Internet Mathematics Vol. 2, No. 1: 73-120

A Survey on PageRank Computing

Pavel Berkhin

Abstract. This survey reviews the research related to PageRank computing. Components of a PageRank vector serve as authority weights for web pages independent of their textual content, solely based on the hyperlink structure of the web. PageRank is typically used as a web search ranking component. This defines the importance of the model and the data structures that underly PageRank processing. Computing even a single PageRank is a difficult computational task. Computing many PageRanks is a much more complex challenge. Recently, significant effort has been invested in building sets of personalized PageRank vectors. PageRank is also used in many diverse applications other than ranking.

We are interested in the theoretical foundations of the PageRank formulation, in the acceleration of PageRank computing, in the effects of particular aspects of web graph structure on the optimal organization of computations, and in PageRank stability. We also review alternative models that lead to authority indices similar to PageRank and the role of such indices in applications other than web search. We also discuss link-based search personalization and outline some aspects of PageRank infrastructure from associated measures of convergence to link preprocessing.

1. Introduction

The tremendous popularity of Internet search engines presents a striking example of a successful fusion of many developments in different areas of computer science and related fields from statistics to information retrieval to natural language processing. One of the most difficult problems in web search is the ranking of the results recalled in response to a user query. Since contemporary web crawls discover billions of pages, broad topic queries may result in recalling hundreds of thousand of pages containing the query terms. Only a few dozens of the recalled pages are actually presented to the user. Moreover, these results are presented in order of relevance. A variety of ranking features are used by Internet search engines to come up with a good order. Some of the query-independent features are based on page content. Some utilize the hyperlink structure of the web.

The ascent of Google to prominence is to a certain degree associated with the concept of a new relevance feature of the second kind, PageRank. PageRank assigns authority weights to each page of a web graph based on the web hyperlink structure. It is based on an assumption that authoritative pages usually contain links to good pages, therefore endorsing them in a degree inversely proportional to the number of their out-links. This fundamental assumption is not always true in practice partly because good pages may erroneously link to bad ones, partly because they have a natural tendency to emphasize pages within their own sites, and partly because of the consistent efforts of malicious sites containing spam. While this affects its usefulness, PageRank remains an important relevance feature when combined with other ranking components. Many issues are related to PageRank computing: from filtering, weighting, and compressing the link data to the most advantageous modification of the underlying model to finding fast methods of computing its solution.

PageRank is based on a random surfer model [Page et al. 98] and may be viewed as a stationary distribution of a Markov chain. The PageRank solution is a principal eigenvector of a linear system that can be found via the power method. We discuss this model and some technical modifications (teleportation mechanism) needed to safeguard the existence of a solution and the convergence of an iterative process in Section 2.

The sheer size of the World Wide Web presents a challenge to PageRank computing. Different developments to speed-up the power method include extrapolation, adaptive, and block structure methods. PageRank can also be reformulated in terms of a linear system. Linear systems have a rich arsenal of fast iterative methods. The effective solution of PageRank depends on a special enumeration of its nodes corresponding to a DAG structure. We discuss effective procedures for PageRank computing and its stability with respect to changes in the web graph and in the teleportation coefficient in Section 3.

Simultaneously with the random surfer model, a different but closely related approach, the HITS algorithm [Kleinberg 99], was invented. This algorithm results in two authority vectors. PageRank results in one. More importantly, HITS is designed to work at query time, while PageRank is precomputed in advance of its utilization in a search. HITS, its modifications, and other later models leading to a concept of authority weights defined on a directed graph such as, for example, SALSA, as well as the use of such weights in applications other than web search relevance ranking are surveyed in Section 4.

The PageRank vector serves as only one of the many relevance features for ranking web search results. Generic PageRank corresponds to a uniform teleportation. In principle, PageRanks with different teleportations can be used. A nonuniform teleportation makes sense, since it leads to a topical or user-specific personalized PageRank [Page et al. 98, Haveliwala 02b]. Computing many such PageRanks is difficult, but in some cases it may be significantly optimized. Recent results in personalized PageRank are presented in Section 5.

Remarkable efforts have been made to facilitate the PageRank computing infrastructure. They range from special means to estimate PageRank convergence, to web graph analysis, to special data compression techniques. We present them in Section 6.

2. Definitions and Fundamentals

In this section we formally define PageRank and consider the theoretical foundations for its existence and for the convergence of the power iteration.

2.1. Definitions

The PageRank vector (*PageRank citation ranking weight*) was introduced in [Page et al. 98] and [Brin and Page 98]. It assigns to pages authority scores that are higher for pages with *many in-links* from *authoritative* pages with relatively *few out-links*. Let $W(G, L)$ be a directed graph with vertices (or nodes) G that are web HTML pages and directed edges L that are hyperlinks. The edges L can be associated with a $n \times n$ sparse *adjacency* matrix that we denote by the same symbol L, where $n = |G|$ is the number of web pages. An element L_{ij} is equal to 1 if a page i has a link $i \to j$ to a page j and is equal to 0 otherwise. The out-degree of a node i is the number of outgoing links, $deg(i) = \sum_j L_{ij}$. All the definitions can be generalized to a weighted directed graph, in which case $L_{ij} \geq 0$. A transition matrix is defined as $P_{ij} = L_{ij}/deg(i)$ when $deg(i) > 0$. For such i it is *row-stochastic*, meaning that i-row elements sum to one. We set $P_{ij} = 0$ when $deg(i) = 0$.

Consider the following *random surfer model*. A surfer travels along the directed graph. If at a step k he is located at a page i, then at the next step $k+1$ he moves uniformly at random to any out-neighbor j of i. Obviously, pages can be considered as n states of a Markov chain with a transition matrix P. Given a distribution $p^{(k)} = \left(p_i^{(k)}\right)$ of probabilities for a surfer to be at a page i at a step

k, the probability for being at a page j on the next step is proportional to

$$p_j^{(k+1)} = \sum\nolimits_{i \to j} p_i^{(k)}/deg(i) = \sum\nolimits_i P_{ij} p_i^{(k)} \quad or \quad p^{(k+1)} = P^T p^{(k)}. \tag{2.1}$$

(If all pages have out-links, $p_j^{(k+1)}$ is *indeed* a probability distribution.) An equilibrium state of transformation (2.1) correctly reflects our modeling objectives. The more incoming links a page has, the higher its importance. The importance of a page j is also proportional to the importance of its in-neighbors and inversely proportional to the neighbor's out-degrees. This motivates the following definition.

Definition 2.1. A PageRank vector is a stationary point of the transformation (2.1) with nonnegative components (a stationary distribution for a Markov chain)

$$p = Ap, \qquad A = P^T. \tag{2.2}$$

Since we are interested in a probability distribution, the sum of the p-components is assumed to be one. We use an L_1 norm $\|x\| = \sum |x_i|$ as our standard norm.

This definition has a problem. Matrix P has zero rows for *dangling* pages (pages that have no outgoing links), since for them $deg(i) = 0$, while a Markov chain transition matrix has to be row-stochastic. Dangling pages serve as *sinks* or *attractors*: some p-volume comes to them but then cannot leave. Different remedies have been explored. One approach [Page et al. 98, Section 2.7] suggests getting rid of dangling pages. The rationale is that these pages do not influence any other pages. Notice that deleting dangling pages generates new dangling pages. Another alternative considered in that paper is to renormalize $P^T p^{(k+1)}$ so that it remains a probability distribution to compensate for sinks. The authors of [Kamvar et al. 03b] consider adding deleted dangling pages on final iterations. Still another option advocated by [Jeh and Widom 02b] is to add a self-link to each dangling page and, after computation, to adjust the constructed PageRank. Another approach [Bianchini et al. 03] suggests introducing an ideal page (sink) with a self-link to which each dangling page links. Along a similar line, see [Eiron et al. 04]. The most widely used device [Page et al. 98, Haveliwala et al. 03b, Kamvar et al. 03c, Langville and Meyer 05] modifies the matrix P by adding artificial links that uniformly connect dangling pages to all G pages. Let v be a distribution, for example, $v = e, e = e_n = (1/n, \ldots, 1/n)$, and let d be a dangling page indicator, $d_i = \delta(deg(i), 0)$. Here, as usual, $\delta(a, b) = 1$ when $a = b$ and is zero otherwise. Consider the matrix

$$P' = P + d \cdot v^T. \tag{2.3}$$

Rows corresponding to dangling pages are changed to v. The trick is popular since, as we will see shortly, it is also useful in a general situation of nondangling pages and is easy to compute. Aside from a particular approach, the decisive fact is that there are many dangling pages to deal with. (In reality, on a Yahoo! web graph from data collected in August, 2004, and consisting of billions of pages, more than 60% of pages have not been crawled, and therefore, these pages are dangling for sure. Around 14% of actually crawled pages are also dangling. In a corresponding host graph 62% of the pages are dangling pages.)

When does the stationary point in (2.2) exist and when does the process in (2.1), called the *power method* or *power iteration*, converge to the solution? For a nonnegative row-stochastic matrix P independent of the initial distribution $p^{(0)}$ (elements are nonnegative and sum to one), the power method converges to a principal eigenvector p of A, p has positive elements, and the principal eigenvalue $\lambda_1 = 1$ is simple as long as two conditions are satisfied:

1. The graph $W(G, L)$ is *strongly connected*, meaning that there is a directed path from each node to any other node.

2. The graph $W(G, L)$ is *aperiodic*, meaning that for any pages i and j there are paths from i to j (with whatever repetitions) of any length l except for a finite set of lengths.

While Condition 2 is guaranteed for the web, Condition 1 is routinely violated. For example, there are many pages without in-links. To satisfy both conditions, the trick used for dangling pages is repeated: the matrix is modified to

$$P'' = cP' + (1-c)E, \quad E = (1, \dots, 1) \cdot v^T, \quad 0 < c < 1. \tag{2.4}$$

Here all rows of E coincide with v. The transformation (2.4) conserves the stochastic property (it transforms a probability distribution into another probability distribution). According to (2.4), from a nondangling page a surfer follows one of the local out-links with probability c and jumps to some $j \in G$ with probability $(1-c)v_j$. For this reason, v is known as a *teleportation* vector. A uniform $v = e$ is usually employed, resulting in a uniform PageRank. However, in a personalization context nonuniform jumps are used. Consequently, v is also known as a *personalization* vector. Different recipes for choosing c range from 0.85 [Page et al. 98, Kamvar et al. 03b] to 0.99 [Haveliwala et al. 03b, Kamvar et al. 03c]. Values 0.85–0.9 are used in practice.

The original P is sparse. We still can benefit from this fortunate fact after the transformations (2.3) and (2.4), since the multiplication with matrix P'' can be carried out [Kamvar et al. 03c] in terms of the original P: the vector

$y = Ax = P''^T x$ can be computed by the following code:

$$\begin{aligned} y &= cP^T x, \\ \gamma &= \|x\| - \|y\|, \\ y &= y + \gamma v. \end{aligned} \tag{2.5}$$

General information related to PageRank can be found in a survey article [Langville and Meyer 04a]. It contains some explanatory examples corresponding to small graphs. Henzinger [Henzinger 00] also reviews related issues.

2.2. Related Theory

The *Perron-Frobenius* theorem for positive matrices is proved in [Axelsson 94, Chapter 4]. Under Condition 1 of strong connectivity, it guarantees that λ_1 is simple and that the corresponding eigenvector has positive components. The fact that the eigenvalue $\lambda_1 = 1$ is a consequence of stochasticity. Condition 2 of aperiodicity relates not to the existence of a solution but to the convergence of (2.1), which is not guaranteed in its absence. For example, for a strongly connected two-node graph $a \leftrightarrow b$ that does not satisfy the aperiodicity condition (since all paths from a to b have odd lengths), the iterations (2.1) for $p^{(0)} = (q, 1-q)$ oscillate without convergence between $p^{(0)}$ and $p^{(1)} = (1-q, q)$.

A matrix is *reducible* if after a permutation of coordinates into two nonempty groups, first of size d and second of size $n-d$, it has a block triangular form

$$P = \begin{pmatrix} P_{11} & P_{12} \\ 0 & P_{22} \end{pmatrix}, \tag{2.6}$$

where $dim(P_{11}) = d \times d$, $dim(P_{22}) = (n-d) \times (n-d)$. It is called *irreducible* otherwise [Axelsson 94, Definition 4.2, Theorem 4.2]. An adjacency matrix and a transition matrix of a directed graph are irreducible if and only if the associated graph is strongly connected [Axelsson 94, Theorem 4.3]. If the graph W is not strongly connected, then factorization of its strongly connected components (SCC) leads to a directed acyclic graph (DAG), discussed in Section 3.4. Every DAG's *final* node (i.e., having no out-links) corresponds to a SCC that serves as an attractor in a corresponding Markov chain. In plain language all the p-volumes gradually get sucked into such states. Condition 1 of strong connectivity is *necessary* for the existence of a unique positive solution of (2.2). If P is reducible, then the system (2.2) either has multiple positive solutions, if $P_{12} = 0$ (combinations of solutions for two blocks), or has no positive solution (since $p_1 = P_{11}^T p_1$ and P_{11} is not stochastic if $P_{12} \neq 0$).

So far, we only know λ_1. A less recognized fact is that eigenvalues of P' and P'' are interrelated: $\lambda_1(P'') = \lambda_1(P') = 1, \lambda_j(P'') = c\lambda_j(P')$ for $j > 1$ [Langville and Meyer 05]. In particular, the second eigenvalue $\lambda_2(P'') \leq c$. In reality, for a web graph $\lambda_2(P'') = c$. This amazing fact, first proven by [Haveliwala and Kamvar 03], is important for extrapolation. It has the astonishing consequence that a teleportation mechanism introduced to secure strong connectivity also determines the convergence rate. The authors also prove that if

$$A = (cP' + (1-c)E)^T, \quad E = (1, \dots, 1) \cdot v^T, \tag{2.7}$$

where P' is row stochastic and has at least two irreducible closed subsets, then A has a second eigenvalue $\lambda_2 = c$. The web graph $W(G, E)$ has many strongly connected components [Kumar et al. 00]. A proof can also be found in [Langville and Meyer 05].

