

## Nelly Litvak

Faculteit Electrotechniek, Wiskunde en Informatica  
Afdeling Toegepaste Wiskunde  
Universiteit Twente, Postbus 217, 7500 AE Enschede  
n.litvak@ewi.utwente.nl

### Onderzoek

# Googling Maths

It is common belief that two arbitrary persons are linked by a chain of at most six acquaintances. This idea was coined in 1929 by the Hungarian writer Frigyes Karinthy in a short story called *Chains*. Later it made it into a romantic play, and a movie called ‘*Six Degrees of Separation*’, both by playwright John Guare. The World Wide Web has a far more complicated structure, and contains many more web pages, than there are humans on the planet. What can be said about its structure and its connectivity? Nelly Litvak, assistant professor in the field of Stochastic Operations Research at the Universiteit Twente, gives an account of what is currently known about the World Wide Web and its search engines.

In our informational society, the *World Wide Web* has quickly become one of the most important media. Given the gigantic size of the Web and its uncontrollable random expansion, the structure of the Web may seem completely chaotic, and the high performance of modern search engines looks almost like a magic. This note addresses two topics. First, we attempt to highlight some well-known structural properties of the World Wide Web and show how they can be modeled mathematically. Second, we will explain the principal scheme of a Web search engine and discuss important ranking algorithms used for listing the search results in an appropriate order.

The common viewpoint in the literature is to present the Web as a *graph*, with the web pages regarded as vertices and the links as *directed* edges. This simplified representation suffices to answer many important questions such as: What is a typical number of in- and out-going links? Does the Web consist of one giant knot of pages and links (the graph is

connected) or is it more like several separate ‘islands’? What is the average path length between two connected pages? These extremely important questions have been partly answered in the famous paper by Broder et al. [7] that is discussed later in this article.

Several typical properties of the Web can also be observed in other complex stochastic networks such as network of collaborations, airline routes networks, biological networks, scientific citations, children’s friendships, and many others [12]. This suggests that a network structure builds up in a certain way, which is similar for various large systems. Understanding how this structure appears enables one to predict the developments in a highly dynamic environment such as the World Wide Web. Currently, *growing network* models with *preferential attachment* are widely accepted as a possible mathematical explanation for many empirically discovered properties of complex networks. The main idea in these models is that the observed structure is a result of a

network growth driven by a ‘rich get richer’ mechanism. That is, a newly created node is more likely to link to nodes that are already well-connected. We will address the growing network models in more detail in this paper.

For a user, the practical availability of the enormous amount of information offered by the Web depends greatly on the efficiency of search engines. At the end of this paper we will briefly explain how a search engine works and focus on the hyperlink-based techniques used for listing the search results in a user-friendly order. In particular, we will explain the relatively simple mathematical model behind the *Google PageRank*.

#### Graph Structure in the Web

As mentioned before, we view the Web as a set of vertices (pages, nodes), and directed edges (links) between them. We say that there is an edge from page  $i$  to page  $j$  if  $i$  has a hyperlink to  $j$ , i.e., a user can go from  $i$  to  $j$  by just one click.

Given the spontaneous chaotic development of the Web, one can hardly expect any regularity in its structure. From the first sight, it looks like the Web graph may have any shape you please. However, the fundamental research by Broder et al. [7] revealed several robust structural properties of the Web. The experiments in [7] were carried out on two large crawls each containing about 200 million pages and 1.5 billion links. To indi-

