From *SIAM News*, Volume 37, Number 9, November 2004

The Ongoing Search for Efficient Web Search Algorithms

By Sara Robinson

Type "newspapers" into the box on Google's search site, and your search will pull up dozens of pages containing comprehensive lists of links to newspaper sites. For some reason, though, Google will miss the sites of the newspapers themselves.

The same search on the lesser-known site Ask Jeeves, by contrast, produces the sites of newspapers, such as *USA Today*, *The Christian Science Monitor*, *The Daily Telegraph*, and *The New York Times*, in addition to the pages found by Google.

It is possible to ferret out the same information from both sets of results, but the comparison is telling: It reflects subtle differences between the underlying mathematical algorithms the sites use to rank their search results. These algorithms, developed in the 1990s, are still a focus of research today.

A Key Idea

By the mid-1990s, the Web had burgeoned into a vast storehouse of information, with hundreds of millions of pages on thousands of topics. Yet for someone seeking reliable information on a particular topic, the local library was still a better option.

Internet search engines of the time, such as those offered by AltaVista, Excite, Lycos, and Hotbot, relied completely on techniques from information retrieval systems to find documents that contained the search terms prominently and with high frequency. But such tools were designed for databases of limited size, consisting of uniformly high-quality information presented in a controlled style and structure. The Web, with its grassroots structure, is a chaotic jumble of pages of variable quality offered in a variety of formats. Confronted with this morass, IR techniques alone would rank junk pages highly if they happened to contain multiple instances of the search term. At the same time, the techniques would miss relevant pages that did not include the term at all.

A search for "cars" or "automobile manufacturers," for example, would not return the Web site of Honda or Ford. A search for lung cancer might rank documents for lawsuits against the tobacco industry higher than an authoritative health site.

Simple keyword searches had another disadvantage: People wishing to draw more traffic to their Web sites could easily exploit them. Businesses quickly learned how to push their Web sites onto the first page of search results: simply incorporate popular search terms, repeated over and over in invisible fonts.

The only hope for extracting useful information from the Web, it seemed, was to have an army of human researchers play the role of a librarian, sifting through the clutter and selecting and organizing the high-quality pages on each topic. Indeed, some popular search sites, such as Yahoo, took exactly this approach, building directories of Web sites that were reviewed and indexed by employees. But with the Web adding hundreds of thousands of pages each day, such directories were necessarily limited in scope.

Faced with two limiting options, computer scientists began to wonder about a system that would have the best features of both. Could they devise a search tool that combined the speed and breadth of the automated tools with some of the intelligence provided by human reviewers? The answer, as it turned out, was yes.

Just as people writing papers cite other papers that motivated their work, people designing Web pages link to other Web pages they like. . . . The accurate, useful Web pages on a topic tend to link to pages of similar quality.

In the late 1990s, two groups of computer scientists came up with different, yet essentially similar approaches to search. The key idea behind both approaches was to mine the human intelligence built into the link structure of the Web: Just as people writing papers cite other papers that motivated their work, people designing Web pages link to other Web pages they like. Most important, the accurate, useful Web pages on a topic tend to link to pages of similar quality.

One of the approaches, offered by two Stanford graduate students in computer science, became the basis for Google. The other, developed by researchers at IBM, has been used extensively in customized search applications for IBM's business customers but was not used in a public search engine until 2000, when computer scientists at Rutgers University incorporated the ideas into a search engine that is now part of Ask Jeeves.

Both methods are highly mathematical, requiring real-time computations on enormous data sets that grow with the size of the Web. Applied mathematicians are now rising to the challenge, finding more efficient ways to implement these complex ranking algorithms.

The IBM Method: Authorities and Hubs

In 1996, Jon Kleinberg, then fresh out of graduate school in computer science, arrived in San Jose, California, for a year-long stint at IBM's Almaden Research Center. At the time, he remembers, there was a lot of interest in the problem of Internet search: "Talks would all start with the same mantra: 'Web search isn't very good.'"

Kleinberg, who is now at Cornell University, had spent part of the previous summer learning about RNA structures; while searching for helpful pages on the topic, he had also gleaned insight into the structure of the Web itself. Sifting through the scores of results called up by keyword searches, he eventually noticed that high-quality sources, such as repositories of research papers and home pages of prominent researchers, tended to be referenced by high-quality reference pages, like course home pages and carefully compiled bibliographies. Thus, Kleinberg realized, useful pages on a topic are of two types: Authoritative sources contain information on the topic; reference pages, which he calls "hubs", provide links to sources. To Kleinberg, this suggested that the links in the reference pages encode judgments about the quality of the source pages; the quality of the reference pages, in turn, can be inferred from the quality of the source pages they link to.