Further information about random walks on directed graphs can be found in [Motwani and Raghavan 95, Ahuja et al. 93, Lovász 96]. Not all of the theory is relevant to PageRank. The Markov model defined by a finite graph with a moderate number of outgoing links has its own intrinsic features. It has been observed [Page et al. 98] that the web graph is expanding, meaning that the number of vertices reachable from a (not too large) subset exceeds by a factor α the cardinality of the initial set. A graph has a high expansion factor if and only if its principal eigenvalue is sufficiently larger than the second eigenvalue ("eigenvalue gap"). A random walk on a graph is *rapidly-mixing* if iterations converge in $\log(|G|)$ time, which in turn depends on the principal eigenvalue separation.

As we explained, pages G can be viewed as Markov chain nodes with a transition matrix P. The theory of Markov chains includes the important concept of ergodic states. Let $N(i, k)$ be the number of times a surfer visited page i during the first k steps. The solution of Equation (2.2) is related to N. When strong connectivity and aperiodicity are in place, the Ergodic theorem [Motwani and Raghavan 95, Theorem 6.2] says that each state is ergodic:

$$\lim_{k \to \infty} N(i, k)/k = p_i.$$

The transition matrix P is related to an adjacency matrix L by the formula $P = D^{-1} \cdot L$, where a diagonal matrix $D = \{deg(i)\}$ contains out-degrees. Generalizations of this formulation are discussed in [Ding et al. 01]. Brandes and Cornelson [Brandes and Cornelsen 03] present a general view of other similar constructions for directed graphs; independently of PageRank that induces an order over G, they also describe a method of *spectral graph layout* resulting in another ordering of *undirected* graphs. The method relates to the important concepts of the *Laplacian matrix* and the minimum energy state *Fiedler vector*.

PageRank can be formulated in a form alternative to a spectral problem. Since $(1,..,1)^T p = \|p\| = 1$, the PageRank vector, in addition to an eigensystem $p = Ap$, $A = P''^T$, also satisfies the equation

$$p = cP'^T p + (1-c)v = cP^T p + (1-c+cs)v, \tag{2.8}$$

where $s = s(p) = d^T \cdot p$ is a "dangling sink." The actual value of s affects the solution only through rescaling it. Therefore, any s, for example, $s = 0$ (that results in the spectral eigensystem solution in the absence of dangling pages), can be used. This fact has important implications. We elaborate on this topic in Section 3.5. Finally, modifying P' with teleportation can be avoided [Langville and Meyer 05, Tomlin 03] if in (2.7) G is augmented with an additional ideal sink node $n+1$:

$$\tilde{p} = \tilde{P}\tilde{p}, \tilde{P} = \begin{pmatrix} cP' & (1-c)(1,\ldots,1) \\ v^T & 0 \end{pmatrix}, \quad \tilde{p} = \begin{pmatrix} p \\ (1-c) \end{pmatrix}.$$

This construction is also useful in handling dangling pages [Eiron et al. 04, Lee et al. 03].

2.3. Usage

The traditional application of PageRank is ranking web search results. The appealing quality of PageRank is that it is available in query time. Since PageRank only uses web graph connectivity and is query-independent, it has to be blended with other features, including query-dependent features. Along with PageRank, another similar algorithm reviewed below and called HITS [Kleinberg 99] works in query time. The benefits of PageRank are best for under-specified (*broad* topic) queries, for example "University" [Page et al. 98]. In other words, it is best for queries with very high recall based on a straightforward all-term inclusion procedure. From an application standpoint, PageRank has some problems. For example, an artificially high rank is assigned to copyright warnings, disclaimers, and highly interlinked mailing archives. Another problem is a potential topic drift. Different remedies were suggested, starting with an insightful article [Bharat and Henzinger 98] (which we get back to in the context of HITS). The major issue is how to merge connectivity and content data. Internet search engines blend PageRank with other relevance features using proprietary ranking formulas and other techniques, and any statistics reflecting its significance in this regard are not public. There is no certainty about the importance of personalized PageRank either, except for the fact that no other obvious link-based personalization feature exists. HostRank, PageRank constructed over the host graph, is another useful relevance feature.

Algorithm 1. (Poor man's PageRank algorithm.)

```
Input: Given a transition matrix $P$, a teleportation vector $v$, and a
       coefficient $c$
Output: Compute PageRank $p$
begin
    Initialize $p^{(0)} = v$, $k = 0$
    repeat
        $p^{(k+1)} = cP^T p^{(k)}$
        $\gamma = \|p^{(k)}\| - \|p^{(k+1)}\|$
        $p^{(k+1)} = p^{(k+1)} + \gamma v$
        $\delta = \|p^{(k+1)} - p^{(k)}\|$
        $k = k + 1$
    until $\delta < \epsilon$
end
return $p^{(k)}$
```

Before moving further we would like to present the reader with the "Poor man's PageRank algorithm" implementation of the power iteration that takes (2.5) into account; see Algorithm 1.

3. Acceleration of Convergence

As we explained, our goal is to find a stationary point p for an eigensystem

$$p = Ap, \qquad A = P''^T. \tag{3.1}$$

We start with some distribution $p^{(0)}$ (e.g., $p^{(0)} = e$) and use the power iteration

$$p^{(k+1)} = Ap^{(k)} \tag{3.2}$$

to compute the eigenvector associated with the principal eigenvalue $\lambda_1 = 1$. According to (2.5), A-multiplication successfully exploits the sparse nature of the transition matrix P (on average, there are between 7 and 20 out-links per page depending on the pruning used) and thus, a major iteration has $O(n)$ computational complexity. This is good news; however, the graph size n is very large. We use a simple L_1 stopping criterion for transparency, but we will return to this issue in Section 6.1.

Before we investigate more complex algorithms, what is the convergence rate of the power iteration? We already know that $\lambda_1(P) = 1, \lambda_2(P) = c$. It is easy to see that a convergence rate of the power iteration process for *any* P

is $\lambda_2(P)/\lambda_1(P)$ or c in our case. We would like to provide somewhat simpler reasoning specifically related to PageRank that leads to a slightly weaker result. Following [Bianchini et al. 03], if we subtract from Equation (2.8), $p = cP'^T p + (1-c)v$, its iterative version (3.2), we get

$$p - p^{(k+1)} = cP'^T (p - p^{(k)}) = c^k \left(P'^T\right)^k (p - p^{(0)}),$$

and hence, since $\|P'^T\| \le 1$, the rate of convergence is bounded by c. The number of iterations $k \approx \log(\epsilon)/\log(c)$ needed to achieve ϵ-accuracy is independent of the graph.

Next we review different algorithmic methods to accelerate the power method (3.2) and other methods of computing PageRank. In Section 6 we talk about infrastructural means to achieve computational efficiently. Since a web graph is volatile, there is a natural limit on how accurate we would like to be. Also, the whole adventure with computing PageRank makes sense only if the ideal object is stable. This is also one of our concerns in this section.

3.1. Extrapolation Methods

Extrapolation methods are based on the expansion of iterates $p^{(k)}$ in a series

$$p^{(k)} = u_1 + \lambda_2^k u_2 + \cdots + \lambda_m^k u_m + \cdots, \tag{3.3}$$

where terms u_m are (not orthonormal) eigenvectors of A (we utilize that $\lambda_1 = 1$). These methods try to improve approximation of the ideal solution u_1 by $p^{(k)}$ through clever suppression of the higher terms. More specifically, while computing power iterations, from time to time, we construct a linear combination of several iterates that exactly eliminates one or more harmonics after the very first term and regard the remaining tail as approximately relatively small. In doing so, we assume that $1 < \lambda_2 \le \lambda_3 \le \ldots \le \lambda_m \ldots$ and carefully analyze coefficients of the first m terms in series (3.3). The constructed combination is used as a closer starting point for further iterations.

Kamvar, Schlosser, and Garcia-Molina [Kamvar et al. 03c] present a variety of extrapolation methods. The simplest Aitken method aims at eliminating the second sub-principal term. Setting $m = 2$ and truncating the tail (denoted by "$\ldots$"), we get

$$\begin{aligned} p^{(k-2)} &= u_1 + u_2 \quad + \ldots \\ p^{(k-1)} &= u_1 + \lambda_2 u_2 + \ldots \\ p^{(k)} &= u_1 + \lambda_2^2 u_2 + \ldots . \end{aligned} \tag{3.4}$$

These are three equations with three unknowns, and we can eliminate two of them, u_2 and λ_2. This is what the Aitken method does. After an initial speedup, the acceleration is not significant. They also suggest a straightforward generalization—a *quadratic extrapolation* that is based on a similar set of equations for $m = 3$,

$$\begin{aligned} p^{(k-3)} &= u_1 + u_2 \quad + u_3 \quad + \dots \\ p^{(k-2)} &= u_1 + \lambda_2 u_2 + \lambda_3 u_3 + \dots \\ p^{(k-1)} &= u_1 + \lambda_2^2 u_2 + \lambda_3^2 u_3 + \dots \\ p^{(k)} &= u_1 + \lambda_2^3 u_2 + \lambda_3^3 u_3 + \dots . \end{aligned} \tag{3.5}$$

Based on some matrix algebra, the authors developed formulas to eliminate unknowns corresponding to the second and third terms. Both the Aitken method and quadratic extrapolation are applied periodically within (3.2). Reported experimental speedup for quadratic extrapolation is 59%.

The presented extrapolation techniques were immediately superseded by an elegant discovery: $\lambda_2(P'') = c$ [Haveliwala and Kamvar 03] (see Section 2.2). The rate of convergence $\|p^{(k+1)} - p^{(k)}\| / \|p^{(k)} - p^{(k-1)}\| \to |\lambda_2/\lambda_1| = c$ is confirmed by experiments. In the Markov chain context, $|\lambda_1| - |\lambda_2| = 1 - c$ is known as stability. The authors also relate eigenvectors corresponding to $\lambda_2 = c$ to spam detection. In extrapolation methods, knowing λ_2 has a profound consequence: we do not need to bother computing it in Equations (3.4) and (3.5).

Extrapolation methods developed in [Haveliwala et al. 03b] utilize this discovery. $\mathbf{A}^d$ extrapolation starts with a representation similar to that in Equation (3.5) under the assumption that $m = d + 1$ and λ_j for $j = 2 : d + 1$ form a system of d-roots of c^d. This means, in particular, that $\lambda_j^d = c^d$. For example, for $d = 2$, $\lambda_2 = c$ and $\lambda_3 = -c$. We see now that

$$\begin{aligned} p^{(k-d)} &= u_1 + u_2 \quad + \dots + u_{d+1} \quad + \dots \\ p^{(k)} &= u_1 + \lambda_2^d u_2 + \dots + \lambda_{d+1}^d u_d + \dots \end{aligned} \tag{3.6}$$

and, hence, the equation $p^{new} = u_1 = \frac{p^{(k)} - c^d p^{(k-d)}}{1 - c^d}$ becomes a viable extrapolation. It is suggested to be used periodically in conjunction with the power method (3.2). For $c = 0.85$ the best speedups correspond to $d = 4$ (25.8%) and $d = 6$ (30%). In our experience this extrapolation works as predicted with large graphs.

3.2. Adaptive Method

There is no need to compute PageRank components exactly. How much accuracy is actually required is an intriguing question that we revisit in Section 6.1. Whatever the required accuracy is, there is a disparity in the magnitudes of PageRank components: authorities of pages differ dramatically, and most values are small (the lower bound $(1-c)/n$ is achieved on pages without in-links). A less transparent observation is that small components converge much faster even when measured on a relative scale $\epsilon_i(k) = \left|p_i^{(k+1)} - p_i^{(k)}\right| / p_i^{(k)}$ [Kamvar et al. 03a]. For example, the authors show that, on a graph of 280,000 nodes (3M links) for $c = 0.85$, around 80% of PageRank components converged ($\epsilon_i(k) < \epsilon = 10^{-3}$) in less than 15 iterations and their average magnitude was 0.37 of the average magnitude of the remaining components. The proposed acceleration, *adaptive method*, simply stops iterating on already converged components.

Assume that $p = (p_N, p_C)$ is a split of p into m not yet converged and $(n-m)$ already converged components. Let A_N and A_C be the corresponding $m \times n$ and $(n-m) \times n$ submatrices of A. Then, we have

$$\begin{pmatrix} p_N^{(k+1)} \\ p_C^{(k+1)} \end{pmatrix} = \begin{pmatrix} A_N \\ A_C \end{pmatrix} \cdot \begin{pmatrix} p_N^{(k)} \\ p_C^{(k)} \end{pmatrix} \qquad \text{or} \qquad \begin{aligned} p_N^{(k+1)} &= A_N p^{(k)} \\ p_C^{(k+1)} &= p_C^{(k)}, \end{aligned} \tag{3.7}$$

where $p_C^{(k)} \approx A_C p^{(k)}$ and we only need to iterate over the nonconverged p_N. Here, $\dim(A_N) = m \times n$ and $\dim(A_C) = (n-m) \times n$. When m becomes smaller, the size of A_N becomes smaller as well, and many computations are avoided. The modified algorithm presented in the paper actually deletes the links corresponding to converged pages from a sparse matrix A. This means decomposition $A_N = [A_{NN}, A_{NC}]$.

The explicit reordering of A is important, and a smart strategy is used to enable housekeeping. The savings are about 20% in overall time. Little hope for the reordering of a several-billion-node web graph currently exists. This restricts the application of this method to smaller graphs such as, for example, a host graph. The adaptive method requires slightly more major iterations k to achieve convergence.

3.3. Block Structure Method

The web has a particular structure. On the very primitive level, there are many more links within the same domains (hosts) than between pages of different domains (hosts). Kamvar et al. brilliantly exploit this idea [Kamvar et al. 03b]. The intuition behind the idea is that blocks roughly represent the whole graph, but there are many fewer blocks than pages and they are more interconnected. Therefore, computing "aggregate" PageRank is easy. If we are also able to approximate PageRank within blocks that is again a series of smaller problems, then we may find a good starting point from which to ignite a full graph algorithm. The authors compute PageRank via the algorithm *blockRank* in several steps. A slightly updated version of this development is presented in Algorithm 2.

