The Sleekest Link Algorithm

Desmond J. Higham* and Alan Taylor**
Department of Mathematics, University of Strathclyde, Glasgow G1 1XH, UK.

How does Google decide which web sites are important? It uses an ingenious algorithm that exploits the structure of the web and is resistant to hacking. Here, we describe this PageRank algorithm, illustrate it by example, and show how it can be interpreted as a Jacobi iteration and a teleporting random walk. We also ask the algorithm to rank the undergraduate mathematics classes offered at the University of Strathclyde. PageRank draws upon ideas from linear algebra, graph theory and stochastic processes, and it throws up research-level challenges in scientific computing. It thus forms an exciting and modern application area that could brighten up many a mathematics class syllabus.

Introduction

Everybody knows about the search engine Google, and most people use it. Why is it so successful? At the Google website http://www.google.com/technology/ all is revealed.

> 'The heart of our software is PageRank™; a system for ranking web pages developed by our founders Larry Page and Sergey Brin at Stanford University. And while we have dozens of engineers working to improve every aspect of Google on a daily basis, PageRank continues to provide the basis for all of our web search tools.'

PageRank, a sleek algorithm in computational graph theory, shows how one killer mathematical idea can build up a global brand name. Google began as a research project for Ph.D. candidates Page and Brin when they were, respectively, 24 and 23 years old. It now answers over 200 million queries per day.

Our aim here is to describe PageRank, illustrate it via simple examples, and use it to pull together ideas from numerical analysis and stochastic processes. We also point out, via a somewhat frivolous example, how its utility extends well beyond the world wide web.

The observations in sections 4 and 5 are not new. Indeed, both the linear system/eigenvector formulation and the random walk interpretation are mentioned in the original work [15]. However, we believe that there are benefits to be had from a unified, low-level review – in particular, teachers in further and higher education may find that this material can be slipped into a class on linear algebra, graph theory, stochastic processes or scientific computation. We also see it as a jumping-off point for student projects.

The PageRank Algorithm

Search engines must do a number of tasks. Here are three important ones.

Task 1 Locate web pages and store pertinent information in some sort of archive.
Task 2 In response to a user's query, perform a real-time computation on the archive to find a list of relevant web pages.
Task 3 Decide the order in which to report these pages to the user, on the basis of (a) their relevance to the query and (b) their overall importance.

PageRank pertains exclusively to the third task. Its mission is to measure the importance of each page on the web, so as to inform part (b) of the beauty contest.

Although the precise details of Google's inner workings are not available to the general public, the basic PageRank algorithm has been publicised by Page and Brin [15]. Typing '**PageRank**' into Google brings up many articles, tutorials, essays, threads, and even diatribes. As is the way with the web, these are of variable quality. (Presumably, those with the highest PageRankings will be the best!)

Table 1: Entries in the adjacency matrix for Figure 1. Final column shows the degree of each page.

	Home	Biography	Photos	Hobby	degree
Home	0	1	1	1	3
Biography	1	0	0	0	1
Photos	1	0	0	0	1
Hobby	1	0	1	0	2

To describe the algorithm, we first think of the web as a directed graph. Figure 1 shows a dramatically simplified model of the web that consists of 4 pages, HOME, PHOTOS, HOBBY and BIOGRAPHY. The arrows indicate links, so, for example, there is a link from HOME to BIOGRAPHY and from HOBBY to PHOTOS. We can store this information in a 4 by 4 *adjacency matrix*, W, as illustrated in Table 1. Generally, $W(i, j)$ is equal to 1 if there is a link from the ith page to the jth page, and zero otherwise. The *degree* of the ith page (or, more precisely, the out degree) is defined as the total number of links going out from that node. Equivalently, it is the sum of the entries along the ith row of the adjacency matrix. Table 1 includes the degree information for Figure 1.

To generalize this concept, we need some notation.

- Let there be N web pages, arbitrarily labelled 1, 2, . . . , N.

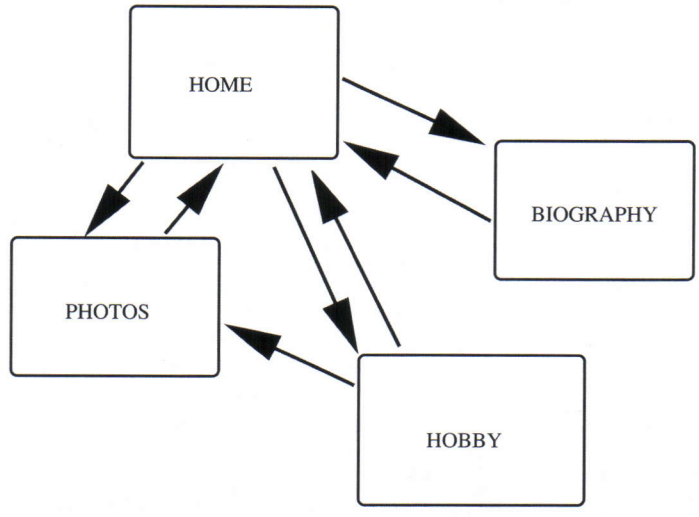


Figure 1: Example of web pages with connections.

