsewhere, where $\deg(q)$ can be chosen to be the in-or the out-degree of q, based on all incident links, intra-host links or inter-host links.
c. π_h could be based on an intra-host PageRank calculation, e.g. in the spirit of Kamvar et al. (2003b).

Let $S = [s_{i,j}]$ and $\tilde{S} = [\tilde{s}_{i,j}]$ be the following $n \times n$ stochastic matrices:

$$s_{i,j} = \begin{cases} \pi_{h(j)}(j) & h(i) = h(j) \\ 0 & \text{otherwise} \end{cases}$$

$$\tilde{s}_{ij} = \begin{cases} |h(i)|^{-1} & h(i) = h(j) \\ 0 & \text{otherwise} \end{cases}$$

Both S and $\tilde{S}$ are block matrices over the set of all Web pages and their blocks correspond to same-host pages. In S, each row of the sub-matrix that corresponds to the pages of host h is equal to π_h. In $\tilde{S}$, each row of the submatrix that corresponds to the pages of host h describes a uniform distribution over the pages of h. Thus $\tilde{S}S = S$ regardless of the choice of π_h.

Clearly, the random walk T' described above is defined by the stochastic matrix ST: the intra-host move is given by S, and the T-based move follows. Furthermore, for any irreducible, aperiodic $n \times n$ stochastic matrix T, both ST and $ST\tilde{S}$ are also irreducible and aperiodic, and so both matrices have uniquely defined (positive, normalized) principal eigenvectors. In particular, there exists a unique positive and L_1-normalized principal eigenvector of ST, that corresponds to the stationary probability distribution of our modified random surfer model, T'.

Our goal can now be formally expressed—we aim to efficiently compute a distribution vector β that satisfies

$$\beta ST = \beta \,. \tag{1}$$

As argued above, $ST\tilde{S}$ is also an irreducible and aperiodic stochastic matrix. Our first step in calculating β is to compute a probability vector α such that $\alpha ST\tilde{S} = \alpha$.

We claim that $\beta \triangleq \alpha ST$ satisfies Eq. (1). Indeed,

$$\begin{aligned} \beta ST &= (\alpha ST)ST = (\alpha ST)(\tilde{S}S)T \\ &= (\alpha ST\tilde{S})ST = \alpha ST = \beta \end{aligned}$$

At first glance, it seems that so far we have not gained any efficiency: instead of calculating β directly, we have chosen to calculate β via α, which is also a principal eigenvector of an $n \times n$ stochastic matrix. In what follows, however, we show that α can be computed quickly.

The matrix $ST\tilde{S}$ satisfies the following equations:

$$h(p) = h(q) \Longrightarrow \forall r, \quad ST\tilde{S}(p, r) = ST\tilde{S}(q, r)$$

$$h(p) = h(q) \Longrightarrow \forall r, \quad ST\tilde{S}(r, p) = ST\tilde{S}(r, q)$$

Therefore, same-host sources/destinations are indistinguishable and hence the iterations required to find a can be carried out using $m \times m$ transition matrices and m-dimensional probability vectors, whose columns correspond to the distinct hosts. The transition probability from host h_1 to host h_2 is given by

$$\sum_{i \in h_1} \pi_{h_1}(i) \cdot \sum_{j \in h_2} t_{i,j}.$$

Let $\tilde{\alpha}$ denote the m-dimensional stationary distribution vector corresponding to the above transition probabilities. Intra-host symmetry considerations imply that (the n-dimensional) α can be derived from $\tilde{\alpha}$ by simply dividing the probability assigned to each host by the

number of hosted pages. Formally,

$$\alpha(p) = \tilde{\alpha}_{h(p)}/|h(p)|.$$

However, there is no need to explicitly compute α, since one can easily compute αS directly from $\tilde{\alpha}$:

$$(\alpha S)(p) = \pi_{h(p)}(p) \cdot \tilde{\alpha}_{h(p)}.$$

All that is left for obtaining β is to multiply αS by T.

Note that the algorithm calculates an m-dimensional probability vector $\tilde{\alpha}$, which implies an n-dimensional probability vector α in which all pages on a given host are assigned the same probability. However, pages of the same host are weighted differently in αS and certainly in $\beta = \alpha ST$. We have thus managed to calculate a distribution in page-granularity while performing eigenvector calculations in host-granularity.