Algorithm 2. (Block structure method (*blockRank*).)

```
Input: Given a graph $W$ over nodes $G = \bigcup G_I$, $I = 1 : N$,
a teleportation vector $v$, and a coefficient $c$
Output: Compute PageRank $p$
begin
    for $I = 1 : N$ do
        let $P_{II}$ be a transition matrix over block $G_I$
        for $i \in G_I$ set $v_{I,i} = v_i / \sum_{j \in G_I} v_j$
        $l_I = pageRank(P_{II}, v_I, v_I)$
    end
    for $I = 1 : N$ do
        $\tilde{v}_I = \sum_{i \in G_I} v_i$
        for $J = 1 : N$ do
            $\tilde{L}_{IJ} = \sum_{i \in I, j \in J} P_{ij} l_i$
        end
    end
    let $\tilde{P}$ be a transition matrix corresponding to weighted block structure $\tilde{L}$
    $b = pageRank(\tilde{P}, \tilde{v}, \tilde{v})$
    set $s = (b_1 l_1, b_2 l_2, \ldots, b_N l_n)$
    $p = pageRank(P, s, v)$
end
return $p$
```

We assume that a full graph $W(G, L)$ is defined over nodes G that can be divided into (disjoint) blocks G_I, so that $G = \bigcup G_I$, $I = 1 : N$ (block indices are denoted by capitals, I, J). Let $pageRank(P, p^{(0)}, v)$ denote a general purpose (eigensystem) PageRank-computing algorithm for a transition matrix P starting with $p^{(0)}$, utilizing teleportation vector v, and using whatever acceleration (Kendall distance (see (6.1)) is suggested as the stopping criterion).

The method starts with computing N *local* PageRanks l_I. At the next step we aggregate connectivity information to a block level introducing a weighted directed graph structure $\tilde{L}$ on the set of blocks. A link from a block I to a block J has weight $\tilde{L}_{IJ} = \sum_{i \in I, j \in J} P_{ij} l_i$ combining links between their constituents weighted by their local PageRanks. We now compute the BlockRank authority vector b for a block-level transition matrix $\tilde{P}$. Finally, a vector s that is a rough approximation to full PageRank is assembled. It is used as the initial guess to compute full PageRank.

This algorithm has many useful properties: (1) it does not need much accuracy in computing local PageRanks, (2) it allows obvious parallelization, (3) it may keep within-the-block and between-the-block connectivity data in the core memory, and (4) it benefits from the fact that relatively few blocks change after a subsequent crawl. We return to this method in the personalization context.

Similar ideas for computing HostRank are also advocated by [Broder et al. 04]. The context is reversed from constructing a good approximation s to be used as the initial guess to considering s as a final object that is called HostRank. Other host aggregations, in particular uniform l_I, are suggested. The described methodology has actually been tried on real-life AltaVista 2003 web data.

The block approach to PageRank performs well on large web graphs and, probably, currently constitutes the most practical acceleration technique.

3.4. DAG Structure of the Web

A different, but related, block structure of the web is induced by general graph factorization. Consider blocks equal to (disjoint) strongly connected components (SCC) of W. Since strong connectivity is an equivalence relationship, we can factorize G into some number q of super-nodes, each representing one SCC block. The super-nodes form a *directed acyclic graph* (DAG). The DAG structure effectively defines a partial "flow" order on a super-graph. The largest central SCC contains Yahoo!, MSN, and similar common sites. There are also preceding super-nodes (that have paths leading to the central component) and succeeding super-nodes (that have paths leading from the central component). So, the blocked graph looks like a bow tie. Finding a set of all SCCs is related to a depth-first search (DFS). In graph theory an indexation that enumerates nodes within SCCs contiguously and SCCs themselves in increasing "flow" order is known under the term *topological sort*. For insights on out-of-core memory DFS, see [Chiang et al. 95].

Under such enumeration (corresponding to a permutation of the original P), we get a block upper triangle representation

$$P = \begin{pmatrix} P_{11} & P_{12} & \dots & P_{1q} \\ & P_{22} & \dots & P_{2q} \\ & \dots & \dots & \\ & & & P_{qq} \end{pmatrix}. \tag{3.8}$$

It establishes a relation between strong connectivity and irreducibility (see (2.6)): matrix P is irreducible if and only if there is a single SCC ($q = 1$). The corresponding partitioning of PageRank p leads to significant improvements [Arasu et al. 02]. Equation (2.8) has the form

$$p = cP^T p + b, b = const \cdot v. \tag{3.9}$$

Contrary to some references, $const \neq (1-c)$ but depends on a sink term in Equation (2.8). This does not matter, since the solution will only differ by a scalar. In view of (3.8), Equation (3.9) can be cast as a system of linear equations, $j = 0 : q$,

$$p_j = cP_{jj}^T p_j + f_j, \qquad f_j = b_j + cP_{1j}^T p_1 + \cdots + cP_{j-1,j}^T p_{j-1}. \tag{3.10}$$

We do not know how this method performs in practice on large graphs.

3.5. Spectral and Equation Solvers

So far we tried to accelerate the solution of the spectral problem for eigensystem (3.1) with $\lambda_1 = 1$. Meanwhile, Equation (3.9) reformulates a problem in terms of a linear system:

$$(1 - cP^T)p = b \ (= const \cdot v). \tag{3.11}$$

We would like to reflect briefly on this simple fact. While solving eigensystem (3.1) is based on the simple iterative procedure (3.2), analogous simple iterations

$$p^{(k)} = cP^T p^{(k-1)} + b, \tag{3.12}$$

known as the Jacobi method, can be used to find a solution of the linear system (3.11). While eigensystem formulation of PageRank is well studied, finding an effective solution of the linear PageRank equation (3.11) is a relatively new field.

The connection between (normalized) eigensystem PageRank p_E and linear PageRank p_L is defined by the formula

$$p_L = \frac{1-c}{c \cdot sink(p) + 1 - c} \; p_E. \tag{3.13}$$

Linear PageRank p_L has an application advantage: it depends linearly on the teleportation vector, while traditional eigensystem PageRank p_E does not. As a consequence, it is easy to construct linear combinations of, for example, topic-specific PageRanks that are crucial in personalization.

Linear systems have a rich arsenal of different iterative methods [Axelsson 94]. The simplest ones, *Jacobi* and *Gauss-Seidel*, were successfully tried for Equation (3.9) [Arasu et al. 02, Del Corso et al. 04]. The Jacobi method is already defined by (3.12). The Gauss-Seidel method is similar, but it reutilizes already computed components:

$$p_i^{(k+1)} = (1-c)v_i + c\sum_{j<i} a_{ij}p_j^{(k+1)} + c\sum_{j>i} a_{ij}p_j^{(k)}. \tag{3.14}$$

Thus, the representation (3.8) becomes very useful: many terms in Equation (3.14) vanish. In numerical experiments Gauss-Seidel saves up to 40% of iterations, though it has overhead costs. It has an important advantage of working in place. In experiments it works fine with large graphs. However, Gauss-Seidel has one disadvantage: it is very hard to parallelize.

Two other simple *splitting* methods are the *successive overrelaxation* method (SOR), which depends on a parameter ω, and the similar *symmetric successive overrelaxation method* (SSOR), both known to be faster than Gauss-Seidel. For a detailed description of their application to finite Markov chains, see [Stewart 99]. For general information see [Axelsson 94, Chapters 5–7] and [Golub and Loan 96]. One important practical observation that is more about art than about science is that performance of these methods depends on the enumeration order, and there is experimental evidence that (3.8) is beneficial. The feasibility of SOR for a small W is numerically proven, since a good ω of the form $1+\epsilon$ can be found. Existence of a good ω for large web graphs with more than five to ten million nodes is questionable.

It follows from Equation (3.11) [Haveliwala et al. 03a] that $p = const \cdot (1 - cP^T)^{-1}v = Qv$. Here, $(1 - cP^T)$ is diagonally dominant and invertible. When v has only one nonzero element, p coincides with a column of the matrix Q. So, in principle, Q contains full information for personalization (see Section 5). Computing Q in practice is, of course, infeasible.

3.6. Advanced Numerical Linear Algebra Methods

While previous sections were either dealing with straightforward acceleration of the power method or with traditional methods for linear PageRank formulation related to it, here we review a few approaches that substantially deviate from this paradigm. Indeed, PageRank represents a huge but standard instance of computing a stationary distribution for an irreducible Markov chain that, in turn, is an instance of a general eigensystem problem, which, as we explained, can be translated into a linear system formulation. Recently, an interest in applying modern numerical linear algebra methods to PageRank has emerged. At this moment it is too early to judge how they will perform on real-life web graphs. Since the reviewed methods are based on complex constructions that are beyond the scope of our presentation, we provide no details. The reader can find explanations and further tips in the references mentioned below.

A comprehensive review of numerical methods related to finding stationary distributions of Markov chains can be found in [Stewart 99]. Along with iterative methods, finding equilibrium can be performed by direct methods. A direct projection method for finding Markov chain stationary distribution is presented in [Benzi 04].

In Section 3.3, the block structure method was considered. The idea of aggregating different states (pages in our cases) in several subgraphs is very well known in the theory of computing stationary distributions for Markov chains under the name of *aggregation/disaggregation method*. Its iterative analog was applied to PageRank [Langville and Meyer 04b]. The method is equivalent to the preconditioning of the power method by LU factorization. Extensive analysis of the convergence properties of this algorithm has been done by [Ipsen and Kirkland 04]. We do not know how it performs on real web graphs. Lee, Golub, and Zenios [Lee et al. 03] study the performance on relatively large web graphs of a two-stage decomposition technique related to an aggregation of dangling pages (corresponding to (3.8) with $q = 2$). In the language of Markov chains, the property of having upper triangular form is called *lumpability*. The authors report that the algorithm converges in 20% of the original running time.

The Krylov subspace for a matrix A is defined as

$$K^k(A, v) = span\left\{v, Av, A^2v, \ldots, A^{k-1}v\right\}.$$

Modern numerical linear algebra uses these subspaces in different advanced iterative procedures; [Golub and Greif 04] applies different versions of a celebrated *Arnoldi* algorithm, which is an instance of such a procedure. The Arnoldi algorithm and its variations (e.g., combination with SVD) are used to find the rank-one subspace of $I - P''^T$. This insightful report also studies dependency of PageRank on the teleportation coefficient c (see Section 3.7).

Experiments with an application of other instances of Krylov subspace methods *GMRES*, *BiCG*, and *BiCGSTAB* to linear system PageRank can be found in [Gleich et al. 04]. In many cases these methods outperform the power method and the Jacobi method by 50%, even though their convergence is not stationary, meaning that between iterations the errors behave erratically. Experiments also showed that the convergence of these methods degrades less than the convergence of the power method with a reduction in teleportation when $c \to 1$. The big advantage of these methods is that they are parallelizable. Parallel versions have been benchmarked on several web graphs. For another effort to parallelize PageRank, see [Manaskasemsak and Rungsawang 04].

3.7. Sensitivity Analysis

The teleportation parameter is something that we added to the model for purely technical reasons. When teleportation gradually decreases ($c \to 1$), the negative effects of spam decrease as well, but so does the convergence rate. For this reason it is important to find the dynamics of PageRank $p(c)$ as a function of c [Golub and Greif 04]. Linear system PageRank satisfies the equation

$$p = cP^T p + (1 - c)v. \tag{3.15}$$

We would like to find $p' = \partial p / \partial c$. Differentiating with respect to c, we get

$$p' = cP^T p' + P^T p - v,$$

or after substituting $P^T p$ from Equation (3.15), we get

$$p' = cP^T p' + (p - v)/c = cP^T p' + (1 - c)\hat{v}, \qquad \hat{v} = (p - v)/\left(c(1 - c)\right).$$

This elegant formula shows that p' satisfies the same linear system as PageRank itself for the right-hand side $\hat{v}$. So, (1) stability decreases when $c \to 1$, and (2) the components that deviate the most from the teleportation vector suffer the highest instability.

Any web graph is only an approximate snapshot of reality that depends on many factors from crawler performance and configuration to a setting of parameters defining link pruning to an actual constant appearance of new pages and a decay of old ones. Therefore, computed PageRank is defined over volatile data, and its stability with respect to changes in an underlying web graph becomes a very important consideration that makes or breaks its practical ranking utility. It is also very important computationally, since there is no point in iterating trying to attain accuracy beyond the natural bounds of sensitivity.

The stability of PageRank with respect to small perturbations in the structure of a graph W is studied in [Ng et al. 01b, Ng et al. 01a]. They prove a bound on the change in PageRank caused by a change of links touching (coming to or from) a subset of pages C:

$$\|\tilde{p} - p\| \le \frac{2}{1 - c} E_C, \qquad E_C = \sum\nolimits_{i \in C} p_i.$$

The authors also analyze the HITS algorithm (see Section 4.1) and provide examples that demonstrate that PageRank is more stable than HITS. They show that the sensitivity of HITS depends on *eigengap*, which is the difference between the principal and second eigenvalues of $L^T L$, where the matrix L is the adjacency matrix of a subgraph associated with the HITS algorithm.

Similar results regarding robustness of PageRank are reported in [Bianchini et al. 02, Bianchini et al. 03]. The authors studied subsets (communities) of pages and found useful representations for the quantity E_C introduced above that they called the "energy" of C. They got a slightly better bound

$$\|\tilde{p} - p\| \leq \frac{2c}{1-c} E_C.$$

The next stage in sensitivity research is presented in [Borodin et al. 01] and [Lempel and Moran 03]. In addition to a L_1 stability, these papers also study *rank stability*, which is the stability of indices not with respect to changes in weights but with respect to changes in induced rankings. Ranking measures (for example, (6.1)) are explained in Section 6.1. The authors analyze several models other than PagRank (like, for example, HITS and SALSA) that are introduced in the next section. Their (mostly negative asymptotic) results concerning stability and rank stability were obtained under certain special conditions on graph topology and were formulated as estimates in terms of a number of altered links.