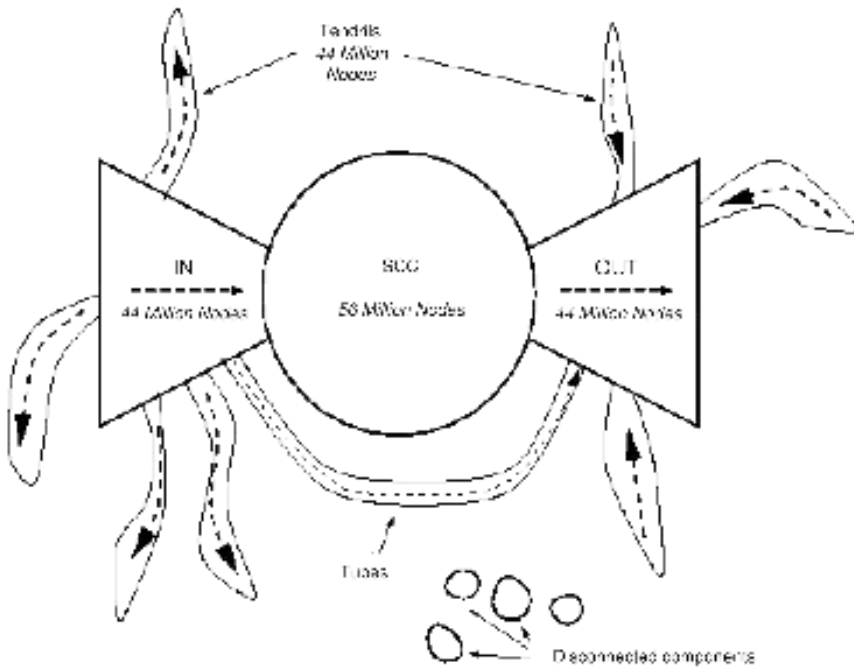


Figure 1 Graph Structure of the Web (from [7])

cate the importance of the Broder et al. paper, we just mention that it was cited in about 500 articles! Strikingly, most typical traits discovered in the World Wide Web have also been observed in other complex networks such as social networks, networks of scientific citations, biological networks, etc.

In Figure 1 we depict the structure of the Web as discovered in [7]. We see that the majority of the pages are united in one connected component, which has a shape of a ‘bow tie’. For any two pages  $i$  and  $j$  in the ‘bow tie’, there is a hyperlink path either from  $i$  to  $j$  or from  $j$  to  $i$ . In the middle, there is a *Strongly Connected Component (SCC)* containing more than one quarter of all pages. SCC as used in graph theory stands for a set of nodes where each node can be reached from any other node by traversing directed edges. For the Web, it means that in the SCC, each page can be reached from any other page by clicking on hyperlinks. Next, there are large IN and OUT components. The pages in IN (OUT) have a path to (from) the SCC, but not back. There are also smaller groups such as *Tendrils* branching from IN or leading directly to OUT, and *Tubes* offering a path from IN to OUT. The little ‘islands’ represent the *Disconnected components*, which amount to less than 10% of the Web.

From the above, one may have the impression that the Web is greatly connected, and that for two random pages  $i$  and  $j$ , a hyperlink

path from  $i$  to  $j$  is likely to exist. However, a closer look suggests that this is not the case. Roughly speaking, a hyperlink path exists only if page  $i$  belongs to IN+SCC, and page  $j$  is in SCC+OUT. As both IN and OUT contain slightly less than 1/4 of all pages, the probability that the path exists is (only!) about 24%.

Assuming that a path from one page to another exists, one may ask what the average path length is? Despite the enormous size of the Web, the average path turns out to be relatively short. Experiments in [7] report about 16 clicks only! Moreover, if links can be traversed in both ways, the average path length reduces to the value of about 7.

This phenomenon — a short average distance between the nodes in large networks — has been known for a long time as a *small-*

*world effect*. One of the most famous experiments in this respect was carried out by Stanley Milgram in the sixties in a context of social networks. The participants were asked to pass a letter to their first-hand acquaintances so that it would finally reach the assigned targets individual. About 1/4 of the letters reached the target passing, on average, through the hands of only about 6 people! [12]

While looking quite astonishing, the small-world effect actually has a simple mathematical explanation. Here is a greatly simplified argument, which is far from being rigorous but helps to grasp the main idea. For a given node, assume that the number of nodes within a distance  $r$  is roughly  $a^r$ , where  $a > 0$  is a constant. This assumption is true for many real-life networks. Now, let  $l$  be the maximal distance from one node to another. Then it follows from geometric series that the total number of nodes  $N$  is

$$N = 1 + a + a^2 + \dots + a^l = (a^{l+1} - 1)/(a - 1) \approx a^{l+1}/(a - 1).$$

Hence, for large  $N$ , the value  $l$  turns out to be of the order  $\log(N)$ . For example, in a network of one billion nodes, this number is of the order 10, which is exactly the small-world effect observed in experiments. For more detail on the small-world effect, we refer to the brilliant survey by Mark Newman [12] and references therein.