To summarize, below is the algorithm to derive a PageRank approximation based on host aggregation, given the matrix T and the distributions $\pi_1, \ldots, \pi_m$ that correspond to the m hosts of H:

1. Define an $m \times m$ stochastic matrix $\tilde{T} = [\tilde{t}_{i,j}]$ as follows:

$$\tilde{t}_{H_i,H_j} = \sum_{p \in H_i} \pi_i(p) \cdot \sum_{q \in H_j} t_{p,q}. \tag{2}$$

2. Calculate the principal eigenvector of $\tilde{T}$: compute an m-dimensional distribution vector $\tilde{\alpha}$ satisfying $\tilde{\alpha}\tilde{T} = \tilde{\alpha}$.
3. Compute an n-dimensional probability distribution γ, where for each node p,

$$\gamma(p) = \tilde{\alpha}_{h(p)} \cdot \pi_{h(p)}(p)$$

4. The stationary distribution of T' is the vector $\beta \triangleq \gamma T$.

We now revisit in more detail the *BlockRank* algorithm proposed by Kamvar et al. (2003b), and highlight several differences between their work and ours. BlockRank is a method that accelerates PageRank computations by selecting an initial distribution vector v from which (empirically) fewer Power iterations are needed until convergence, as compared with Power iterations starting from the uniform distribution. The vector v of BlockRank is closely related to the vector γ produced by our algorithm when (1) T, the original random walk, represents PageRank and (2) each distribution πH_i is set to be the intra-host PageRank vector of H_i. The differences between v and the appropriate γ flavor concern the definition of teleportations in the random walk over the hosts: in BlockRank, teleportations between hosts are uniform over all hosts, whereas in our approach, it follows from Eq. (2) that teleportations land on each host in proportion to the number of pages on that host.

After computing v, BlockRank performs Power iterations from v until convergence to PageRank, or equivalently—repeatedly multiplies distribution vectors by the stochastic matrix T, until converging to T's principal eigenvector. We, on the other hand, are interested in the principal eigenvector of the matrix ST, which we essentially get with single multiplication of γ by T. While it is clear that our approach is speedier than BlockRank, the artifacts of both algorithms are different as we do not end up calculating PageRank.

3.1. Efficiency

How does the time needed by our algorithm compare to the usual computation of PageRank? The standard power-iteration computation of PageRank converges in a few dozen iterations. Each iteration requires one pass over the complete list of links for the entire Web graph. In contrast our algorithm needs only two passes over the entire set of links: the first when defining $\tilde{T}$ (step 1), and the second when transforming γ into β (step 4).[4] The power iterations part in our algorithm is linear in the number of links of the host-graph, which typically is much smaller (maybe by a factor of 20), than the number of links in the page-graph.

Note that the reduction in the number of links between the page-graph and the host-graph has implications beyond the simple number-of-operations accounting: modern search engines are required to compute link-based measures on connectivity data of billions of pages. This corresponds to tens of billions of links, an amount of data whose representation exceeds the RAM capabilities of most single-machine platforms. Moving to smaller graphs such as the host graph may enable the connectivity data to once again fit in the RAM of a single machine. In such a case our algorithm will have to perform only two (inevitably slow) passes over the full list of links, versus 25–40 passes for standard PageRank.

Furthermore, moving to smaller graphs that can be fully held in memory simplifies the development of software that analyzes and manipulates the transition probability matrix to achieve faster convergence.

4. Experiments

The section reports on experiments with a specific flavor of host-aggregated PageRank approximation. In this flavor, hereby called the "U-model" ("U" as in *uniform*), we set all intra-host distributions πH_i to be uniform. Thus, departing from page p involves the following two-step process:

1. Jump uniformly at random to a page $q \in h(p)$.
2. Perform a regular PageRank step from page q.

Note that in the terminology of the previous section, the matrix S that corresponds to the "U-model" is simply $\tilde{S}$.

Our experiments are based on a Web-graph containing over 1446 million pages with almost 6650 million links, from a crawl of the Web conducted by AltaVista in September 2003. The graph, which is stored and accessed using AltaVista's Connectivity Server (Bharat et al., 1998), was built on an Alpha server with 4 CPUs (each running at 667 MHz with its own internal floating point processor) and 32 gigabytes of memory. The number of unique hosts in this graph is about 31 million (just over 2% of the number of pages), with 241 million host-to-host edges (3.6% of the number of page-to-page links).