From a practical standpoint, a number of altered links is a less attractive measure than the weighted measure E_C utilizing page importance (for example, crawlers that are mostly responsible for web graph perturbation distinguish between "important" and "unimportant" pages, and E_C provides a convenient abstraction for this). Further improvement to stability estimates in terms of a weighted measure was achieved by [Lee and Borodin 96]. The authors prove that it is enough to take into account only the weights of a "forward" subset $F \subseteq C$ of pages whose out-links (not in-links) have been changed:

$$\|\tilde{p} - p\| \leq \frac{2c}{1-c} \sum\nolimits_{i \in F} p_i.$$

A number of interesting results are obtained in [Chien et al. 01]. In addition to sensitivity analysis, the authors try to address the problem of incremental recomputing of PageRank after small perturbations (*evolution*) in the web structure. Adding some edges results in a new matrix $\tilde{P} = P + E$. The authors rigorously analyze the problem of inserting a single link. The suggested method is to (1) isolate a subgraph W_1, where the influence of a change is significant, from the total web graph; (2) collapse the remaining nodes of $W \backslash W_1$ into a heavy supernode Ω; and (3) compute an approximate PageRank on this new graph structure $W_1 + \Omega$.

Constructing a subgraph can be outlined as follows: assign a unit weight to a node i of a newly inserted link $i \to j$, and propagate this weight from i transferring $c/\deg(i)$ to each out-link including j. After several steps of the breadth-first-search (BFS) algorithm, retain the accessed nodes with a total weight above a prescribed threshold. Transition probabilities to/from the supernode also have instructive definitions. For example, the transition term from a supernode Ω to a node $j \in W_1$ is defined in terms of uniform PageRank p as

$$P_{\Omega j} = \frac{1}{Q} \sum\nolimits_{i \in \Omega} p_i P_{ij}, \qquad Q = \sum\nolimits_{i \in \Omega} p_i.$$

In practice, each crawl results in new IDs for all pages. Finding related IDs is a major problem. A much more humble suggestion is to simply initialize a small portion of outstanding pages with their former PageRank values, distributing the rest of the authority uniformly.

4. Other Approaches

Along with the random surfer model, other usages of hyperlink data were suggested for the purpose of computing the authority weight of a web page. Historically, [Larson 96] was one of the first to apply ideas of bibliometrics to the web. An even earlier pre-Internet attempt to utilize graph structure was done by [Frisse 88]. Another approach [Carrière and Kazman 97] suggests characterizing a page by the number of its in-links and introduces the concept of a neighborhood subgraph. A major boost in the context of web search relevance is associated with the appearance of the HITS algorithm. We survey it and other similar algorithms in this section. We also review applications of PageRank beyond ranking the web pages.

4.1. HITS Algorithm: Hubs and Authorities

A framework similar to PageRank computing introduced in [Kleinberg 99] has several distinctive features: (1) it works with a web subgraph specific to a particular query, rather than with a full graph W; (2) it computes two weights, *authority* weight x_i and *hub* weight y_i, for each page instead of one PageRank weight; and (3) it allows clustering of results for multi-topic or polarized queries such as "jaguar" or "abortion."

A subgraph used in the computations is constructed as follows. The top t (around 200) results recalled for a given query are picked according to a text-based relevance criterion. This is a *root* set. All pages pointed by out-links of a root set are added, along with up to d (around 50) pages corresponding to in-links of each page in a root set (some hubs have enormous amounts of in-links). The result is a *focused* (or *neighborhood* or *augmented*) subgraph corresponding to a query. One of the goals is to allow the engine to report highly relevant pages that do not even contain the query term.

The authority weight of a page is an aggregated significance of the hubs that point to it ("beauty lies in the eye of the beholder"), while the hub weight of a page is an aggregated significance of authorities to which it points. After initialization, the authority weight vector x and the hub weight vector y are subjected to an iterative process (L is an adjacency matrix)

$$\bar{x}_j^{(k+1)} = \sum\nolimits_{i \to j} y_i^{(k)} = L^T y^{(k)}, \qquad x^{(k+1)} = \bar{x}^{(k+1)} / \left\| \bar{x}^{(k+1)} \right\|_2 \tag{4.1}$$

$$\bar{y}_i^{(k+1)} = \sum\nolimits_{i \to j} x_j^{(k)} = L x^{(k)}, \qquad y^{(k+1)} = \bar{y}^{(k+1)} / \left\| \bar{y}^{(k+1)} \right\|_2 \tag{4.2}$$

including some renormalization (on this occasion in L_2, $\|x\|_2^2 = \sum x_i^2$). The limit weights are principal eigenvectors of symmetric matrices: $L^T L$ for the authority vector x and LL^T for the hub vector y. Matrix $L^T L$ is called a *co-citation* matrix, since its (i, j) element $(L^T L)_{ij} = \sum_k L_{ki} L_{kj}$ is equal to the number of pages k jointly pointing to i and j. Matrix LL^T is called a *bibliographic coupling* matrix (see [Lempel and Moran 00, Section 2], for references). The process (4.1)–(4.2) converges under the mild assumption that the principal eigenvalue of $L^T L$ is simple.

Distinguishing between hubs and authorities is a wonderful concept. From a graph theoretic standpoint, the introduced framework relates to a bipartite graph. From an algebraic standpoint, it relates to symmetric matrices rather than to a nonsymmetric matrix P involved in a random surfer model. This explains why subgraph computation is possible—there are no irreducibility and aperiodicity requirements. Computing in a subgraph is a powerful option that allows all kinds of speedups. If there is a subset of highly interlinked pages i (presumably on a single topic), the i-components of principal eigenvectors will be large, while other page components will be small, and so, the principal eigenvectors identify the prevailing topic. This is a fortunate scenario. In the unfortunate scenario we have "topic drift." PageRank, since it operates on a full web graph, is generally considered (maybe more and maybe less correctly) more immune to this problem.

Unlike in the case of the random surfer model, nonprincipal eigenvectors have meaning in HITS. While the coordinates of principal eigenvectors are all positive, subprincipal eigenvectors have both positive and negative components. Pages corresponding to several highest magnitude positive and negative components of a few subprincipal eigenvectors may relate to distinct topics, especially for polarized queries such as "abortion."

At IBM's Almaden Research Center the ideas of Kleinberg were implemented into the HITS (Hyperlink-Induced Topic Search) algorithm [Gibson et al. 98]. The authors analyze the dependency of the results (e.g., top 10 authorities and top 10 hubs) on a root set of size R and on a number of iterations N. Full convergence happens at $N = 50$, but a smaller number of iterations (up to $N = 10$)

actually works well. *Principal* and *nonprincipal* eigenvectors and corresponding communities have been considered, and a parallel between HITS and *latent semantic indexing* [Hofmann 99] has been pointed out. HITS works best for queries representing *broad topics* and in situations where the amount of relevant pages and their hyperlinks is the largest. It has a tendency to generalize a topic. It is easier to spam HITS than PageRank, since a page can always link to great pages and, thus, to increase its hub score. It is trickier for a page to manipulate its own authority score, since linking to a page from a few obscure pages does not help. There is no clarity about industrial anti-spam methods for PageRank.

Further development of HITS relates to the algorithm ARC (Automatic Resource Compiler) [Chakrabarti et al. 98b], which qualifies hyperlinks with nonuniform weights. Specifically, given a link $i \to j$, it considers the text in the vicinity of the <href> tag of a referring page i. Different B-sized windows are examined. The text within the <href> tag is called an *anchor text*. The authors extend it by several characters before and after to an *extended anchor text*. Instead of having uniform connectivity, different edges are assigned different weights w_{ij} proportional to a number of term matches between the B-window and the query string. Now the process (4.1)–(4.2) is augmented with the edges' weights.

Still another attempt to improve HITS is described in the excellent paper [Bharat and Henzinger 98]. It also joins connectivity information with an IR measure of text similarity. The authors try to rectify three problems: (1) many documents on one host pointing to one page on another host that promotes the hub weights of the referring pages, (2) automatically generated links, and (3) topic drift related to well-connected pages not very relevant to the original query. Remedies include demoting the weights of the links from a single host to a single page on another host and pruning pages based on their similarity with the query. The usual IR-cosine similarity is considered as a page weight along with several strategies of pruning. When computing term-similarity, the authors do not use a query string but a *query extension* (its concatenation with the first 1,000 words of pages in a root set). For a related topic of search in context and query extension, see [Finkelstein et al. 02]. For IR issues see [Baeza-Yates and Ribeiro-Neto 99].

The idea of weighted edges in the HITS algorithm is further advanced in [Chakrabarti et al. 98a]. Though this development deals with more general *entities* than HTML pages, in this survey we only concentrate on hyperlinks weighting. Weights—*affinities*—take into account three factors: (1) the default value for any link, (2) whether the source or the destination falls within the root set, and (3) the contribution of every term of a query string. Among other interesting observations are the facts that hubs have only relative value, and so hubs covered by other hubs can be omitted, and that many hubs have heterogeneous collections of links that have to be distinguished (in this regard, see [Bar-Yossef and Rajagopalan 02]).

Different improvements, such as the ones described above, resulted in a project called CLEVER that is an extension of HITS [IBM 99].

Finally, the probabilistic foundations of HITS and, in particular, the PHITS algorithm are discussed in [Cohn and Chang 00]. PHITS is based on *latent semantic indexing* (LSI) [Hofmann 99]. An LSI assumption of the existence of k unknown (*latent*) concepts leads to another development [Achlioptas et al. 01]. In principle, a page importance as an authority or as a hub can be conditioned by each of the k topics leading to $n \times k$ authority and hub matrices $\mathbf{x}$ and $\mathbf{y}$. Using the distributions of terms per topic and a distribution of topics per query, the authors develop the SP algorithm that is a k-dimensional latent semantic analysis version of HITS.

In pre-Internet times, concepts similar to hubs and authorities (of *index* and *reference* nodes) were suggested in [Botafogo et al. 92]. Another early development belongs to Frisse [Frisse 88]. He suggests to increase a page's relevance if pages to which it links are relevant to a query. His paper is also interesting in another regard: the importance of a node representing a piece of medical information is combined from (1) the importance of its forward links and (2) its term-relevance to a query. Therefore, this development already combines link-based and textual relevance. Many ideas similar to PageRank and to HITS are suggested in the Social Network analysis literature. References can be found in the *Status or Rank Prestige* (Section 5.3) and the *Correspondence Analysis* (Section 8.6) sections of the monograph, *Social Networks. Methods and Applications* [Wasserman and Faust 94].

4.2. Random Walk Models for Hubs and Authorities

The SALSA (Stochastic Approach for Link-Structure Analysis) algorithm, introduced in [Lempel and Moran 00, Lempel and Moran 01], utilizes a random walk on two sides of a bipartite graph, authorities (pages with nontrivial in-degree) and hubs (pages with nontrivial out-degree), associated with the query subgraph similar to HITS. To define, for example, the authority random walk, consider transitions consisting of double moves, first back from i and then forward to j:

$$a_{ij} = \sum\nolimits_{k \to i, k \to j} \frac{1}{\textit{in-deg}(i)} \frac{1}{\textit{out-deg}(k)}.$$

It is shown that for an irreducible component the solution can be found in a closed form of a normalized in-degree. The generalization is made for weighted links. The authors report that, in comparison with HITS, SALSA is less vulnerable to the *tightly knit community* (TKC) effect that occurs when a small number of links are strongly interconnected.

A very appealing attempt was made to provide a model for hub/authority vectors that incorporates teleportation [Rafiei and Mendelzon 00]. For example, the authors consider a transformation

$$x_j^{(k+1)} = c \sum\nolimits_{i \to j} y_i^{(k)} / \mathit{out\text{-}deg}(i) + (1-c) v_j / 2.$$

This opens a venue for personalization under the favorable conditions of working in a small subgraph and dealing with a symmetric matrix. While the authors start with the intent of adding teleportation to the HITS framework, due to the presence of a denominator *out-deg*(i), they essentially reintroduce SALSA with teleportation. SALSA with teleportation has the distinction of being rediscovered in [Ng et al. 01a] under the name Randomized HITS.

4.3. Other Models

Here, we briefly survey some other interesting ideas representing diverse theoretical developments in the field.

A major shortcoming of HITS, a potential "topic drift," has led to many efforts. Some of them are described in the previous subsections, but some went further in the direction of changing the basic underlying model. Borodin et al. carefully analyze both HITS and SALSA and suggest a number of modifications [Borodin et al. 01, Borodin et al. 05]. One proposition is, for example, to still compute the authority score x_j as a sum of hub scores y_i over all $i \to j$ but, when computing the hub scores y_i, to only retain pages j corresponding to the K currently highest authorities. Here, K is an algorithm parameter. The motivation is the fact that mediocre authorities are not actually clicked on in "real search" and, therefore, should not count. The resulting algorithm is called Hub-Threshold HITS. The authors also investigate a Bayesian approach to estimate the probability of the link $i \to j$. In addition to the Hub-Threshold HITS and the Bayesian algorithm, the authors suggest other algorithms to construct authority vectors on directed graphs, including the HITS-SALSA "hybrid," the Hub-Averaging algorithm, the Authority-Threshold algorithm, and the Simplified Bayesian algorithm. They analyze stability and rank stability, mentioned previously, along with other mathematical properties of these algorithms.

So far in dealing with a page, we only considered its immediate predecessors and successors. Distant nodes project their influence indirectly through a surfing process. Mathematically, it corresponds to a *first-order* Markov model. A model that directly takes into account more distant relatives with progressive damping factors is also possible (see [Kleinberg 99, Section 5.1.1]). For example, Marchiori

[Marchiori 97] considers the influence of whole paths $\pi = \ldots \to i_3 \to i_2 \to i_1 \to j$ pointing to j via the formula

$$y[j] = \sum\nolimits_{\pi \in \Pi(j)} F \cdot x[i_1] + F^2 \cdot x[i_2] + F^3 \cdot x[i_3] + \ldots,$$

where $\Pi(j)$ is a set of paths pointing to j and F is a factor, $0 < F < 1$. One of the earliest authority measures ever is the Katz index [Katz 53]

$$y_j = \sum\nolimits_{k \geq 1} \sum\nolimits_{i} F^k L_{ij}^k,$$

where L^k is the kth power of the adjacency matrix. Keeping only one term ($k = 1$) of this index, we get the sum of the elements of a column of the adjacency matrix $\sum_i L_{ij}$ known as *In-Degree*, probably the simplest authority index of all. Unfortunately, it is extremely sensitive to spam.

The idea of a hub was further developed by [Bharat and Mihaila 01] into the concept of an *expert* page: a page specifically created to direct users towards resources, or operationally, a page about a certain topic having links to many non-affiliated domains.