To find the best hubs and authorities on a particular topic, he devised the following procedure: First, he defined a page's hub weight as the sum of the authority weights of the pages it points to, and its authority weight as the sum of the hub weights of the pages pointing to it. Then, starting with a set of results from a standard, keyword search, he added pages that linked to and were linked to by those pages; in this way, he ensured the inclusion of good hubs and authorities that might not contain the keywords. Using approximate values for the hub and authority rankings of each page, he then followed an iterative procedure, repeatedly updating the rankings until they remained fixed.

More specifically, for a collection of Web pages obtained from a keyword search, he started by assigning each page an authority weight and a hub weight of one, and then normalizing to make the sum of the squares of each type of weight equal one. Then, taking the connectivity matrix M for the collection of pages (i.e., $m_{ij} = 1$ if there is a link from page i to page j and 0 otherwise) and multiplying its transpose by the vector of hub weights updates the authority weights. Similarly, multiplying M by the vector of authority weights updates the hub weights. For repeated updates, the hub weight vector or the authority weight vector is multiplied by the matrix MM^T or M^TM, respectively.

Because each of these matrices is non-negative and positive semi-definite, an eigenvector can be computed via the power method, i.e., by repeatedly applying the matrix to a vector and then renormalizing the result. If the matrix is tweaked to make it irreducible, the Perron–Frobenius theorem guarantees convergence to a positive, dominant eigenvector that is unique when normalized. For MM^T, this eigenvector will correspond to the hub weights; application of M^T to that eigenvector yields the authority weights.

When he used this method to rank the results for a keyword search, Kleinberg found a dramatic improvement in the relevance of the results. The method would home in on the mini-networks that self-organized around a given topic, and prominently display the most popular pages within the most densely linked cluster of pages. Kleinberg first published his results as an IBM tech report later that year, followed by a paper in the *Proceedings of the Ninth Annual ACM–SIAM Symposium on Discrete Algorithms* in 1998. A group of IBM researchers led by Prabhakar Raghavan, now chief technologist at Verity, Inc., continued to refine the approach, eventually using it as a basis for customized search applications.

The Stanford Approach

Meanwhile, a few miles away, Stanford graduate students Sergey Brin and Larry Page were also thinking about the use of links for Web search rankings. They had already considered computational and storage issues for search engines, in work that led to a prototype design of a search engine based on clusters of PCs. In 1998, they described an approach to ranking search results that differed slightly from Kleinberg's.

For their model, they imagined a Web surfer, starting at a Web page and then riding from one page to another along randomly chosen links. Often, such link chains lead to dead ends—pages with no outgoing links—or circles of interconnected pages. Thus, a certain fraction of the time, they also had their Web surfer jump to a randomly chosen page anywhere on the Web. This surfing pattern forms a Markov chain with a stationary distribution in which each Web page has a limiting probability of being visited. Brin and Page called this probability the PageRank of that page.

The PageRank of a Web page is a measure of how popular it is as a function of the Web's inlinks and outlinks. By listing the results of a keyword search in order of their PageRank, Brin and Page reasoned, search engines could do a better job of directing people to the most popular of the pages containing their keywords.

To compute a PageRank value for every page on the Web, Brin and Page started with the enormous connectivity matrix (m_{ij}) for the entire indexed Web. Assuming that the surfer follows a link from the current page a fraction p of the time and takes a random hop $1 - p$ of the time, they constructed a second $n \times n$ matrix (g_{ij}), the Google matrix, which is the transition probability matrix for the Markov chain. Here, $g_{ij} = p\, m_{ij}/b_i + \delta$, where $\delta = (1 - p)/n$ and b_i is the outdegree of page i, assuming that i is not a dangling node (dead-end page). (Originally, Google chose p to be 0.85.) After a small adjustment to compensate for dangling nodes, the matrix G becomes an irreducible, stochastic matrix, and the Perron–Frobenius theorem again guarantees a unique normalized left eigenvector whose entries are the PageRanks of all of the pages on the Web.