Computing PageRank in this setting required 12.5 h., while computing the U-model scores took 5.8 h. (a speedup factor of about 2.1). It should be noted that the PageRank computations use the robust and optimized infrastructure of the connectivity server, while our modified algorithm was written in an ad-hoc, non-optimized manner. We predict that by optimizing our implementation in the spirit of the connectivity server, speedup factors can approach the full potential indicated by the ratio of the number of links in the Web graph and

[4] This analysis assumes that the set of distributions π_h are given, e.g. as by-products of the build process of the connectivity database, which is certainly the case for the experiments presented in the next section.

Table 1 Sampling schedule by PageRank decile

Starting rank	Ending rank	Sampling probability
1	1000	0.2
1001	10000	0.02
10001	100000	0.002
100001	1000000	0.0002
1000001	10000000	0.00002
10000001	100000000	0.000002
100000001	1000000000	0.0000002
1000000001	14470000000	0.00000002

the corresponding figure in the host-graph. In particular, it follows from the discussion in Section 3 that optimized implementations can achieve considerable speedup factors relative to BlockRank (Kamvar et al., 2003b), which itself empirically speeds up PageRank by factors of 2–3.

In what follows we provide statistical comparisons between the U-model flavor and PageRank. We show that U-model approximates PageRank very closely, especially if one considers the correlation between the ranks induced by these measures.

To assess the relation between PageRank and its approximation, the U-model, we sampled from the available 1446 million pages. However, we did not take a simple random sample since, by virtue of the power-law distribution of PageRank, the bulk of the population are pages with few or no inlinks. Such pages would attain similar, low scores by practically any link-based static score measure. Furthermore, as a consequence of their low scores, these pages will rarely appear as the top ranking results for queries, and so small fluctuations in their static scores will hardly be noticed by search engine users.

Therefore, instead of sampling uniformly, we used PageRank to stratify the population and uniformly sampled from each decile. This produced a sample containing many more representative pages with larger values of PageRank than would have been possible with a similar sized simple random sample. In particular, we sorted pages according to PageRank and then uniformly sampled within each decile according to the schedule in Table 1. The resulting sample contained 1298 pages distributed over the full range of PageRank values. For each page in the sample we had available both the PageRank and U-model values.

In this sample U-model has Pearson correlation 0.81 with PageRank. Given the amply large sample size and the high correlation, a standard statistical test rejects with high confidence the hypothesis that U-model is statistically independent of PageRank. This agrees with expectation since U-model is designed to approximate PageRank.

To examine the relation more closely, we fit a linear model via least-squares with U-model as the single predictor and PageRank as the response. If U-model is indeed a good approximation to PageRank we expect the linear fit to be very good, with slope close to 1.0, and with well behaved residuals. See Fig. 1 for a scatter plot display of this fit. The linear regression line, with error bars, is indicated on the plot. Note that the slope of the fit, 0.9851, is close to 1.0 as hoped. However several data point are far from the regression line.

See Fig. 2 for a closer look at the discrepancy between PageRank and its U-model approximation. Most of standardized residuals from the fit are very small. Indeed, there are more residuals close to zero than expected if the residuals are compared to a standard Gaussian distribution. However, the outlying points (with large residuals) tend to correspond to pages with large PageRank values, suggesting that U-Model could be a less precise approximation of PageRank in the very high end of the scale.