Ding et al. attempt to find a framework unifying the PageRank and HITS models [Ding et al. 01]. Both PageRank and HITS become two extreme cases in continua of algorithms that depend on the normalization used to produce a transition matrix P from an adjacency matrix L (no normalization in HITS, dividing by $deg(i)$, $P = D^{-1} \cdot L$, in PageRank).

Another approach to PageRank computing is advocated in [Abiteboul et al. 03]. It is also applicable in conjunction with focused crawling and online, when the whole web graph is not yet constructed. The authors prove that it is not necessary to visit each page once per iteration. Instead, it is possible to focus on important pages as soon as each page is visited frequently enough! Two numbers, cash c_i and history h_i, are associated with each page i. They are initialized to $1/n$ and 0. At each iterative step some page i is selected (with unequal but positive probability). Then, its history is accumulated and its cash is distributed and reset. (The overall cash is preserved, but the overall history $H = h_1 + \cdots + h_n$ increases.) Under broad assumptions, it is proved that the estimates $(h_i + c_i)/(H + 1)$ or simply h_i/H converge to PageRank. This significant computational development is presented in Algorithm 3 (On-Line Importance Computing), where for simplicity we assumed that a graph is strongly connected.

Tomlin starts with the observation that a random surfer model is a special case of a network flow problem on a directed graph [Tomlin 03]. Imagine that a

Algorithm 3. (OPIC.)

```
Input: Given link data $L$
Output: Compute PageRank $p$
begin
    Initialize $c_i = 1/n$, $h_i = 0$, $H = 0$
    repeat
        select $i$ randomly with non-zero probability
        $h_i += c_i$
        for each $i \to j$ do
            $c_j += c_i/deg(i)$
        end
        $H += c_i$
        $c_i = 0$
    until converged
    for each $i$ do
        $p_i = (h_i + c_i)/(H + 1)$
    end
end
return $p$
```

p_{ij} fraction of users follows from i to j during a time unit, so that total incoming j-traffic is $P_j = \sum_{i \to j} p_{ij}$. For simplicity we will assume strong connectivity. In the random surfer model we explicitly set $p_{jk} = P_j/deg(j)$. In a network flow framework, all we have to satisfy is the *conservation* requirement: for each j incoming and outgoing traffic coincides, i.e.,

$$P_j = \sum_{i \to j} p_{ij} = \sum_{j \to k} p_{jk}. \tag{4.3}$$

As soon as constraints consisting of the conservation requirement (4.3) and a normalization requirement $\sum_{ij} p_{ij} = 1$ are satisfied, we are free to chose any p. Following the famous publication [Jaynes 57], it is suggested to maximize the entropy $-\sum_{ij} p_{ij} log(p_{ij})$ subject to introduced constraints. A standard matrix-balancing algorithm for maximum entropy problems is considered. Still another attempt to build PageRank subject to certain constraints (of desirability for certain pages to have particular PageRank) is presented in [Tsoi et al. 03].

Four different modifications to the PageRank model take page decay (link rot) into consideration [Eiron et al. 04]; see also [Bar-Yossef et al. 04]. The guiding principle of this development is to reduce the PageRank of pages having links to bad pages (e.g., 404 HTTP return code). For example, a "push-back" modification returns a fraction of a current bad page PageRank value to its contributors. It is shown that, algebraically, it means that a usual transition matrix P is multiplied on the right by some row-stochastic matrix B. Another simpler suggestion, the "self-link" algorithm, is to introduce self-links that are followed with some page-dependent probability γ_i inversely proportional to the number of bad out-links. A self-link always promotes a page authority.

Xi et al. concentrate on the generalization of underlying data [Xi et al. 04]. While previous models worked with web hyperlinks, the authors add into play users (*social network*) and, in principle, other objects of different types. While links between objects of the same type, *intra-links*, were traditionally used to build authority vectors on directed graphs, now we also have *inter-links* that link objects of different types. They suggest a framework, *link fusion*, incorporating different authority weights, one per type, that are reinforced through both intra- and inter-links.

4.4. Other Applications

The outstanding popularity of PageRank in web search ranking naturally suggested its use in other contexts. In order to keep our focus on the major application of scoring web page authorities, we only briefly review some of them. Many other examples can be found in the extensive literature on random walks and Markov chains.

It is appropriate to start with two applications of PageRank in web search technology itself other than for ranking search results. The first application deals with the important topic of spam, pages that try fraudulently to mislead search engines about their importance. Among others, spam practices include adding to a page some good unrelated content (sometimes in invisible font) to increase its coverage and creating artificial pages that link to a spam target. Since the smallest PageRank of a page i equals $cv_i = c/n$ under the assumption of uniform teleportation, creating M supporting links immediately guarantees a score of at least $m = Mc/n$ for a spam page (actually $1/(1-c^2)$ times more when a spam target links back to supporting pages). A judicial use of trusted pages to which to teleport instead of uniform teleportation is suggested in [Gyöngyi et al. 04b]. The resulting PageRank is called TrustRank. It can be used in spam detection [Gyöngyi et al. 04a] through careful approximation of m (effective mass) with the linear combination of PageRank and TrustRank.

Another important problem in web search is deciding on what to crawl. Naïve crawling of everything is infeasible and has no point since certain sites can be crawled almost indefinitely. Therefore, crawlers are designed to pay more attention to some pages than to others. They do so based on *importance metrics* associated with each of the existent pages [Cho and Garcia-Molina 00]. PageRank constitutes a valuable instance of such a metric. Topic-specific PageRank may serve as an importance metric for a focused crawler [Chakrabarti et al. 99, Abiteboul et al. 03].

A *Trust Network* is a directed weighted graph of users where an edge $i \to j$ with weight s_{ij} represents the trust (agent) i has to j. For our purpose, semantics and application of trusts are not important. In particular, we will assume that $0 \leq s_{ij} \leq 1$. It is important that trusts reflect a local ith viewpoint of its out-neighbors, and furthermore it is very desirable to have global reliability weights of a distant j. For example, a recommender system would weight different jth ratings according to their reliability. Reliability could be i-personalized (one-node teleportation) or uniform (uniform teleportation). Restricting ourselves to a uniform case, we can define global reliability by using a random surfer model, in which a surfer moves from agent i to agent j with probability $s_{ij}/\sum_{i \to k} s_{ik}$. The resulting PageRank is introduced in [Kamvar et al. 03d] under the name EigenTrust. A personalized analog is also very beneficial in a recommender system.

PageRank has been used recently in the context of keyword searching in relational databases when a user does not know its schema [Bhalotia et al. 02] without any query language. A proposed technique is called BANKS (Browsing and Keyword Searching). It models database tuples as nodes and cross references between them as edges. PageRank naturally appears in this context to weight nodes. Along similar lines, a *database graph* corresponding to a relational database and a set of queries is introduced in [Geerts et al. 04]. The objective is to show only the most significant tuples. In a huge database, this objective cannot be served by the traditional ORDER BY operator. The paper contains a significant technical apparatus that is beyond our scope. Briefly, two database tuples i and j are linked, $i \to j$, with respect to the set of queries if j is a valid target for a query with sources i. Now, a random walk can be defined, and PageRank or HITS can be constructed.

To further illustrate applications of a random surfer model, we point out two examples. The first example is learning word dependencies in natural language processing [Toutanova et al. 04]. The second example is the problem of rank aggregation. When one has multiple partial orderings over a set of indices $1{:}n$, it is desirable to find a permutation that minimally violates the given orderings. An example application is relevance in meta-search. One elegant way to solve this problem is to associate it with the problem of a random walk over $1{:}n$. Dwork et al. suggest several such models [Dwork et al. 01]. For example, starting at i, with equal probability select one of the given partial orders, and then with probability $1/(k+1)$ move to one of the k indices preceding i in this partial order or stay at i. The resulting PageRank provides a score that is used to construct aggregated order.

5. Personalization and PageRank Sets

Among the many approaches to customization of search such as, for example, those based on user profiles, we deal in this section with personalization that utilizes PageRank.

Computation of PageRank is fundamentally an offline process. In this regard it has an advantage over the HITS algorithm performed online, since its query time is small. On the other hand, while HITS depends on a particular query, PageRank is query-blind. It relies on nothing particular to an individual search. The idea of using a nonuniform teleport to personalize PageRank was proposed early in [Brin and Page 98, Page et al. 98]. Adequate technology to implement this idea was developed gradually. For an overview of many personalization efforts, see [Haveliwala et al. 03a].

5.1. Topic-Sensitive PageRank

The idea of a topic-sensitive PageRank is developed in [Haveliwala 02b]. To compute topic-sensitive PageRank, a set of top topics T_j from some hierarchy (16 topics from ODP in this study) are identified, and instead of the uniform personalization vector v in (2.4), a topic specific vector $v_{(j)}$,

$$v_{(j)i} = 1/|T_j|, \text{ if page } i \text{ belongs to topic } T_j, \text{ and } v_{(j)i} = 0 \text{ otherwise,} \tag{5.1}$$

is used for teleportation leading to the topic-specific PageRank $p_{(j)}$.

To ensure irreducibility, pages that are not reachable from T_j have to be removed. (Another solution would be to allow a mixture with a small fraction of the uniform teleportation distribution.)

With respect to the usage of topic-sensitive PageRank, two scenarios are discussed. In the first scenario, the appropriate topic is deduced from the query. In the second scenario, some context is provided. When the topic is deduced from the query, we set probabilities

$$P(T_j|query) \propto P(T_j) \prod\nolimits_{terms \in query} P(term|T_j). \tag{5.2}$$

A (soft) query-specific PageRank is now defined as a blend of topic-specific PageRanks

$$p_i = \sum\nolimits_j P(T_j|query)p_{(j)i}.$$

Maintaining about one hundred topic-sensitive PageRanks is feasible. While the described approach provides for personalization indirectly through a query, user-specific *priors* (e.g., directly selected by a user) can be taken into account in a similar fashion. Overall, in different demo versions this type of personalization is used the most, which indirectly confirms its practical usefulness.

5.2. Block Personalization

If we restrict personalization preferences to domain (host) blocks, the blockRank algorithm provides clear opportunities. Following [Kamvar et al. 03b] and using the notation of Section 3.3, we utilize equations

$$b = pageRank(\tilde{P}, \tilde{v}, \tilde{v}), \tag{5.3}$$

$$p = pageRank(P, s, v). \tag{5.4}$$

Personalization is achieved by using nonuniform teleportation v. Here, the construction of $\tilde{v}$ (from v by aggregation) and of an initial guess s for the overall process (from b and local PageRanks) remains the same as in the blockRank algorithm. What is different is the nonuniform choice of teleportation vector v. Picking an arbitrary v is infeasible; therefore, only N degrees of freedom are allowed: instead of v, we deal with an N-dimensional personalization v_N. The surfer travels over the web and from time to time teleports to one of the hosts (home pages) proportionally to distribution v_N. Computing personalized blockRank b this way is much easier than computing p, since $N << n$. A full teleportation vector is defined as $v(j) = v_N(J)p(j|J)$, and a conditional probability $p(j|J)$ can be identified with components of the Jth local PageRank $l_J(j)$.

Though a few iterations to approximate p in (5.4) would suffice, even that is not feasible in query time for a large graph. On the other hand, this development is very appealing since block-dependent teleportation constitutes a clear model.

5.3. Scaled Personalization by Jeh and Widom

We now describe the outstanding results of [Jeh and Widom 02b]. The authors developed an approach to computing *Personalized PageRank vectors* (PPV). Each PPV is a solution of the equation (in this subsection we use notation $C = 1 - c$ for consistency with the paper)

$$p = (1 - C)P^T p + Cv, \tag{5.5}$$

where the personalization vector v relates to user-specified bookmarks with weights. More specifically, v is a linear combination of some *elementary* vectors $x_a(i) = \delta_{ai}$ (each teleporting to a single "bookmark" page a)

$$v = \sum\nolimits_a g_a x_a.$$

If, for example, $g_a = 1/n$, we get $v = e$, and p is a uniform PageRank. The authors suggest a framework that, for bookmarks a belonging to a highly linked subset of hub pages H, provides a scalable and effective solution to build PPV. Let $r_a = (r_a(b))$ be a solution of Equation (5.5) corresponding to a basis $v = x_a$. A general PPV can be expanded in a linear sum of such basis vectors r_a. We would like to provide some intuitive justifications before diving into technical details. The presented framework shares three common practices: (1) it leverages already precomputed results, (2) it provides cooperative computing of several interrelated objects, and (3) it effectively encodes the results. The precomputed objects constitute what is called the *hub skeleton*. They can be leveraged in computing any PPV, since a general PPV projection on a hub can be subtracted and what remains is "simpler" and sparse. To build a projection, the authors express PPV as sums over all the passes connecting two pages: an origin a and a destination b. Longer passes contribute more to a concept of expected P-distance, but they affect the result less, which brings us to a modified concept of an inverse P-distance. Now we summarize the major achievements of the paper.

- Consider the concept of *expected distance* defined as

$$d(a, b) = \sum\nolimits_{t:a \to b} P[t] l(t). \tag{5.6}$$

  Here, $t =< t_0, t_1, \ldots, t_k >$ varies among all paths or *tours* from $t_0 = a$ to $t_k = b$ ($t_j \neq b$ for $j < k$) with potential cycles, length $l(t) = k$. $P[t] = \prod_{j=0:k-1} 1/deg(j)$ is the probability of a tour t according to the uniform random walk model. We set $d(a, b) = \infty$ if there is no tour from a to b, and we set $d(a, a) = 0$. The expected distance is the average number of steps that a random surfer takes to get from a to b for the first time. By modifying (5.6) the authors prove that the basis vectors can be expressed in terms of *inverse P-distance*

$$r_a(b) = \sum\nolimits_{t:a \to b} P[t] C (1 - C)^{l(t)}. \tag{5.7}$$

- Instead of storing n elements of r_a, a sparse partial vector $r_a - r_a^H$ that has many zeros can be stored. Here r_a^H is defined as in (5.7), but only tours touching H are used. A tour t touches H if $t_j \in H$ for some $0 < j < k$; in particular, such tours are at least 2-long. For large, highly linked H most tours touch it, and so a partial vector is sparse.