**Power laws**

Let us consider the number of in- and outgoing links, called, respectively, *in-degree* and *out-degree*, of a Web page. The question is, for instance, what is the fraction  $p_k$  of pages whose in-degree is exactly  $k$ ? The experiments carried out in different times on different crawl sizes agree that  $p_k$  is approximately proportional to  $k^{-2.1}$ . In Figure 2 (left side), we present the experimental results on the in-

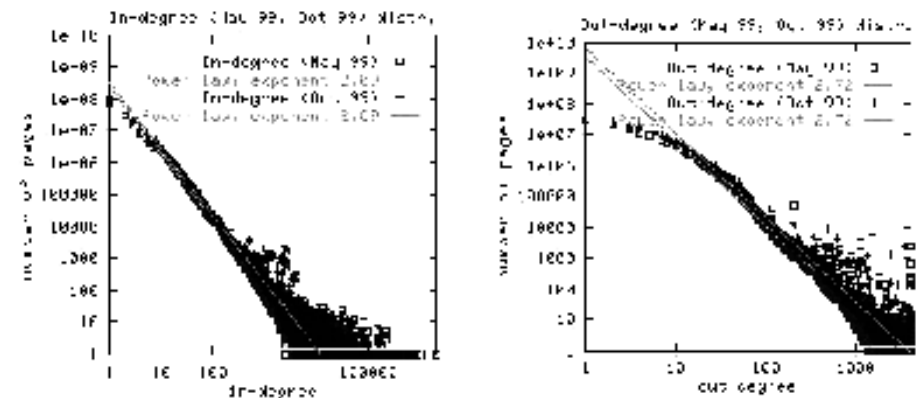


Figure 2 In- and out-degree (May99, Oct99) in the log-log scale (from [7])

degree distribution [7].

Plotted in the log-log scale, for each in-degree  $k$ , the values of  $p_k$  concentrate around a straight line:  $\log(p_k) = -2.09 \log(k) + \text{const}$ , which signals the power law:  $p_k \approx \text{const} \cdot k^{-2.09}$ .

Put in words, the power law means that the majority of the pages have a relatively small in-degree but there is a noticeable group of pages whose in-degree is high. To see that, let us evaluate the number of pages with in-degree 1000. According to the power law distribution, the fraction of such pages is of the order  $1000^{-2.1} \approx 10^{-6}$ . Hence, for a net of one billion pages, the number of pages whose in-degree is one thousand is of the order  $10^9 \cdot 10^{-6} = 1000$ . A group of this magnitude can not be neglected in any reasonable network analysis.

To demonstrate a difference with, for instance, the exponential law, assume that  $p_k$  is of the order  $10^{-k}$ . Then for  $k = 1000$ , the proportion  $10^{-1000}$  results in a negligible group of pages, even in a truly giant network. As we know that *there are* well-connected nodes (think for instance of the homepage of Google or CNN), the power law seems to be a more realistic model for the in-degree distribution. As we see in the right plot in Figure 2, the out-degree also obeys a power law but it has an exponent of about  $-2.7$ . Thus, for large  $k$ , the probability of in-degree  $k$  is larger than the probability of out-degree  $k$ .

Clearly, the Web can be subdivided into large logically united components, for instance, by domain or by topic. Surprisingly, it turns out that such large components have a structure similar to the Web as a whole, which is a result of many essentially independent stochastic processes evolving in the Web at various scales.

This phenomenon, called *self-similarity*, was observed in a number of experiments on different crawl sizes, and analyzed in detail in [8]. There it was shown that a Web-like structure is present in so-called *thematically unified clusters*, i.e. sets of pages that share some common feature, for instance, content, domain, or geographical location.

The authors also note that in a purely random set of pages the structure will be lost. Indeed, assume that a sample of one million pages out of possible one billion is chosen at random. Then the probability that both ends of some edge belong to the chosen sample is  $(10^6/10^9) \cdot (10^6/10^9) = 10^{-6}$ . Since the average number of links per page is just about 8, we get on average  $8 \cdot 10^9 \cdot 10^{-6} = 8000$  links in the random collection of one million

nodes. With such a small amount of links one can hardly expect any interesting graph-theoretic structure.

**Mathematical models of the Web: preferential attachment**

Currently, growing network models with preferential attachment are widely accepted as a possible mathematical explanation of many empirically discovered properties of the Web. In these models, a newly created page is more likely to link to pages that are already well-connected. The most famous model of this sort was suggested in 1999 by Barabasi and Albert [3], and many modifications have appeared since then. We will closely follow Newman [12] in explaining how the model works and why it leads to the power law in-degree distribution.