* Supported by a Research Fellowship from The Leverhulme Trust.
** Supported by a Vacation Scholarship from the Carnegie Trust for the Universities of Scotland.

- Let W be the $N \times N$ adjacency matrix, so the (i, j) element, w_{ij}, equals 1 if there is a link from page i to page j, and equals zero otherwise.
- Let $\deg_i$ denote the degree of the ith node, so $\deg_i := \sum_{j=1}^{N} w_{ij}$.

We also assume that that $\deg_i \neq 0$ for all $1 \leq i \leq N$. (In practice, pages with no outgoing links – *dangling pages* – must be treated specially.)

The PageRank algorithm proceeds iteratively, assigning a value $r_j^{[n]}$ to the jth page at the nth iteration. The iteration is

$$r_j^{[n]} = (1-d) + d\sum_{i=1}^{N} \frac{w_{ij} r_i^{[n-1]}}{\deg_i}. \qquad (1)$$

Here, d is some constant in the range $0 < d < 1$. We will use $d = 0.85$ unless otherwise stated. The iteration can be justified quite easily. The key idea is to regard a link from i to j as a vote of confidence for page j from page i. So the importance of page j can be measured by looking at the links coming in to that page. It makes sense to weight the votes according to the level of importance of the voter – a vote from an important page should carry more weight than a vote from an unimportant page. It also makes sense to give each node an equal influence by scaling the weight of its vote by the number of votes that it casts. This means that the weight w_{ij} gets scaled to $w_{ij} / \deg_i$. The final twist in the algorithm is the introduction of the constant d. Each node gets given a ranking of $1 - d$ for free, and then gets d times the value arising from those votes of confidence. (We will see later that d is a little less mysterious when other interpretations of the algorithm are taken.)

Summarizing the arguments above, the iteration (1) for updating the ranking of page j could be described as follows.

For each page i that points to j, add in the scaled value of the current ranking, $r_i^{[n-1]} / \deg_i$.

Take d times this sum and add $1 - d$.

Examples

For the example in Figure 1 we have

$$W = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix}.$$

Taking $d = 0.85$ and starting with the initial guess $r_i^0 \equiv 1$, we obtained the following results, to four decimal places,

	Iteration number					
	1	2	3	...	19	20
HOME	1.000	2.2750	1.4321	...	1.7687	1.7697
BIOGRAPHY	1.000	0.4333	0.7946	...	0.6515	0.6511
PHOTOS	1.000	0.8583	0.9788	...	0.9282	0.9280
HOBBY	1.000	0.4333	0.7946	...	0.6515	0.6511

We see that after 20 iterations, the PageRank values appear to have settled down in their first two decimal places. (Convergence of the algorithm is discussed in the next section.) As we would expect, HOME, which is pointed to by all other pages, receives the highest ranking. PHOTOS is next, because of the link there from HOBBY, and BIOGRAPHY and HOBBY share last place.

Suppose we alter the network by adding a link from PHOTOS to BIOGRAPHY, see Figure 2. Because PHOTOS is more important than HOBBY, we would expect this new link to count more than the link from HOBBY to PHOTOS, so BIOGRAPHY should jump up the rankings. PageRank on this new network gives

	Iteration number					
	1	2	3	...	19	20
HOME	1.000	1.850	1.4285	...	1.5851	1.5852
BIOGRAPHY	1.000	0.8583	0.0390	...	0.9620	0.9620
PHOTOS	1.000	0.8583	0.8583	...	0.8538	0.8538
HOBBY	1.000	0.4333	0.6742	...	0.5991	0.5991

which confirms our prediction.