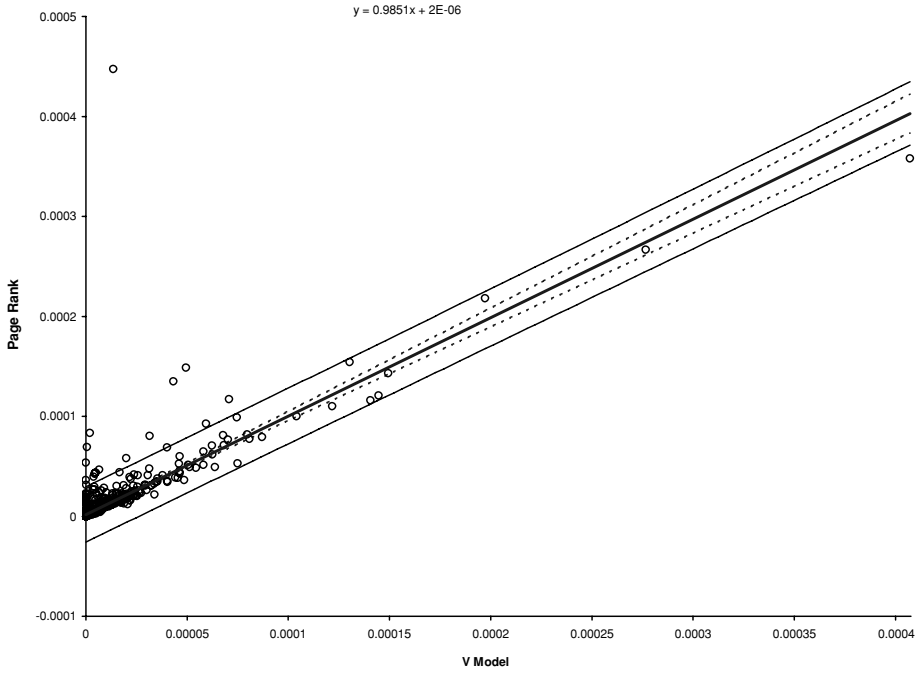

Fig. 1 Linear fit of PageRank vs U-model

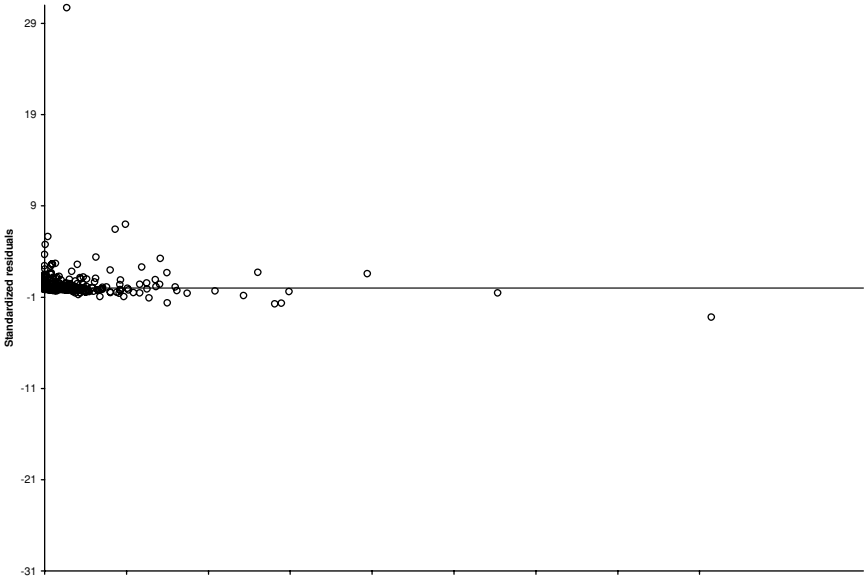

Fig. 2 Standardized residuals

The best assessment of the U-model approximation would be task oriented; does U-model offer as good information as PageRank for relevance ranking of Web pages? However, this requires considerable machinery outside the scope of this paper. A reasonable surrogate is to ask if the rank order of pages given by U-model resembles the order given by PageRank. In particular, if the orderings were identical the two measures would offer the same information for relevance ranking even if their values deviated considerably. In fact, the Spearman rank-order correlation (Snedecor and Cochran, 1989) between U-model and PageRank is 0.95, considerably higher than the Pearson correlation of 0.81. This suggests that U-model can be used as an effective approximation for PageRank in relevance ranking.

In terms of use for ranking of Web search results, we do not know whether the (small) differences between our model and the original PageRank are for better or for worse. One possible advantage of our model is that it is almost insensitive to changes in the internal organization of hosts. A reorganization that modifies the linkage patterns between the pages on a given host might change the relative ranking of these pages, but will have much less effect on pages outside the host, since the "weight" of the host is uniformly distributed over all its pages.