In [3], a networks starts with one node. When a new node appears, it has  $m \geq 1$  *undirected*, or, equivalently, *bi-directed* links to distribute among the existing nodes. In doing so, a node follows the ‘rich get richer’ strategy, meaning that the probability that some node  $v$  gets a new link is proportional to the current in-degree of  $v$ . Thus, if the fraction of nodes with in-degree  $k$  is  $p_k$ , then the probability that a new link goes to this group is

$$\frac{k p_k}{1 \cdot p_1 + 2 \cdot p_2 + 3 \cdot p_3 + \dots + k p_k + \dots} = \frac{k p_k}{2m}, \quad k \geq 1.$$

The denominator on the left-hand side is first defined so that the sum of the probabilities equals 1; we then notice that it equals the average number of links per page, which is  $2m$  since each node brings  $m$  bi-directed edges.

Now, with each new node, the group of in-degree  $k$  receives on average  $m \cdot k p_k / (2m) = (1/2)k p_k$  links, which is independent of  $m$ . Thus, the number of vertices with in-degree  $k$  *decreases* by this amount since these nodes join the group of in-degree  $k + 1$ . On the other hand, on average  $(1/2)(k - 1)p_{k-1}$  nodes of in-degree  $k - 1$  will also receive a new link, so the number of vertices with in-degree  $k$  will *increase* by this number. If the total number of nodes is very large then the proportion of nodes with in-degree  $k$  almost does not change (in fact, this proportion converges to a constant when the number of nodes goes to infinity). So, when the  $n$ th new node is added and  $n$  is large enough, the number of nodes with in-degree  $k$  changes approximately by  $n p_k - (n - 1) p_k = p_k$ . Equating the incre-

ments in the number of nodes with in-degree  $k$ , we can write so-called master equations, which hold when the number of vertices in the graph goes to infinity:

$$p_k = \begin{cases} \frac{1}{2}(k - 1)p_{k-1} - \frac{1}{2}k p_k, & \text{for } k > m \\ 1 - \frac{1}{2}m p_m, & \text{for } k = m \end{cases}$$

Here the last equation reflects that there is always one new node with exactly  $m$  links, and at the same time the group of such nodes decreases by  $(1/2)m p_m$ , as happens for any other value of  $k$ . Writing the equation for  $p_m$  we get  $p_m = 2/(m + 2)$ , and for  $k \neq m$  we obtain  $p_k = p_{k-1} \cdot (k - 1)/(k + 2)$ . Recursively, we arrive at

$$p_k = \frac{(k - 1)(k - 2) \dots m}{(k + 2)(k + 1) \dots (m + 3)} p_m = \frac{2m}{(k + 2)(k + 1)k}.$$

Thus, for large  $k$ , we have  $p_k \sim k^{-3}$ , which is a power law with exponent 3. We note that the present model deviates from the experimental results suggesting  $p_k \sim k^{-2.1}$ . However, this deviation was resolved in later generalizations by other authors.

The significance of the Barabasi and Albert model is that besides modeling the growing random graph that exhibits the power law in-degree distribution, it also aims to explain *why* such distribution appears. We note that this model is in the spirit of the earlier model developed in 1965 by Derek de Solla Price in his work on scientific citations. Even before that in the 1950s, Herbert Simon showed that the power law distributions arise from the ‘rich get richer’ mechanism, also referred to as the *Matthew effect* [12] in sociology. As an example, we note that something like the power law can be observed in a group of school children: a few boys and girls are very popular while others have only one or two friends. Is it not natural to explain this by the tendency of the children to make friends with popular, or ‘well-connected’ classmates?

The models with preferential attachment have received much attention in the network literature. There is a lot of research on generalizing these models in such a way that they better reflect complicated features of the Web such as directed links, the hierarchical structure, appearing and disappearing of links and even the willingness of users to link to highly ranked pages [16]. The other research direction concerns a rigorous mathematical