- To restore $r_a = (r_a - r_a^H) + r_a^H$, the last term is required. It is shown that only a hub skeleton that is the array of values of $r_a(h)$ over hub pages is needed for this. Beautifully,

$$r_a^H = \frac{1}{C} \sum\nolimits_{h \in H} (r_a(h) - Cx_a(h))(r_h - r_h^H - Cx_h). \tag{5.8}$$

- Let $O(i, a)$ be out-neighbors of a, $i = 1 : deg(a)$. From a page a the surfer moves with probability $(1 - C)/\deg(a)$ to each of $O(i, a)$. The following decomposition theorem is proved:

$$r_a = \frac{1 - C}{\deg(a)} \sum\nolimits_{i=1:\deg(a)} r_{O(i,a)} + Cx_a.$$

  It states that the a-view of the web is an average of $O(i, a)$-views of the web for out-neighbors endorsed by a plus extra self-importance.

- The decomposition theorem is instrumental in computing r_a. Several approximate iterative procedures are suggested. They are used to compute (a) the partial vectors $r_a - r_a^H$, $a \in H$, (b) the hub skeleton $\{r_a(h) | a, h \in H\}$, and (c) the web skeleton $\{r_a(h) | a \in G, h \in H\}$.

5.4. Query-Dependent PageRank

Richardson and Domingos suggest a personalization process that actually modifies the random surfer model [Richardson and Domingos 02]. Their *Intelligent Surfer* takes query q into account:

$$p_j^{(k+1)} = c \sum_{i \to j} P_q(i \to j) \cdot p_i^{(k)} + (1 - c)P'_q(j). \tag{5.9}$$

Let $R_q(j)$ be a measure of relevance between a query q and a page j (TF/IDF similarity can be used). The authors suggest using

$$P_q(i \to j) = \frac{R_q(j)}{\sum_{i \to k} R_q(k)}, \qquad P'_q(j) = \frac{R_q(j)}{\sum_{k \in G} R_q(k)}.$$

Here, both the link weights and the teleportation distribution are defined in terms of the relevance between page content and a query. So, the constructed term-dependent PageRanks are zero over the pages that do not contain the term. Based on this observation, the authors elaborate on the scalability of their approach. Though computing billions of link weights $P_q(i \to j)$ is not easy, the idea of blending IR methods into a random walk model is appealing (see also [Chakrabarti et al. 98b, Brin and Page 98]). For the opposite trend of leveraging connectivity data in an IR setting, see [Jeh and Widom 02a].

5.5. Personalization in HITS

Personalization of PageRank has a clear input: the personalization (teleportation) vector v. As we explained in Section 4.2, [Rafiei and Mendelzon 00] extends the concept of teleportation to authority/hub setting (the authors talk about HITS, but actually it is SALSA). Though they work with a uniform v, their approach, in principle, can be used for personalization.

Chang et al. have been interested in customization [Chang et al. 00]. They present a way to personalize HITS by incorporating user feedback on a particular page j. One way of doing this is simply to raise page j's authority and to distribute it through the propagation mechanism. This, however, runs into the trouble of an abnormal increase of closely related pages (a phenomenon known as *nepotism*). Instead, the authors suggest an elegant way to increase the authority of a page j indirectly through a small change in the overall graph geometry that is consistent with user feedback and is comprehensive in terms of effects on other pages. To do so, notice that an authority vector x is a fixed point of the following transformation:

$$\bar{x}_j = \sum\nolimits_{k,i} L_{ki} L_{kj} x_i.$$

Our goal is to increase its jth component. Its gradient $grad\,(\bar{x}_j) = \{\partial \bar{x}_j / \partial L_{ki}\} = \{L_{kj} x_i\}$ can be used to adjust elements L_{ki} of the original adjacency matrix

$$\bar{L} = L + \gamma \cdot grad\,(\bar{x}_j)\,.$$

For small γ, Taylor's formula guarantees consistency with feedback (the authors suggest renormalizing $\bar{L}$, which would also allow for negative feedbacks). So HITS is invoked twice: first, to compute x and, second, to compute $\bar{x}$ after a matrix L is updated. The study is performed not for a query search but on an online index *Cora* of computer science literature. Among other difficulties, implementing this idea for a query search would require inclusion of a page j into a query subgraph.

6. Nuts and Bolts

Computational challenges and the application importance of PageRank led to many special developments facilitating its computation beyond just accelerating its convergence. Those developments are surveyed in this section. In practice, infrastructural issues such as, for example, fast access to link data or appropriate pruning of links are most important in speeding up the computation of PageRank and getting useful results.

6.1. Stopping Criteria

The essential question for PageRank computing is how far we would like to iterate. Since our goal is to rank pages, there is no sense to iterate beyond the accuracy that establishes a linear order over the pages (we will see shortly that even this is too restrictive). While the simplest stopping criterion for the iterations (2.1) is $\|p^{(k+1)} - p^{(k)}\| < \epsilon$ in L_1 norm, a more relevant one would be based on an estimation of how the resulting ranking is affected.

The ranking order defines a permutation σ over the pages. Let $\sigma(i)$ be the rank of page i (the smaller the better). Classic dissimilarity measures between two ranking orders [Dwork et al. 01] include Kendall tau K distance and Spearman's footrule F distance

$$K(\sigma_1, \sigma_2) = \# \{(i < j) \mid sgn\,((\sigma_1(i) - \sigma_1(j))/(\sigma_2(i) - \sigma_2(j)) = -1\}, \quad (6.1)$$

$$F(\sigma_1, \sigma_2) = \sum\nolimits_i |\sigma_1(i) - \sigma_2(i)|. \quad (6.2)$$

Both distances can be normalized, since the maximum values for K and F are $n(n-1)/2$ and $2\lfloor (n+1)/2 \rfloor \lceil (n-1)/2 \rceil$, respectively. The introduced concepts have probabilistic significance. For example, Kendall tau distance defines the probability for a random pair of pages to be reversed by two rankings. It alternatively can be interpreted as a maximum likelihood. Asymmetric measures penalizing top results' distortions are used in proprietary software. Ranking measures of controlling convergence have been used in actual numerical studies [Haveliwala 99, Kamvar et al. 03b, Kamvar et al. 03c]. The authors report a good correlation between a simple L_1 convergence and a rank convergence. Our experiments confirm this result, but they also show that at some moment rank convergence reaches its saturation, while L_1 convergence monotonically improves. Therefore, monitoring of a simple L_1 convergence is only good in a certain range. Even more important in the evaluation of rank convergence is the issue of rank stability discussed in Section 3.7. There is no benefit in iterating beyond a bound imposed by rank stability. We do not know of any studies clarifying this point.

Beyond serving as a measure controlling PageRank convergence, Kendall tau distance has another PageRank application. It is rooted in a desire to map PageRank values into a finite set of values, one per range interval. Such a process is called *vector quantization*. Its goal is to encode all the values of PageRank (float-point numbers) by a relatively small number of indices. This allows for a significant compression. Such compression is very important for a general-purpose PageRank, but even more so for topic-specific PageRanks. As a result of this lossy compression, some violations happen and Kendall tau distance can be used to control the errors [Haveliwala 02a].

A direct application of rank-based measures to PageRank is somewhat beyond the point, since we are not in the business of establishing a linear order for the web. What we actually need is an assertion that ranks within an "average" query result set do not differ wildly [Haveliwala 99]. Result sets for an editorially selected set of queries can be used to monitor convergence, averaging dissimilarity measures imposed by PageRank over each result set. Moreover, in realistic relevance analysis, even a whole recalled result set is too large. Correspondingly, it is important to be able to compare just two *top k lists* τ_1 and τ_2 corresponding to top results D_1 and D_2 for each order. Those can be different pages and [Fagin et al. 03] analyzes this problem. For example, let $\sigma \succ \tau$ if τ is a restriction of σ. For two top k lists τ_1, τ_2 we can set

$$K_{min}(\tau_1, \tau_2) = min\left\{K(\sigma_1, \sigma_2)|\ \ \sigma_1 \succ \tau_1, \sigma_2 \succ \tau_2\right\},$$

where σ_1, σ_2 are defined on $D = D_1 \bigcup D_2$. Assessing results on the top k lists for a set of predefined queries appears to be practical and to reflect application specifics. In reality, rank violations in the first few results have a much more severe consequence for web search than rank violations down the tail. Putting it together, a reasonable modification of, for example, (6.2) can be given by the formula

$$F'(\sigma_1, \sigma_2) = \sum\nolimits_{q \in Q} \left\{ \sum\nolimits_{i \in R(q)} |\sigma_1(i) - \sigma_2(i)|/i^{\gamma} \right\},$$

where a query q goes over a set of predefined queries Q, i enumerates pages within query result sets $R(q)$, and $\gamma > 0$ discounts results in the tail.

6.2. Computing Infrastructure

The major computational cost of PageRank is in matrix-vector multiplication. To understand further reasoning, assume that links $i \to j$ are stored in an "array" L that is sorted first by i and then by j, and consider Algorithm 4 which implements this operation. This implementation of a sparse multiplication $y = P^T x$ accesses components x_i and edges (i, j) in a natural order. Alas, vector y_j is accessed in a random fashion. When it does not fit in memory, disastrous paging happens.

A straightforward improvement is suggested in [Haveliwala 99] and is presented in Algorithm 5. Divide n nodes into β blocks of length n/β (the last one is potentially smaller). Let b be a block index. Divide links L into β subsets $L[b] = \{(i, j) : j \in \text{block } b\}$, each sorted first by i and then by j. As before, x_i and edges (i,j) are accessed in a natural order. In addition, components y_j of each block b are now in memory and no paging happens. There are different flavors of this idea. Assuming that each time we read α sources i in memory,

Algorithm 4. (Naïve matrix-vector multiplication.)

```
Input: Given $x$ and array $L = \{(i, j)\}$
Output: Compute $y = P^T x$
begin
    for each $j$ do
        $y[j] = 0$
    end
    while $L$ is not exhausted do
        read a batch of $(i, j)$ from $L$ in memory
        for each $i$ in a batch do
            $z = 1/deg(i)$
            for each $j$ such that $(i, j)$ is in a batch do
                $y[j] += z \cdot x[i]$
            end
        end
    end
end
```

and that the average $\deg(i)$ is d, we can estimate optimal β. We need space $\alpha + n/\beta + \alpha \cdot d/\beta$ for (1) the sources x vector portion (α elements), (2) the targets y vector portion (n/β elements), and (3) the $L(b)-$ batch portion of links ($\alpha \cdot d/\beta$), assuming average out-degree d. If the available memory is M, the constraint is

$$\alpha + n/\beta + \alpha \cdot d/\beta \leq M.$$

In the simplest case, $\alpha = n/\beta$, which brings us to the condition $2n/\beta + nd/\beta^2 \leq M$. Blocked algorithms are important. They provide for parallelization in an industrial setting.

A different implementation of sparse multiplication, the sort-merge algorithm, is suggested in [Chen et al. 02]. Each source page i impacts a destination page j by amount $w = x_i/\deg(i)$. The problem with Algorithm 4 is that while the sources i come in order, the targets j come randomly. To avoid huge associated I/O, the sort-merge algorithm saves *packets* that are target/impact pairs (j, w). Over the span of a major iteration, packets are kept in a memory buffer. When full, the buffer is compressed (j-aggregation) and if this does not help, it is written to a disk. All the packets on disk are then sorted by destination j, and only then accumulation happens.

Algorithm 5. (Block matrix-vector multiplication.)

```
Input: Given $x$ and $\beta$ arrays $L[b], b = 1 : \beta$
Output: Compute $y = P^T x$
begin
    for $b = 1 : \beta$ do
        for $j$ in block $b$ do
            $y[j] = 0$
        end
        while $L[b]$ is not exhausted do
            read a batch of $(i, j)$ from $L[b]$ in memory
            for each $i$ in a batch do
                $z = 1/deg(i)$
                for each $j$ such that $(i, j)$ is in a batch do
                    $y[j] += z \cdot x[i]$
                end
            end
        end
    end
end
```

6.3. Hyperlinks

Hyperlinks have a very different nature [Bharat and Henzinger 98]. Usually links corresponding to the same host (domain), called *intrinsic* or *internal* links, are not very trustworthy and are either pruned or assigned smaller weights. Some internal links (e.g., reference to a designer page, navigational links, copyright warnings, or disclaimers) are simply useless. A finer extraction of still-useful intrinsic links based on analysis of frames is possible. Sometimes the removal of intrinsic links is blank wrong, as, for example, in the case of the GeoCities site, where homepages with already sparse link structure are further pruned if they point to each other. The rest of the hyperlinks are called *transverse* or *external* links. There are different reasons for pruning some of these links as well. Potential examples include advertisements, mass endorsement, and highly interlinked mailing archives. Links, in principle, may have different nonnegative weights [Kleinberg 99, Section 5.1.1], so that weight-proportional rather than uniform distribution is used. In principle, a measure of page decay (link rot) can be taken into account [Bar-Yossef et al. 04, Eiron et al. 04].

Though a full discussion of link filtering is beyond our scope, we would like to point to the concept of *nepotistic* links that are created for reasons other than target-page merit [Davison 00]. The author selects several attributes to be used in a machine learning procedure (C4.5) trained to predict nepotistic links. The largest class of nepotistic links is link-based spam [Gyöngyi et al. 04b] discussed in Section 4.4.

An effective web-link database with API appropriate for its surfing is known as a *connectivity server* [Bharat et al. 98, Randall et al. 02]. The quality of a connectivity server is crucial to PageRank. Contemporary information on this subject is highly proprietary.

In Section 3.4, we introduced an upper triangular block representation for a transition matrix. A proper web page enumeration is very important. An extensive related study of web structure [Kumar et al. 00] reflects, in particular, on its "bow tie" form with the diameter of central SCC. It also contains a large material regarding power laws for the number of the in/out-links and for the number of undirected/directed strongly connected components. In particular, the central SCC at the time amounted to around 28% of the total pages. According to [Broder et al. 00], its diameter is at least 28. There are also a number of disconnected components. This work was partly performed at AltaVista and has all the traits of dealing with real data. An immediate application is the following: if a personalized vector v contains few bookmarks, the PageRank for the pages located up the stream of these bookmarks would be zero. Regarding information on power laws for the Internet, also see [Faloutsos et al. 99]. For in/out-degree power law distributions (exponents are 2.15/2.75), also see [Guillaume et al. 02]. More specifically, the numbers of pages with in-degree or out-degree j are proportional to $1/j^{\gamma}$. A categorical distribution whose jth most likely outcome is proportional to $1/j^{\gamma}$ is also known as the Zipf distribution [Adler and Mitzenmacher 01]. For a thorough introduction into the fundamentals of the power law, see [Mitzenmacher 03].