(Brin, Page, and other Google researchers were not available for interviews for this article because of the SEC-mandated "quiet period" following the company's recent IPO.)

The Methods Compared

Because there is a clear set of top authorities for most topics, the two methods typically produce similar search results. For a significant fraction of queries, though, the methods differ in interesting ways. Some sites, such as those of major newspapers or universities, often don't contain generic descriptions of themselves; a Google search will miss such sites, returning only hub pages that point to them. Kleinberg's method, by contrast, finds both the hubs and the sites themselves.

A search on "buffalo" demonstrates the ability of Kleinberg's algorithm to tease out small communities. Google lists scores of

pages of sites related to the city of Buffalo, New York, before getting to the animal, presumably because the PageRank for the relatively obscure animal sites is dwarfed by those of the sites about the city. Kleinberg's algorithm gives high rankings to sites of both types.

The methods are computationally distinct as well. The Brin–Page approach uses a single ranking of all Web pages to return appropriate results for a search, while the Kleinberg method customizes its ranking to each search. Because PageRank is query-independent, Google can compute it offline, updating only once every month or so to reflect the growth and evolution of the Web. Kleinberg's method requires a real-time ranking computation following each search. At first this seemed to be a prohibitive barrier to its use in a large-scale search engine.

Because PageRank is query-independent, Google can compute it offline, updating only once every month or so to reflect the growth and evolution of the Web. Kleinberg's method requires a real-time ranking computation following each search.

Another downside to Kleinberg's method is its vulnerability to spamming. A person can influence the hub score of a site just by adding links to popular sites. Having established a few good hubs, he can then use them to improve the authority score of any site he chooses.

Realizing the Ideas: Google and Teoma

After developing their search techniques, Kleinberg says, Brin and Page shopped them around to some of the popular search sites. The new techniques sparked little interest—at the time, the industry focus was on developing portal sites that would control a Web user's experience. Search site executives told Brin and Page that as long as their site's search experience wasn't significantly worse than that of its competitors, they saw little to be gained by improving it, Kleinberg says.

Eventually, Brin and Page decided to start their own Web search company, which they named Google, after "googol," which means 10 to the 100th power. Launched in 1999, the Google site was immediately popular, although it took a few years and another, clever, mathematically based idea—Google Adwords—before the profits started rolling in.

As the Web has evolved, so has Google: In Kleinberg's words, "Its plain face hides a monster of ever increasing complexity." While Google relies heavily on PageRank for ranking its search results, it uses at least a hundred other metrics as well, making use of such things as the content of "anchor text," the highlighted description a user clicks on to follow a link. Such methods are powerful heuristics for sharpening the relevance of link analysis, but they also leave Google more vulnerable to spammers. A search on the term "miserable failure," for instance, returns a Web page about George Bush as the top result, a type of mischief known as "Google bombing." To help thwart spammers, Google keeps its exact ranking methods secret and changes them frequently.

IBM, too, tried to market the new ideas to search sites, also meeting with little success. Instead, the company elected to focus on business applications. It wasn't until 2000 that Kleinberg's ideas were realized in a large-scale consumer search engine called Teoma, the Gaelic word for "expert."

Developed by computer scientists at Rutgers University and acquired by Ask Jeeves in 2001, Teoma provides search results in three categories: results, suggestions for refinement, and link collections compiled by experts and enthusiasts. The results section is similar to Kleinberg's authorities, while the link collections are like his hubs. The suggestions for refinement correspond to terms describing the naturally occurring linked Web communities related to the search.

Teoma founder and chief scientist Tao Yang, who retains the same title at Ask Jeeves, acknowledges that the Teoma founders "learned a lot" from the IBM "demo" project, but says that Teoma has a different, more scalable, approach to solving the problem.

"The issue is how to do it realistically on the fly," he says. "We are the only ones who have figured out how to do it and we don't want to say how we do it." Teoma's algorithm, he adds, extracts the Web communities, whereas the IBM algorithm does not.

Asked whether the company pays licensing fees to IBM for its patent on Kleinberg's method, Jim Lanzone, senior vice-president of search properties for Ask Jeeves, says that it does not. "If there were patent issues, we wouldn't have been able to go forward," he says.

Nevertheless, the roots of Ask Jeeves are apparent in its search results. Searches on "newspapers" and "buffalo" produce re-sults consistent with Kleinberg's approach.