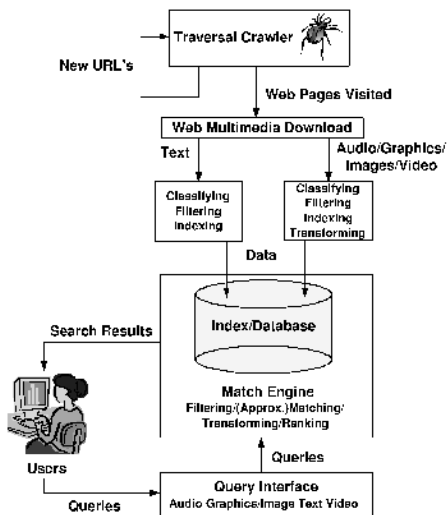


Figure 3 General scheme of a search engine

analysis of (generalized) preferential attachment models, based on the theory of random graphs. In particular, the power law distributions were rigorously derived, and it was also shown that the models with preferential attachment exhibit a small-world effect [4–5]. The ‘rich get richer’ mechanism appears to be responsible for many typical developments in complex networks.

### Search Engines

Studying and modeling Web structures is one of the main challenges for designing Web search engines [9], which are of extreme importance for navigating the Web. Here, we briefly discuss the working of a search engine, and consider two prominent hyperlink-based ranking techniques (in particular, the Google PageRank) for selecting important and interesting Web pages. The general scheme of a so-called crawler-based search engine is presented in the above figure.

Using this scheme we highlight essential components of the Web search, using a nice paper by Gallianno Cosme published in Search Engine Guide in May 2005 ([www.searchengineguide.com](http://www.searchengineguide.com)). According to that paper, a crawler-based search engine consists of three main parts: the spider (also known as crawler or robot), the index, and the software.

The *spider* is a program that visits pages and follows hyperlinks to move from one page to another. The main goal is to obtain the most recent copy of each page but, as we will see later, it may be also important for recording the hyperlink structure. The spiders start their journey from the pages that are already in the search engine database. The most active spiders on the Web are Googlebot

(Google), Slurp (Yahoo!) and MSNBot (MSN Search).

The copies of crawled pages are stored in the *index*, which is essentially a giant catalogue or database where the pages are classified, filtered, indexed, transformed (if needed) and grouped according to some rules, for instance, according to the topic. The size of the index is truly enormous. Just to give an idea, the latest figure revealed by Google is 8 billion pages! In practice, both crawling and indexing require significant computing time and capacity.

When a user inserts a query in a query interface (like a famous white page of Google), the search engine finds relevant pages in its database using software based on sophisticated algorithms and state-of-art information retrieval techniques. In Figure 3 this software tool is denoted as Match Engine. It is beyond the scope of this paper to discuss the methods for retrieving relevant pages from the database. We observe only that, in one way or another, the query is compared with the key-words related to a Web page. Naturally, these are mainly the keywords and the text included in the page itself. However, curiously enough, the text of the hyperlinks connecting to a page is also taken into account, at least by Google. That is why the query ‘*miserable failure*’ returns a biography of G.W. Bush although these words have never been mentioned in this document! The relevant pages are then ranked in some order that the search engine finds most appropriate, and the results are presented to the user. The ranking algorithm is a well-kept secret, and it depends on many factors, for instance, on geographical location of the user and maybe even on his/her last searches.

We would like to emphasize that the main structural feature of a search engine is that crawling and indexing happen without involving the user, who ‘only’ needs to consult a database and receive the results from the *index* rather than from the Web itself. Search engines are equipped with modern software and powerful processors that scan the index extremely fast so that the search results appear on a screen almost instantaneously. As a minor drawback of this scheme, the user may access only the pages listed in the index, which is not complete and not entirely up-to-date. For instance, new pages that have been crawled but have not yet been added to the index, will *not* be available to those searching with the search engine. This is the reason why Web site owners have to make sure that their sites are timely indexed and highly

ranked. In business, there is a whole branch of marketing called Search Engine Optimization that develops techniques for increasing the Web site’s ranking performance.

In the end, the main goal of any search engine is to satisfy the user, so we may trust that the vastly expanding index is frequently updated, and matching/ranking algorithms are steadily improving to provide us with the desirable results. Note, by the way, that the index and the matching/ranking mechanisms are different for different search engines, and therefore it is not uncommon to use several search engines for the same query.

### Node ranking based on the hyperlink structure

We claimed that the knowledge of the hyperlink structure is important for search engines. Obviously, such knowledge helps to optimize the spider’s crawl, and it can be used for matching as well. However, some search engines, and particularly Google, also use hyperlinks for ranking the Web pages according to their importance.