Other statistics for different web graphs, including sizes of strongly connected components, are presented in [Dill et al. 01]. Regarding distribution of PageRank values, see [Pandurangan et al. 02]. Among other things, the authors present web graph models that utilize PageRank: a new edge is generated by taking the current PageRank into account.

In the study [Kumar et al. 99] different important topics, such as in/out-degree distributions, different pruning strategies, and finding strongly interconnected web communities, are considered. Around 20,000 of such are found. It also addresses the phenomenon of *mirrors* and *shingles* (that is, almost identical pages, see [Broder et al. 97]). Merging of such pages (duplicates) is ubiquitous and very important.

On the *duplicate hosts* detection problem, see [Henzinger 03]. For an inverse attempt to divide pages into *pagelets* based on analysis of frames, see [Bar-Yossef and Rajagopalan 02]. An approach to study web structure that exploits both link structure and textual information (*information foraging*) is proposed in [Pirolli et al. 96].

In reality, any static web graph only gives certain approximation to the real web. Indeed, crawling of many sites is prohibited, while other sites can be crawled indefinitely, and the crawler exercises certain heuristics to prevent this. Another difficulty is related to database-driven dynamic pages. All these reasons lead to the presence of many dangling pages (only a portion of which is genuinely dangling). They constitute a web frontier, a term coined in [Eiron et al. 04].

6.4. Link Data Compression

Reading link data (matrix L) into memory is an expensive process, and it is performed during each of many iterations. Therefore, potential compression of this data can be beneficial [Randall et al. 02, Suel and Yuan 01]. One suggestion is to exploit the closeness of many references: if a page has (sorted) links $l_1, \ldots, l_k$, (e.g., 1,000,007, 1,000,052, 1,000,108), then only the first link and the differences $d_j = l_j - l_{j-1}$ can be stored (e.g., 1,000,007, 45, 56). Other smart strategies are also considered.

Another source of potential compressibility is identified in [Suel and Yuan 01]. Some pages have very high in-degree. In other words they use in-link data disproportionally frequently. The authors suggest encoding the first highest q in-degree pages by Huffman codes (see also [Adler and Mitzenmacher 01]).

An overview of web compression issues can also be found in [Witten et al. 99]. Unlike many other nice ideas, link compression is actually used in real connectivity servers. Probably the most advanced compression rates of 3.08 bit per link (2.89 bits per link for transposed graph) were achieved for a 118 M graph with 1 G of links by [Boldi and Vigna 03]. It is not exactly clear how much such compression rates degrade performance during the decoding stage.

7. Conclusions

The development of methods for the efficient computation of PageRank and similar authority vectors over directed graphs is a vibrant area of contemporary research. It has a grand application: search engine relevance and personalization. This application explains a diffusion of ideas related to leveraging content-based information into the austere formulation of random walks over a graph. PageRank also has a grand challenge: being the largest matrix computation in the world. This challenge dictates the emergence of structural and computational methods coming from linear algebra and graph theory. We believe that these trends will continue their momentum.

Acknowledgments. Many issues related to this survey were discussed with my colleagues at Yahoo!. I thank them all. In particular, I thank Byron Dom, Farzin Maghoul, and Jan Pedersen for many useful discussions. I am also grateful to David Cossock and Kostas Tsioutsiouliklis for their meticulous editing. This text also greatly benefited from the many insightful suggestions of dedicated reviewers. I am truly indebted to them.

References

[Abiteboul et al. 03] Serge Abiteboul, Mihai Preda, and Gregory Cobna. "Adaptive On-Line Page Importance Computation." In *Proceedings of the Twelfth International Conference on World Wide Web*, pp. 280–290. New York: ACM Press, 2003.

[Achlioptas et al. 01] Dimitris Achlioptas, Amos Fiat, Anna Karlin, and Frank McSherry. "Web Search via Hub Synthesis." In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS '01)*, pp. 500–509. Los Alamitos, CA: IEEE Computer Society, 2001.

[Adler and Mitzenmacher 01] M. Adler and M. Mitzenmacher. "Toward Compressing Web Graphs." In *Proceedings of the IEEE Data Compression Conference (DCC '01)*, pp. 203–212. Los Alamitos, CA: IEEE Computer Society, 2001.

[Ahuja et al. 93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[Arasu et al. 02] Arvind Arasu, Jasmine Novak, Andrew Tomkins, and John Tomlin. "PageRank Computation and the Structure of the Web: Experiments and Algorithms." In *Proceedings of the Eleventh International Conference on World Wide Web, Alternate Poster Tracks*. Available from World Wide Web (http://www2002.org/CDROM/poster/173.pdf), 2002.

[Axelsson 94] Owe Axelsson. *Iterative Solution Methods*. New York: Cambridge University Press, 1994.

[Baeza-Yates and Ribeiro-Neto 99] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Reading, MA: Addison-Wesley, 1999.

[Bar-Yossef and Rajagopalan 02] Z. Bar-Yossef and S. Rajagopalan. "Template Detection via Data Mining and Its Applications." In *Proceedings of the Eleventh International Conference on World Wide Web*, pp. 580–591. New York: ACM Press, 2002.

[Bar-Yossef et al. 04] Z. Bar-Yossef, A. Broder, R. Kumar, and A. Tomkins. "Sic transit gloria telae: Towards an Understanding of the Web's Decay." In *Proceedings of the Thirteenth International Conference on World Wide Web*, pp. 328–337. New York: ACM Press, 2004.

[Benzi 04] M. Benzi. "A Direct Projection Method for Markov Chain." *Linear Algebra and Its Applications* 386 (2004), 27–49.

[Bhalotia et al. 02] Gaurav Bhalotia, Arvind Hulgeri, Charuta Nakhe, Soumen Chakrabarti, and S. Sudarshan. "Keyword Searching and Browsing in Databases using BANKS." In *Proceedings of 18th International Conference on Data Engineering (ICDE '02)*, pp. 104–111. Los Alamitos, CA: IEEE Computer Society, 2002.

[Bharat and Henzinger 98] Krishna Bharat and Monika Henzinger. "Improved Algorithms for Topic Distillation in a Hyperlinked Environment." In *Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 104–111. New York: ACM Press, 1998.

[Bharat and Mihaila 01] K. Bharat and G. A. Mihaila. "When Experts Agree: Using Experts to Rank Popular Topics." In *Proceedings of the Tenth International Conference on World Wide Web*, pp. 597–602. New York: ACM Press, 2001.

[Bharat et al. 98] K. Bharat, A. Broder, M. R. Henzinger, P. Kumar, and S. Venkatasubramanian. "The Connectivity Server: Fast Access to Linkage Information on the Web." In *Proceedings of the Seventh International Conference on World Wide Web*, pp. 469–477. Amsterdam: Elsevier Science Publishers, 1998.

[Bianchini et al. 02] M. Bianchini, M. Gori, and F. Scarselli. "PageRank: A Circuital Analysis." In *Proceedings of the Eleventh International Conference on World Wide Web, Alternate Poster Tracks*. Available from World Wide Web (http://www2002.org/CDROM/poster/165.pdf), 2002.

[Bianchini et al. 03] Monica Bianchini, Marco Gori, and Franco Scarselli. "Inside PageRank." Technical Report, University of Siena, 2003.

[Boldi and Vigna 03] P. Boldi and S. Vigna. "The WebGraph Framework I: Compression Techniques." Technical Report TR-293-03, Universita di Milano, Dipartimento di Scienze dell'Informazione, 2003.

[Borodin et al. 01] Allan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal, and Panayiotis Tsaparas. "Finding Authorities and Hubs from Link Structures on the World Wide Web." In *Proceedings of the Tenth International Conference on World Wide Web*, pp. 415–429. New York: ACM Press, 2001.

[Borodin et al. 05] Allan Borodin, Gareth O. Roberts, Jeffrey S. Rosenthal, and Panayiotis Tsaparas. "Link Analysis Ranking: Algorithms, Theory and Experiments." *ACM Transactions on Internet Technology (TOIT)* 5:1 (2005), 231–297.

[Botafogo et al. 92] R. A. Botafogo, E. Rivlin, and B. Shneiderman. "Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics." *ACM Transactions on Information Systems* 10:2 (1992), 142–180.

[Brandes and Cornelsen 03] U. Brandes and S. Cornelsen. "Visual Ranking of Link Structures." *Journal of Graph Algorithms and Applications* 7:2 (2003), 181–201.

[Brin and Page 98] Sergey Brin and Lawrence Page. "The Anatomy of a Large-Scale Hypertextual Web Search Engine." *Computer Networks and ISDN Systems* 33 (1998), 107–117.

[Broder et al. 97] A. Broder, S. Glassman, M. Manasse, and G. Zweig. "Syntactic Clustering of the Web." In *Proceedings of the Sixth International Conference on World Wide Web*, pp. 391–404. Amsterdam: Elsevier Science Publishers Ltd., 1997.

[Broder et al. 00] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. "Graph Structure in the Web." *Computer Networks (Ninth International World Wide Web Conference)* 33:1 (2000), 309–320.

[Broder et al. 04] Andrei Z. Broder, Ronny Lempel, Farzin Maghoul, and Jan Pedersen. "Efficient PageRank Approximation via Graph Aggregation." In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters*, pp. 484–485. New York: ACM Press, 2004.

[Carrière and Kazman 97] J. Carrière and R. Kazman. "WebQuery: Searching and Visualizing the Web through Connectivity." In *Selected Papers from the Sixth International Conference on World Wide Web*, pp. 1257–1267. Essex, UK: Elsevier Science Publishers Ltd., 1997.

[Chakrabarti et al. 98a] S. Chakrabarti, B. Dom, D. Gibson, S. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. "Spectral Filtering for Resource Discovery." ACM SIGR Workshop on Hypertext Information Retrieval for the Web, 1998.

[Chakrabarti et al. 98b] Soumen Chakrabarti, Byron Dom, Prabhakar Raghavan, Sridhar Rajagopalan, David Gibson, and Jon Kleinberg. "Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text." In *Proceedings of the Seventh International Conference on World Wide Web*, pp. 65–74. Amsterdam: Elsevier Science Publishers B. V., 1998.

[Chakrabarti et al. 99] S. Chakrabarti, M. van den Berg, and B. Dom. "Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery." In *Proceedings of the Eighth International Conference on World Wide Web*, pp. 1623–1640. New York: Elsevier North-Holland, Inc., 1999.

[Chang et al. 00] H. Chang, D. Cohn, and A. McCullum. "Learning to Create Customized Authority Lists." In *Proceedings of the 17th International Conference on Machine Learning*, pp. [27–134]. San Francisco, CA: Morgan Kaufmann, 2000.

[Chen et al. 02] Y. Chen, Q. Gan, and T. Suel. "I/O Efficient Techniques for Computing PageRank." Technical Report TR-CIS-2002-03, Polytechnic University, CIS Department, 2002.

[Chiang et al. 95] Y.-J. Chiang, M. T. Goodrich, E. F. Grove, R. Tamassia, D. E. Vengroff, and J. S. Vitter. "External-Memory Graph Algorithms." In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 139–149. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1995.

[Chien et al. 01] Steve Chien, Cynthia Dwork, and Ravi Kumar. "Towards Exploiting Link Evolution." Unpublished manuscript. Available from World Wide Web (http://citeseer.ist.psu.edu/chien0ltowards.html), 2001.

[Cho and Garcia-Molina 00] Junghoo Cho and Hector Garcia-Molina. "The Evolution of the Web and Implications for an Incremental Crawler." In *Proceedings of the Twenty-Sixth International Conference on Very Large Databases*, pp. 200–209. San Francisco, CA: Morgan Kaufmann, 2000.

[Cohn and Chang 00] D. Cohn and H. Chang. "Learning to Probabilistically Identify Authoritative Documents." In *Proceedings of the 17th International Conference on Machine Learning*, pp. 167–174. San Francisco, CA: Morgan Kaufmann, 2000.

[Davison 00] Brian D. Davison. "Recognizing Nepotistic Links on the Web." In *Artificial Intelligence for Web Search*, pp. 23–28. Menlo Park, CA: AAAI Press, 2000.

[Del Corso et al. 04] G. M. Del Corso, A. Gulli, and F. Romani. "Exploiting Web Matrix Permutations to Speedup PageRank Computation." Technical Report IIT TR-04, Istituto di Informatica e Telematica, 2004.

[Dill et al. 01] Stephen Dill, S. Ravi Kumar, Kevin S. McCurley, Sridhar Rajagopalan, D. Sivakumar, and Andrew Tomkins. "Self-Similarity in the Web." In *Proceedings of the Twenty-Seventh International Conference on Very Large Databases*, pp. 69–78. San Francisco, CA: Morgan Kaufmann, 2001.

[Ding et al. 01] C. Ding, X. He, P. Husbands, H. Zha, and H. Simon. "PageRank, HITS and a Unified Framework for Link Analysis." Technical Report TR 47847, Lawrence Berkeley National Laboratory, 2001.

[Dwork et al. 01] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. "Rank Aggregation Methods for the Web." In *Proceedings of the Tenth International Conference on World Wide Web*, pp. 613–622. New York: ACM Press, 2001.

[Eiron et al. 04] Nadav Eiron, Kevin McCurley, and John Tomlin. "Ranking the Web Frontier." In *Proceedings of the Thirteenth International Conference on World Wide Web*, pp. 309–318. New York: ACM Press, 2004.

[Fagin et al. 03] R. Fagin, R. Kumar, and D. Sivakumar. "Comparing Top k Lists." In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 28–36. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2003.

[Faloutsos et al. 99] M. Faloutsos, P. Faloutsos, and C. Faloutsos. "On Power-Law Relationships of the Internet Topology." In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp. 251–262. New York: ACM Press, 1999.

[Finkelstein et al. 02] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. "Placing Search in Context: The Concept Revisited." *ACM Transactions on Information Systems* 20:1 (2002), 116–131.

[Frisse 88] M. E. Frisse. "Searching for Information in a Hypertext Medical Handbook." *Commun. ACM* 31:7 (1988), 880–886.