In this context we note that Web IR community is somewhat divided on the topic of whether static link-based ranking measures like PageRank are at all useful for IR tasks. Papers citing improvements abound, as well as papers showing very little or no gain from using static score metrics. At the heart of the debate is the lack of an agreed upon retrieval benchmark for Web IR:

- A corpus that constitutes a large and representative sample of the Web (including "optimized" pages, spam, and the likes).
- A large set of queries that is representative of user needs.
- A clear methodology for assessing quality of search results over the scale of data and queries.
- A clear baseline with a state-of-the-art text scoring component (including anchor-text), over which retrieval improvements will be sought.

Furthermore, even though it is unlikely that commercial search companies would publish their own results on such a benchmark, it would be valuable to have them take part in defining the benchmark. In the meantime, we hope that even researchers who are skeptical of PageRank's contribution to Web IR would appreciate the mathemathical ideas and random walk approximation framework presented in this paper.

5. Conclusions

This paper introduced a new framework for calculating random-walk based static ranks of pages on the Web. The framework approximates the behavior of a given random walk on the Web's graph in a scalable manner, by performing most of its calculations on a more compact representation of the graph—a representation that follows from aggregating multiple pages onto a single node. In particular, this compaction significantly reduces the memory requirements of the computations that arise when dealing with Web graphs of large crawls.

As defined in Section 3, our method changes the semantics of random browsing on the Web by decoupling intra-host and inter-host steps. One should note, however, that the method can be defined in terms of any equivalence relation that partitions the set of Web pages. An immediate example is to relax the physical host-based partitioning to one driven by logical sites: sometimes, pages on separate hosts might correspond to the same logical entity

(e.g., different academic units of some university); in other cases, pages on the same host might belong to independent entities—e.g., sites virtually hosted by services such as *Geocities* (`http://www.geocities.com`).

We approximated PageRank with a random walk flavor in which departures from a page consist of a two stage process: first, a random transition to another page of the same host, and then a PageRank-like step from that page. Whereas PageRank requires repeated scans of the Web-graph's links, most of the computations required by our transformation involve scanning the edges of the Web's host-graph. We achieved a speedup factor of 2.1, and predict that more careful implementations can achieve speedups that are closer to the ratio of the sizes of the graph. Furthermore, we showed that the resulting scores indeed approximate PageRank: we were able to produce a ranking that has Spearman rank-order correlation of 0.95 with respect to PageRank.

It should be noted that the performance gains achieved by this approach can be further improved by many of the schemes surveyed in Section 2.1.3. For example, the ideas presented in Haveliwala (1999) and Kamvar et al. (2003c) are applicable to this proposal as well, as the problem remains computing the principal eigenvector of a large and sparse stochastic matrix.

Future efforts should be devoted to experimenting with different aggregates of the Web's graph, and with different stochastic behaviors within those aggregates. Each such flavor should be evaluated in terms of the speedup factors it achieves, and in terms of the search quality that it induces when used in the ranking core of search engines.

References

Abiteboul S, Preda M and Cobena G (2003) Adaptive on-line page importance computation. In: Proc. 12th International WWW Conference (WWW2003), Budapest, Hungary, pp. 280–290

Amitay E, Carmel D, Darlow A, Lempel R and Soffer A (2003) The connectivity sonar: Detecting site functionality by structural patterns. In: Proc. of the ACM Hypertext 2003 Conference, Nottingham, UK. pp. 38–47

Bar-Yossef Z and Rajagopalan S (2002) Template detection via data mining and its applications. In: Proceedings of the 11th International WWW Conference, Honolulu, Hawaii, USA, pp. 580–591.

Barabasi A-L and Albert R (1999) Emergence of scaling in random networks. Science, 286:509–512

Bharat K, Broder A, Henzinger M, Kumar P and Venkatasubramanian S (1998) The connectivity server: Fast access to linkage information on the web. In: 7th International World Wide Web Conference, pp. 104–111

Brin S and Page L (1998) The anatomy of a large-scale hypertextual web search engine. In: Proc. 7th International WWW Conference, pp. 107–117

Broder A (2002) A taxonomy of web search. SIGIR Forum, 36(2):3–10.