Suppose, a search engine has received a query and found relevant pages in its index. Then another problem arises. Namely, thousands of pages may match the query, leaving the question how to define the *most important* page. In the beginning, this problem was solved solely by finding pages with best-matching text. However, with fast expansion of the Web, such methods soon became inefficient. Two innovative path breaking approaches were presented in 1998: one belongs to a well-known academician Jon Kleinberg [11], and another came from two PhD students from Stanford, Sergey Brin and Larry Page [6] known as the ‘founding fathers’ of Google. Although the two approaches are different, the main idea is similar: the page should be ranked high and listed high if many other good pages have a hyperlink to this page, and thus the page is recognized by the Web community as an important source of information. In contrast to the previously used methods, these novel ranking techniques are based not on the content of the pages but on the most fundamental feature of the Web – the hyperlink structure. Naturally, both methods require the knowledge of who is linking to whom. This information is recorded in the *adjacency matrix*  $A$  defined as follows:

$$A_{ij} = \begin{cases} 1, & \text{there is a link from } i \text{ to } j, \\ 0, & \text{otherwise.} \end{cases}$$

Such a matrix can be obtained by the spider while crawling the Web.

In his work [11], Kleinberg considers two sorts of pages: hubs and authorities. A hub serves as a reference giving many links to important authorities. The authorities, on the other hand, contain important information and thus receive many links from the hubs. Formally, let  $a_i$  and  $h_i$  be respectively the authority and the hub score of page  $i = 1, \dots, n$ . Then

$$a_i = \sum_j A_{ji} h_j, \quad h_i = \sum_j A_{ij} a_j.$$

The HITS algorithm suggested by Kleinberg, is as follows:

1. Retrieve a set of relevant pages from the database.
2. Extend this set by adding all pages that have links to and from the selected pages.
3. For the extended set, compute the hub and authority scores.
4. Since the user is mostly interested in authoritative sources, list the search results according to the authority score.

If we want to include this ranking algorithm into the scheme of Figure 3, we should add an *Authority Score Computation* block between the Database and the User (this option is depicted in Figure 4 with dashed arrows).

Fortunately, the authority scores can be computed very fast because the number of pages involved in the computations is not very large, which results in a well solvable linear algebra problem.

The approach of Brin and Page is different. In their work [6], they introduce a universal popularity measure – the PageRank. The PageRank  $PR(i)$  of page  $i$  depends on how many other pages link to  $i$  and how important these pages are. The original formula is as follows:

$$PR(i) = c \sum_j \frac{A_{ji}}{d_j} PR(j) + (1 - c), \quad (1)$$

where  $i = 1 \dots n$ ,  $d_j$  is the number of outgoing links from page  $j$ ,  $n$  is the number of pages in the Web, and  $c$  is a constant between zero and one (Google originally used  $c = 0.85$ ). Brin and Page’s algorithm works as follows:

1. Right after crawling the Web, retrieve and store the matrix  $A$ .
2. Compute the PageRank score for each page and store the PageRank vector.
3. For each query, list the matching pages ac-

ording to their PageRank. In order to reflect this procedure in Figure 3, we have to add a chain that is depicted in Figure 4 by solid arrows: there is a large computation block right after crawling but there is no computation involved after consulting the database, which in general helps to deliver search results faster.

Let us now take a closer look at the famous PageRank formula (1). We see that two factors are taken into account: the quality and the quantity of incoming links. The idea is that if we view a link as a vote, then pages with many links deserve attention. Moreover, if a page has only a few links but these links come from important sites, then this page is also worth browsing.

PageRank has an insightful probabilistic interpretation. The  $PR(i)$  in (1) can be normalized so that they sum up to one. We denote the normalized PageRank values by  $\pi_i$ ’s:

$$\pi_i = \frac{PR(i)}{PR(1) + PR(2) + \dots + PR(n)}, \quad (2)$$

for  $i = 1, \dots, n$ . The vector  $\pi = (\pi_1, \pi_2, \dots, \pi_n)$  is a probability distribution that can be interpreted via the so-called *easily bored surfer* model. Consider a random surfer who starts navigating the Web from a random page. At each page, with probability  $c$ , the surfer follows a randomly chosen hyperlink, and with probability  $1 - c$  he gets ‘bored’ and jumps to a random page. To keep the model equivalent to (1), we have to make a natural assumption that the user always jumps to a random page when reaching a page which does not have out-going links. Such pages, called *dangling nodes*, should not influence the ranking.