[Geerts et al. 04] Floris Geerts, Heikki Mannila, and Evimaria Terzi. "Relational Link-Based Ranking." In *Proceedings of Thirtieth International Conference on Very Large Databases*, edited by M. A. Nascimento et al., pp. 552–563. San Francisco, CA: Morgan Kaufmann, 2004.

[Gibson et al. 98] David Gibson, Jon Kleinberg, and Prabhakar Raghavan. "Inferring Web Communities from Link Topology." In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia*, pp. 225–234. New York: ACM Press, 1998.

[Gleich et al. 04] David Gleich, Leonid Zhukov, and Pavel Berkhin. "Fast Parallel PageRank: A Linear System Approach." Technical Report YRL-2004-038, Yahoo!, 2004.

[Golub and Greif 04] Gene H. Golub and Chen Greif. "Arnoldi-Type Algorithms for Computing Stationary Distribution Vectors, with Application to PageRank." Technical Report SCCM-04-15, Stanford University, 2004.

[Golub and Loan 96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*, Third edition, Johns Hopkins Studies in Mathematical Sciences. Baltimore, MD: Johns Hopkins University Press, 1996.

[Guillaume et al. 02] J-L. Guillaume, M. Latapy, and L. Viennot. "Efficient and Simple Encodings for the Web Graph." In *Proceedings of the Eleventh International Conference on World Wide Web*, *Alternate Poster Tracks*. Available from World Wide Web (http://www2002.org/CDROM/poster/125.pdf), 2002.

[Gyöngyi et al. 04a] Zoltán Gyöngyi, Pavel Berkhin, Hector Garcia-Molina, and Jan Pedersen. "Web Spam Detection Based on Mass Estimation." Technical Report, Stanford University, 2004.

[Gyöngyi et al. 04b] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. "Combating Web spam with TrustRank." In *Proceedings of the Thirtieth International Conference on Very Large Databases*, edited by M. A. Nascimento et al., pp. 576–587. San Francisco, CA: Morgan Kaufmann, 2004.

[Haveliwala 99] Taher Haveliwala. "Efficient Computation of PageRank." Technical Report, Stanford University, 1999.

[Haveliwala 02a] Taher Haveliwala. "Efficient Encodings for Document Ranking Vectors." Technical Report, Stanford University, 2002.

[Haveliwala 02b] Taher Haveliwala. "Topic-Sensitive PageRank." In *Proceedings of the Eleventh International Conference on World Wide Web*, pp. 517–526. New York: ACM Press, 2002.

[Haveliwala and Kamvar 03] Taher Haveliwala and Sepandar Kamvar. "The Second Eigenvalue of the Google Matrix." Technical Report, Stanford University, 2003.

[Haveliwala et al. 03a] T. Haveliwala, S. Kamvar, and G. Jeh. "An Analytical Comparison of Approaches to Personalizing PageRank." Technical Report, Stanford University, 2003.

[Haveliwala et al. 03b] Taher Haveliwala, Sepandar Kamvar, Dan Klein, Chris Manning, and Gene Golub. "Computing PageRank using Power Extrapolation." Technical Report, Stanford University, 2003.

[Henzinger 00] M. Henzinger. "Link Analysis in Web Information Retrieval." *Bulletin of the IEEE Comp. Sosc. Tech. Committee on Data Engineering* 23:3 (2000), 3–8.

[Henzinger 03] M. R. Henzinger. "Algorithmic Challenges in Web Search Engines." *Internet Mathematics* 1:1 (2003), 115–126.

[Hofmann 99] Thomas Hofmann. "Probabilistic Latent Semantic Analysis." In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, edited by K. Laskey and H. Prude, pp. 289–296. San Francisco, CA: Morgan Kaufmann, 1999.

[IBM 99] IBM Almaden Research Center Computer Science Principles and Methodologies Group. "CLEVER Searching." Available from World Wide Web (http://www.almaden.ibm.com/cs/k53/clever.html), 1999.

[Ipsen and Kirkland 04] I. C. F. Ipsen and Steve Kirkland. "Convergence Analysis of an Improved PageRank Algorithm." Technical Report, North Carolina State University, 2004.

[Jaynes 57] E. Jaynes. "Information Theory and Statistical Mechanics." *Physical Review* 106 (1957), 620–630.

[Jeh and Widom 02a] G. Jeh and J. Widom. "SimRank: A Measure of Structural-Context Similarity." In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 538–543. New York: ACM Press, 2002.

[Jeh and Widom 02b] G. Jeh and J. Widom. "Scaling Personalized Web Search." Technical Report, Stanford University, 2002.

[Kamvar et al. 03a] Sepandar Kamvar, Taher Haveliwala, and Gene Golub. "Adaptive Methods for the Computation of PageRank." Technical Report, Stanford University, 2003.

[Kamvar et al. 03b] Sepandar Kamvar, Taher Haveliwala, Christopher Manning, and Gene Golub. "Exploiting the Block Structure of the Web for Computing PageRank." Technical Report, Stanford University, 2003.

[Kamvar et al. 03c] Sepandar Kamvar, Taher Haveliwala, Christopher Manning, and Gene Golub. "Extrapolation Methods for Accelerating the Computation of PageRank." In *Proceedings of the Twelfth International Conference on World Wide Web*, pp. 261–270. New York: ACM Press, 2003.

[Kamvar et al. 03d] Sepandar Kamvar, Mario Schlosser, and Hector Garcia-Molina. "The EigenTrust Algorithm for Reputation Management in P2P Networks." In *Proceedings of the Twelfth International Conference on World Wide Web*, pp. 640–651. New York: ACM Press, 2003.

[Katz 53] L. Katz. "A New Status Index Derived from Sociometric Analysis." *Psychmetrika* 18:1 (1953), 39–43.

[Kleinberg 99] Jon Kleinberg. "Authoritative Sources in a Hyperlinked Environment." *Journal of the ACM* 46:5 (1999), 604–632.

[Kumar et al. 99] S. R. Kumar, P. Raphavan, S. Rajagopalan, and A. Tomkins. "Trawling the Web for Emerging Cyber Communities." In *Proceedings of the Eighth International Conference on World Wide Web*, pp. 1481–1493. New York: Elsevier North-Holland, Inc., 1999.

[Kumar et al. 00] R. Kumar, F. Maghoul, P. Raghavan, and R. Stata. "Graph Structure in the Web." In *Proceedings of the Ninth International Conference on World Wide Web*, pp. 247–256. Amsterdam: North-Holland Publishing Co., 2000.

[Langville and Meyer 04a] Amy Langville and Carl Meyer. "Deeper Inside PageRank." *Internet Mathematics* 1:3 (2004), 335–380.

[Langville and Meyer 04b] Amy N. Langville and Carl D. Meyer. "Updating PageRank with Iterative Aggregation." In *Proceedings of the Thirteenth International World Wide Web Conference on Alternate Track Papers & Posters*, pp. 392–393. New York: ACM Press, 2004.

[Langville and Meyer 05] Amy N. Langville and Carl D. Meyer. "A Survey of Eigenvector Methods of Web Information Retrieval." *SIAM Review* 47:1 (2005), 135–161.

[Larson 96] R. R. Larson. "Bibliometrics of the World Wide Web: An Exploratory Analysis of the Intellectual Structures Of Cyberspace." In *ASIS '96: Proceedings of the 59th ASIS Anual Meeting*, edited by S. Hardin, pp. [71–78]. Medford, NJ: Information Today, 1996.

[Lee and Borodin 96] Hyun Chul Lee and Allan Borodin. "Perturbation of the Hyper-Linked Environment." In *Computing and Combinatorics, 9th Annual International Conference*, Lecture Notes in Computer Science 2697, edited by T. Warnow and B. Zhu, pp. 272–283. New York: Springer, 1996.

[Lee et al. 03] Chris Pan-Chi Lee, Gene H. Golub, and Stefanos A. Zenios. "A Fast Two-Stage Algorithm for Computing PageRank and Its Extensions." Technical Report SCCM-03-15, Stanford University, 2003.

[Lempel and Moran 00] R. Lempel and S. Moran. "The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect." In *Proceedings of the Ninth International Conference on World Wide Web*, pp. 387–401. Amsterdam: North-Holland Publishing Co., 2000.

[Lempel and Moran 01] Ronny Lempel and Shlomo Moran. "SALSA: The Stochastic Approach for Link-Structure Analysis." *ACM Transactions on Information Systems* 19:2 (2001), 131–160.

[Lempel and Moran 03] Ronny Lempel and Shlomo Moran. "Rank-Stability and Rank-Similarity of Link-Based Web Ranking Algorithms in Authority Connected Graphs." *Second Workshop on Algorithms and Models for the Web-Graph*. Budapest, Hungary, 2003.

[Lovász 96] László Lovász. "Random Walks on Graphs: A Survey." In *Combinatorics: Paul Erdős is Eighty*, Bolyai Society Mathematical Studies 2, edited by D. Miklós, V. T. Sos, and T. Szonyin, pp. 353–348. Budapest: János Bolyai Mathematical Society, 1996.

[Manaskasemsak and Rungsawang 04] B. Manaskasemsak and A. Rungsawang. "Parallel PageRank Computation on a Gigabit PC Cluster." In *Proceedings of the 18th International Conference on Advanced Information Networking and Applications*, pp. 273–277. Los Alamitos, CA: IEEE Computer Society, 2004.

[Marchiori 97] M. Marchiori. "The Quest for Correct Information on the Web: Hyper Search Engines." In *Selected Papers from the Sixth International Conference on World Wide Web*, pp. 1225–1235. Essex, UK: Elsevier Science Publishers Ltd., 1997.

[Mitzenmacher 03] Michael Mitzenmacher. "A Brief History of Generative Models for Power Law and Lognormal Distributions." *Internet Mathematics* 1:2 (2003), 226–251.

[Motwani and Raghavan 95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. New York: Cambridge University Press, 1995.

[Ng et al. 01a] A. Y. Ng, A. X. Zheng, and M. I. Jordan. "Stable Algorithms for Link Analysis." In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 258–266. New York: ACM Press, 2001.

[Ng et al. 01b] Andrew Ng, Alice Zheng, and Michael Jordan. "Link Analysis, Eigenvectors and Stability." In *Proceedings of the Seventeenth International Joint Conferences on Artificial Intelligence*, edited by B. Nebel, pp. 903–910. San Francisco, CA: Morgan Kaufmann, 2001.

[Page et al. 98] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. "The PageRank Citation Ranking: Bringing Order to the Web." Technical Report, Stanford University, 1998.

[Pandurangan et al. 02] G. Pandurangan, P. Raghavan, and E. Upfal. "Using PageRank to Characterize Web Structure." In *Computing and Combinatorics, 8th Annual International Conference*, Lecture Notes in Computer Science 2387, edited by O. H. Ibarra and L. Zhang, pp. 330–339. New York: Springer, 2002.

[Pirolli et al. 96] P. Pirolli, J. Pitkow, and R. Rao. "Silk from a Sow's Ear: Extracting Usable Structures from the Web." In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Common Ground*, pp. 118–125. New York: ACM Press, 1996.

[Rafiei and Mendelzon 00] D. Rafiei and A.O. Mendelzon. "What is this Page Known for? Computing Web Page Reputations." In *Proceedings of the Ninth International Conference on World Wide Web*, pp. 823–835. Amsterdam: North-Holland Publishing Co., 2000.

[Randall et al. 02] K. Randall, R. Stata, R. Wickremesinghe, and J. Wiener. "The Link Database: Fast Access to Graphs of the Web." In *Proceedings of the IEEE Data Compression Conference (DCC '02)*, pp. 122–131. Los Alamitos, CA: IEEE Computer Society, 2002.

[Richardson and Domingos 02] Mathew Richardson and Pedro Domingos. "The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank." In *Proceedings of the 2001 Neural Information Processing Systems (NIPS) Conference*, Advances in Neural Information Processing Systems 14, edited by T. G. Dietterich, S. Becker, and Z. Ghahramani, pp. 1441–1448. Cambridge, MA: MIT Press, 2002.

[Stewart 99] William J. Stewart. "Numerical Methods for Computing Stationary Distribution of Finite Irreducible Markov Chains." In *Advances in Computational Probability*, edited by Winfried Grassmann, Chapter 4. Dordrecht: Kluwer Academic Publishers, 1999.

[Suel and Yuan 01] T. Suel and J. Yuan. "Compressing the Graph Structure of the Web." In *Proceedings of the IEEE Data Compression Conference (DCC '01)*, pp. 213–222. Los Alamitos, CA: IEEE Computer Society, 2001.

[Tomlin 03] John Tomlin. "A New Paradigm for Ranking Pages on the World Wide Web." In *Proceedings of the Twelfth International Conference on World Wide Web*, pp. 350–355. New York: ACM Press, 2003.

[Toutanova et al. 04] Kristina Toutanova, Christopher Manning, and Andrew Y. Ng. "Learning Random Walk Models for Inducing Word Dependency Distributions." In *Twenty-First International Conference on Machine Learning*, Article No. 103, ACM International Conference Proceedings Series. New York: ACM Press, 2004.

[Tsoi et al. 03] A. C. Tsoi, G. Morini, F. Scarselli, M. Hagenbuchner, and M. Maggini. "Adaptive Ranking of Web Pages." In *Proceedings of the Twelfth International Conference on World Wide Web*, pp. 356–365. New York: ACM Press, 2003.

[Wasserman and Faust 94] Stanley Wasserman and Katherine Faust. *Social Networks. Methods and Applications.* Cambridge, UK: Cambridge University Press, 1994.

[Witten et al. 99] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images.* San Francisco, CA: Morgan Kaufmann, 1999.

[Xi et al. 04] Wensi Xi, Benyu Zhang, Yizhou Lu, Zheng Chen, Shuicheng Yan, Huajun Zeng, Wei-Ying Ma, and Edward A. Fox. "Link Fusion: A Unified Link Analysis Framework for Multi-Type Interrelated Data Objects." In *Proceedings of the Thirteenth International Conference on World Wide Web*, pp. 319–327. New York: ACM Press, 2004.

Pavel Berkhin, Yahoo!, 701 First Avenue, Sunnyvale, CA 94089
(pberkhin@yahoo-inc.com)

Received May 17, 2004; accepted December 4, 2004.