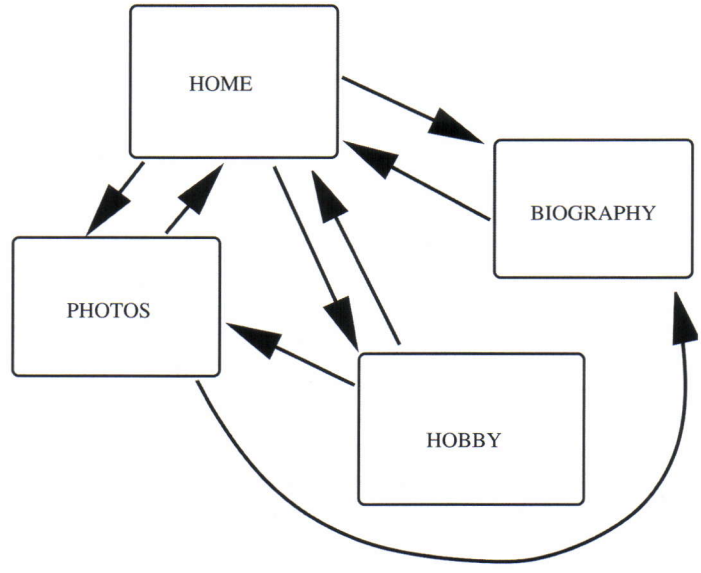


Figure 2: Example of web pages with connections.

Figure 3 shows a network with a home page and a set of five lectures. Each lecture points to the next, with the last one pointing back to HOME. No matter what ordering we use for the pages, the adjacency matrix for this network is

$$W = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

This symmetry suggests that all pages should have equal PageRank, and this is confirmed by the observation that $r_j \equiv 1$ is a fixed point of the iteration (1).

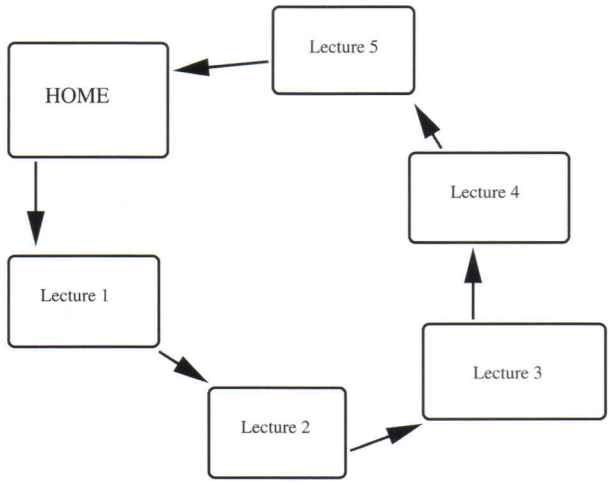


Figure 3: Example of web pages with connections.

If we alter the network so that all lectures point back to HOME, as illustrated in Figure 4, then the ranking becomes, to four decimal places,

HOME	1.9879
LECTURE 1	1.8397
LECTURE 2	0.9319
LECTURE 3	0.5460
LECTURE 4	0.3821
LECTURE 5	0.3124

which is perhaps a more reasonable reflection of the relative importance of the pages.

Our examples suggest that you can tinker with the Google PageRank values for your web pages by reorganizing the local connectivity structure. While this is true, and the relative orderings of your pages may change, the effect on the absolute PageRankings is likely to be minimal. Indeed, one of the big advantages of PageRank over earlier systems is its insensitivity to *spammers* who seek to inflate their rank by devious means. To get a high overall PageRank, your page must be referenced by other high-ranking pages, something that is hard for any particular individual to influence. Also, by focussing solely on the link structure, and not the web page content, PageRank, unlike some predecessors, cannot be misled by the insertion of important sounding words or phrases. Of course the system is not completely hack-proof. The article [2] mentions that a concerted effort by one cybersmith convinced Google to return his friend's home page as the top-ranking match to 'talentless hack.' That article also describes how a well-referenced spoof site

http://www.coxar.pwp.blueyonder.co.uk/

became the number one choice for 'weapons of mass destruction.' There is also a positive feedback aspect to the process: lots of websites now refer to those two examples, giving them extra votes of confidence. A brute force approach to rank inflation may also be adopted; some commercial sites are willing to pay for incoming links [7].

The documents [3, 16] and the book [1] give more details about how web administrators can use PageRank to their advantage.

Having discussed and illustrated the algorithm, in the next two sections we show that PageRank can be looked at from other angles.

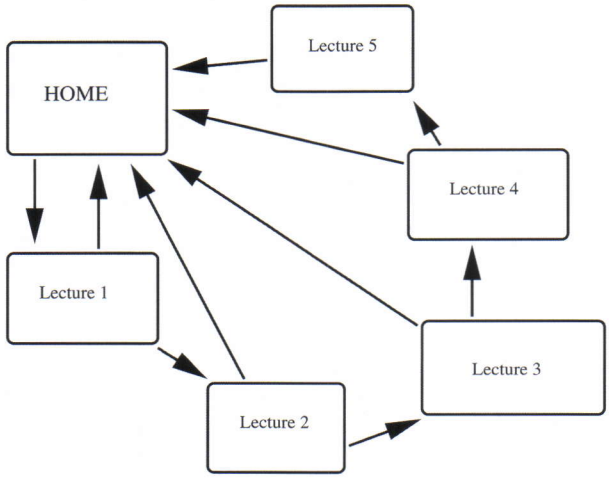


Figure 4: Example of web pages with connections.