The described surfing process can be modeled as an *irreducible Markov chain* [10], since there is a possibility to make a random jump, and thus, any two pages (states) can be reached from each other. Hence, it follows from the theory of

Markov chains that  $\pi_i$  is nothing else but the long-run probability, or long-run fraction of time, that a random surfer spends on page  $i$ . Moreover, this probability is uniquely defined for all  $i = 1, \dots, n$ . Naturally, the higher the probability, the more popular the page is. It follows from the description of the surfing process that all  $\pi_i$  satisfy

$$\pi_i = c \sum_j \frac{A_{ji}}{d_j} \pi_j + c \frac{1}{n} \sum_{j \in D} \pi_j + \frac{1 - c}{n},$$

$$\sum_i \pi_i = 1,$$

with  $i = 1 \dots n$  and  $D$  a set of dangling nodes. The last equation is equivalent to (1) and (2).

The ‘dumping factor’  $c < 1$  is needed in particular because the random jump option guarantees that the unique distribution  $\pi$  exists. With  $c = 1$ , it is quite likely that some pages can not be reached from each other, and then according to the Markov chain theory, the PageRank vector is not well defined.

The PageRank citation ranking technique is very efficient and is actually used by Google, although maybe not in its original form. The disadvantage of this method is however obvious. Equation (1) must hold for each  $i = 1, \dots, n$ , with  $n$  the number of pages in the index. This means that we have a huge linear system with  $n$  equations and  $n$  variables, where  $n$  is of the order of billions. Solving such linear system directly is practically unfeasible. Google originally proposed to use a *power iteration method* that works as follows. First, put  $\pi^{(0)} = (1/n, \dots, 1/n)$ . Then for each  $k \geq 1$  compute

$$\pi^{(k)} = c \sum_j \frac{A_{ji}}{d_j} \pi_j^{(k-1)} + c \frac{1}{n} \sum_{j \in D} \pi_j^{(k-1)} + \frac{1 - c}{n}.$$

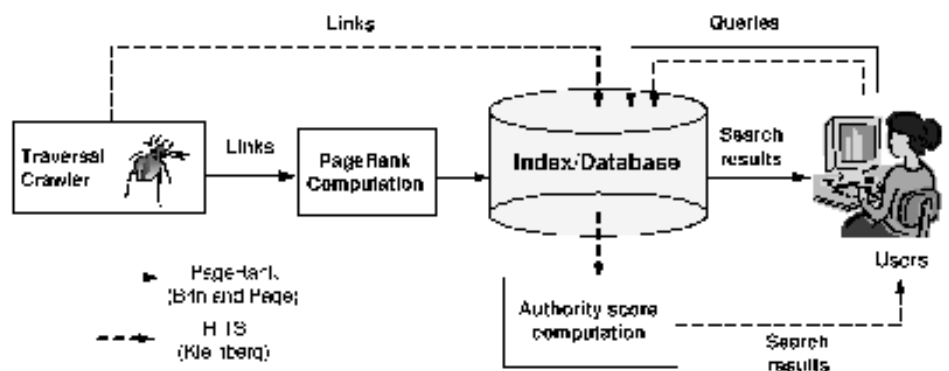


Figure 4 Ranking computation: HITS versus PageRank

The algorithm stops when  $\pi^{(k)} \sim \pi^{(k-1)}$ . In their first work, Brin and Page reported convergence in 50-100 iteration.

It can be shown using Perron-Frobenius theory that the difference between the approximation  $\pi^{(k)}$  and the real PageRank value  $\pi$  is of the order  $c^k$  (see e.g. [13]). Thus, the power iterations converge exponentially, while smaller values of  $c$  ensure an even faster convergence, which is a valid reason to keep  $c$  not too close to 1. On the other hand, in (1), the term that depends on hyperlinks decreases with  $c$ , so small  $c$  results in an almost uniform PageRank. Hence, a reasonable compromise has to be found, and Google's original choice was  $c = 0.85$ . We refer to the interesting and extremely well written survey [13] for more detail on this respect.

The present value of  $c$  and the actual algorithm used by Google nowadays is not known to the public but nevertheless, the PageRank distribution still plays an important role in defining the order of search results. Moreover, according to the publicly available information, power iterations are still used for the

PageRank computation. There are a lot of intelligent techniques developed for making the power method more efficient, such as parallel computing, block iteration methods, rearranging, two-stage methods, and many others [13–15].