Updating PageRank

Both the hubs-and-authorities algorithm and the PageRank algorithm continue to be active topics of research. Due to the success of Google, the PageRank approach in particular has attracted the interest of independent researchers.

Computer scientists began contributing ideas to PageRank soon after the Brin and Page paper appeared, but many applied mathematicians first learned about the PageRank update problem only in 2002. That year, Cleve Moler described the problem in an article in *MATLAB News & Notes*, and Craig Silverstein, Google's director of technology, gave a talk at SIAM's 50th anniversary meeting in Philadelphia. Carl Meyer, a professor of mathematics at North Carolina State University, was in the audience at the Silverstein talk.

Silverstein was "very forthcoming," Meyer recalls, telling the audience that although the Google matrix was, at that time, of order two billion, the company still used the power method to update PageRanks. The computation would start from scratch each time, but even so, the method would converge in only 100-200 steps. (Convergence depends on the second eigenvalue of the matrix, which has magnitude equal to the parameter p, the probability that a link is followed in the Markov chain.)

Many in the SIAM audience were astonished that the company was using such a naïve method for such a large matrix, Meyer says, but Silverstein explained its advantages. In addition to exploiting the sparsity of the Google matrix and the inherent parallelism

in the problem, the power method is simple enough to program that Google can easily adjust its algorithm. Some of the more sophisticated methods, such as subspace iteration, are not feasible for Google because of their storage requirements, Silverstein pointed out.

Intrigued by what he had heard, Meyer began to work on the update problem. "Producing stationary distributions of Markov chains is something I know how to do," he says. He was soon joined by Amy Langville, a postdoc who had done Markov chain-related work for her thesis.

Meyer and Langville have developed a PageRank updating approach that uses iterative aggregation, a technique introduced by economists in the 1960s for finding stationary distributions of Markov chains. The basic idea, which Meyer described in July in a talk at the 2004 Siam Annual Meeting, is to use a divide-and-conquer approach to cut the Markov chain into smaller chains that can be solved independently. The stationary distributions of the smaller chains are then glued together to obtain the steady-state solution of the parent chain.

For the method to be effective, the chain needs to subdivide in a nice way. Because the Web has been shown empirically to be structured like a scale-free network, with the popularity of sites following a power law distribution, the natural subdivision is to lump together the majority of Web sites—which are visited infrequently and whose PageRank doesn't often change—and treat the few, dynamic and heavily trafficked sites as individual states. "It's a specialized technique that only works because of the structure of the Web," Meyer says. Because the approach allows the researchers to work with problems of a much smaller size, storage is not an issue.

Langville and Meyer lack access to the sort of computing resources that would enable them to try their method on data sets on the scale of the entire Web. Nevertheless, they have applied their method to communities within the Web containing tens of thousands of sites and achieved significant improvements in run-time—30% to 80%—over accelerated power methods. The improvement increases with the size of the data set, the researchers say. Meanwhile, the number of pages Google indexes has grown to 4.3 billion, making such alternatives to the power method even more attractive.

When Meyer and Langville presented their work at a recent conference, Google researchers in attendance expressed interest in the work, Meyer says. Since the 2002 Silverstein talk, however, Meyer and Langville say that they have been unable to learn any more details about Google's evolving methods for computing PageRank. (Readers interested in current work on search rankings might want to consult Langville and Meyer's *Understanding Search Engine Rankings*, to be published by Princeton University Press.)

The Rich Get Richer

Even as mathematicians direct their attention to efficient search algorithms, the field continues to evolve in ways that present new computational challen-ges. One new focus is personalization of search results. Google has created a personalized PageRank metric that is available as a demo on its Web site, and Amazon.com has entered the search business with a subsidiary called A9 that plans to personalize search. (The possibilities of personalized search are intriguing to many researchers but sound ominous to many consumers, including this one.)

Another new search engine, launched by Vivisimo, does something similar to Teoma's Web communities, but without using links. The search tool, called Clusty, automatically clusters search results into categories selected from words and phrases contained in the search results themselves.

Whatever the focus, increased competition in search is a good thing, some researchers say, because the dominance of Google influences the structure of the Web itself. Because people are most likely to link to sites that they can easily find on Google, PageRank becomes a self-enhancing function with the highly ranked sites becoming even more highly ranked, a phenomenon known as "the rich get richer."

Sara Robinson is a freelance writer based in Los Angeles, California.