Broder A, Kumar R, Maghoul F, Raghavan P, Rajagopalan S, Stata R, Tomkins A and Wiener J (2000) Graph structure in the web. In: Proc. 9th International WWW Conference. pp. 309–320

Chen Y-Y, Gan Q and Suel T (2002) I/O efficient techniques for computing pagerank. In: Proc. ACM Conference on Information and Knowledge Management (CIKM2002)

Chien S, Dwork C, Kumar R, Simon D and Sivakumar D (2002) Link evolution: Analysis and algorithms. In: Workshop on Algorithms and Models for the Web Graph (WAW), Vancouver, Canada.

Cho J, GarcíAa-Molina H and Page L (1998) Efficient crawling through URL ordering. Computer Networks and ISDN Systems 30(1–7):161–172

Gallager RG (1996) Discrete stochastic processes. Kluwer Academic Publishers

Haveliwala TH (1999) Efficient Computation of PageRank. Technical Report Technical Report, Stanford University

Haveliwala TH (2002) Topic-Sensitive PageRank. In: Proc. 11th International WWW Conference (WWW2002).

Haveliwala TH, Kamvar SD and Jeh G (2003) An Analytical Comparison of Approaches to Personalizing PageRank. Technical Report Technical Report, Stanford University

Henzinger MR, Heydon A, Mitzenmacher M and Najork M (1999) Measuring index quality using random walks on the Web. Computer Networks (Amsterdam, Netherlands 1999) 31(11–16):1291–1303.

Jeh G and Widom J (2003) Scaling personalized web search. In: Proc. 12th International WWW Conference (WWW2003), Budapest, Hungary., pp. 271–279

Jennings A (1977) Matrix computation for engineers and scientists. John Wiley & Sons, Ltd.

Kamvar SD, Haveliwala TH and Golub GH (2003a) Adaptive methods for the computation of PageRank. Technical report, Stanford University.

Kamvar SD, Haveliwala TH, Manning CD and Golub GH (2003b) Exploiting the block structure of the web for computating PageRank. Technical Report Technical Report, Stanford University

Kamvar SD, Haveliwala TH, Manning CD and Golub GH (2003c) Extrapolation methods for accelerating PageRank computations. In: Proc. 12th International WWW Conference (WWW2003), Budapest, Hungary, pp. 261–270

Kleinberg JM, Kumar R, Raghavan P, Rajagopalan S and Tomkins AS (1999) The web as a graph: Measurements, models and methods. In: Proc. of the Fifth International Computing and Combinatorics Conference, pp. 1–17

Lee HC (2002) When the hyperlinked environment is perturbed. In: Workshop on Algorithms and Models for the Web Graph (WAW), Vancouver, Canada

Lempel R and Moran S (2001) Rank-stability and rank-similarity of web link-based ranking algorithms. Technical Report CS-2001-22 (revised version), Dept. of Computer Science, Technion-Israel Institute of Technology

Lifantsev M (2000) Voting models for ranking web pages. In: Proc. International Conference on Internet Computing (IC 2000), Las Vegas, Nevada, pp. 143–148

Ng AY, Zheng AX and Jordan MI (2001) Stable algorithms for link analysis. In: Proc. 24'th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 258–266

Pandurangan G, Raghavan P and Upfal E (2002) Using PageRank to characterize web structure. In: Proc. 8th Annual International Computing and Combinatorics Conference, pp. 330–339

Richardson M and Domingos P (2001) The intelligent surfer: Probabilistic combination of link and content information in PageRank. In: Advances in Neural Information Processing Systems 14 [NIPS 2001], Vancouver, British Columbia, Canada, pp. 1441–1448, MIT Press

Ruhl M, Bharat K, Chang B-W, Henzinger M (2001) Who links to whom: Mining linkage between web sites. In: IEEE International Conference on Data Mining (ICDM), pp. 51–58

Snedecor GW and Cochran WG (1989) Statistical methods. Iowa State University Press, 8th edition

Tomlin J (2003) A new paradigm for ranking pages on the World Wide Web. In: Proc. 12th International WWW Conference (WWW2003), Budapest, Hungary, pp. 350–355

Tsoi AC, Morini G, Scarselli F, Hagenbuchner M and Maggini M (2003) Adaptive ranking of web pages. In: Proc. 12th International WWW Conference (WWW2003), Budapest, Hungary, pp. 356–365

Upstill T, Craswell N and Hawking D (2003) Predicting fame and fortune: PageRank or indegree?. In: Proc. 8th Australasian Document Computing Symposium, Canberra, Australia