Alternative algorithms that allow to compute the PageRank on-line while crawling the Web also exist. One of the methods that works surprisingly well is a *Monte Carlo* algorithm [2]. In a nutshell, this algorithm runs a random surfing process from each page. If a random jump has to be made, the simulation stops and then starts from the next page. At the end, the PageRank of page  $i$  is computed as the number of visits to this page divided by the total number of steps performed. Surprisingly, it is sufficient to run such simulation only *once* from each page to obtain a reasonable estimate of the PageRank.

Another intelligent on-line method is proposed in [1]. Initially, each page receives an equal amount of cash, and whenever a page is crawled, it distributes all its cash among its outgoing links. After several crawls, the

importance of a given page is evaluated as a fraction of cash spent by this page compared to the total amount spent by all pages together. The algorithm converges very fast, does not require any storage of the hyperlink matrix, and quickly adopts to the changes in the Web. Besides, analytical studies of this algorithm give rise to many interesting mathematical problems.

Although the PageRank is not directly related to the number of incoming links there is an intimate connection between these two measures of page popularity. For instance, it turns out that the fraction of pages whose PageRank is about  $k/n$  is roughly proportional to  $k^{-2.1}$  [16], exactly the fraction of pages with  $k$  incoming links! Since the models with preferential attachment explain the power law phenomenon for the in-degree, it is interesting to study the PageRank and its evolution in these models. This leads to a whole class of challenging research problems in the novel exciting area of complex stochastic networks. ◀

## Referenties

- 1 S. Abiteboul, M. Preda and G. Cobena, 'Adaptive on-line page importance computation', in: *The Twelfth International World Wide Web Conference WWW2003*, 2003.
- 2 K. Avrachenkov, N. Litvak, D. Nemirowsky and N. Osipova, *Monte Carlo methods in PageRank computation: When one iteration is sufficient*, Technical Report 1754, University of Twente, 2005.
- 3 A.-L. Barabási, and R. Albert 'Emergence of scaling in random networks', *Science* **286** (1999), pp. 509–512.
- 4 B. Bollobás, O. Riordan, J. Spencer and G.E. Tusnády, 'The degree sequence of a scale-free random graph process', *Random Struct. Algorithms* **18**(3) (2001), pp. 279–290.
- 5 B. Bollobás and O. Riordan, 'The diameter of a scale-free random graph', *Combinatorica* **4** (2004), pp. 5–34.
- 6 S. Brin, L. Page, R. Motwami and T. Winograd *The PageRank citation ranking: bringing order to the web*, Stanford University Technical Report, 1998
- 7 A.Z. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins and J.L. Wiener, 'Graph structure in the Web', *Computer Networks* **33** (2000), pp. 309–320.
- 8 S. Dill, R. Kumar, K. McCurley, S. Rajagopalan, D. Sivakumar, and A. Tomkins, 'Self-similarity in the Web', *ACM Trans. Internet Technology* **2**(3) (2002), pp. 205–223.
- 9 M.R. Henzinger, 'Algorithmic challenges in Web search engines', *Internet Mathematics* **1**(1) (2003), pp. 115–126.
- 10 S. Karlin and H.M. Taylor, *An Introduction to Stochastic Modeling*, Academic Press, San Diego, (1998).
- 11 J. Kleinberg, 'Authoritative sources in a hyperlinked environment', *Journal of the ACM* **46** (1999), pp. 604–632.
- 12 M.E.J. Newman, 'The structure and function of complex networks', *SIAM Rev.* **45**(2) (2003), pp. 167–256.
- 13 A.N. Langville and C.D. Meyer, 'Deeper inside PageRank', *Internet Mathematics* **1**(3) (2005), pp. 335–380.
- 14 A.N. Langville and C.D. Meyer, *A reordering for the PageRank problem*, NCSU CRSC Technical Report CRSC-TR04-16, 2004.
- 15 C.P.-C. Lee, G.H. Golub and S.A. Zenios, *A fast two-stage algorithm for computing PageRank*, Stanford University Technical Report, 2004.
- 16 G. Pandurangan, P. Raghavan and E. Upfal, 'Using PageRank to characterize Web structure', *8th Annual International Computing and Combinatorics Conference (COCOON)*, 2002.