Jacobi Iteration

Suppose we have a linear system of simultaneous equations, $A\mathbf{x} = \mathbf{b}$, where $A \in \mathbb{R}^{N\times N}$ and $\mathbf{b} \in \mathbb{R}^N$ are given and $\mathbf{x} \in \mathbb{R}^N$ is to be found. If we have a current guess for the solution, $\left\{x_k^{[n-1]}\right\}_{k=1}^N$, then we may use the jth equation,

$$\sum_{k=1}^{N} a_{jk}x_k = b_j,$$

to find a new value for x_j. Inserting $x_k^{[n-1]}$ for $k \neq j$ and solving for x_j in this way, we get the iteration

$$x_j^{[n]} = \frac{1}{a_{jj}}\left(b_j - \sum_{k=1,\,k\neq j}^{N} a_{jk}x_k^{[n-1]}\right).$$

Using this formula for $j = 1, 2, \ldots, N$, gives what is known as the *Jacobi iteration*. Of course, we require $a_{jj} \neq 0$ for all $j = 1, \ldots, N$ in order for the iteration to be well defined. If we split A into

$$A = \Sigma + L + U,$$

where Σ is diagonal, L is strictly lower triangular and U is strictly upper triangular, then standard numerical analysis results, see for example, [14], show that the iteration converges to a unique solution $\mathbf{x}$ if

$$\rho(\Sigma^{-1}(L+U)) < 1. \tag{2}$$

Here ρ denotes the *spectral radius* of a matrix, that is, the largest of the moduli of the eigenvalues. Further, the convergence is linear with a rate at least as fast as ρ: given any vector norm $\|\cdot\|$, there is a corresponding constant C such that the error, $\mathrm{err}^{[n]} \in \mathbb{R}^N$ with $\mathrm{err}_k^{[n]} = r_k^{[n]} - x_k$, satisfies

$$\left\|\mathrm{err}^{[n]}\right\| \leq C\rho^{[n]}.$$

Simple algebra shows that the PageRank iteration (1) is precisely the Jacobi iteration applied to the system

$$\left(I - dW^T D^{-1}\right)\mathbf{r} = (1-d)\mathbf{e}, \tag{3}$$

where I is the identity matrix,

$$I = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} \in \mathbb{R}^{N\times N},$$

$W^T \in \mathbb{R}^{N\times N}$ is the transpose of W, so $(W^T)_{ij} = (W)_{ji}$, D is the diagonal degree matrix,

$$D = \begin{bmatrix} d_1 & 0 & \dots & 0 \\ 0 & d_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & d_N \end{bmatrix} \in \mathbb{R}^{N\times N},$$

and $\mathbf{e}$ is the vector of ones,

$$\mathbf{e} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^N.$$

The left-hand side of (2) reduces to $\rho(dW^TD^{-1})$ in this case. By construction, W^TD^{-1} has column sums equal to one, so $\|W^TD^{-1}\|_1 = 1$, where $\|\cdot\|_1$ denotes the L_1 norm. Since the spectral radius is bounded above by any subordinate vector norm we have $\rho(W^TD^{-1}) \le 1$, so

$$\rho(dW^TD^{-1}) \le d.$$

So choosing d < 1 ensures that we satisfy the convergence condition (2).

It is clear from (1) that nonnegativity in the starting vector is preserved; that is

$$r_k^{[0]} \ge 0 \text{ for all } 1 \le k \le N \Rightarrow r_k^{[n]} \ge 0 \text{ for all } 1 \le k \le N \text{ and } 1 \le n.$$

Hence the solution $\mathbf{r}$ must have nonnegative components. Premultiplying by $\mathbf{e}^T$ in (3) gives

$$\mathbf{e}^T\mathbf{r} - d\mathbf{e}^TW^TD^{-1}\mathbf{r} = (1-d)\mathbf{e}^T\mathbf{e},$$

which simplifies to

$$\|\mathbf{r}\|_1 - d\|\mathbf{r}\|_1 = (1-d)N.$$

So $\|\mathbf{r}\|_1 = N$, a fact that can be checked in the examples of the previous section.

In summary

> the PageRank iterates converge to the solution r of the linear system (3). This solution satisfies $r_k \ge 0$ and $\sum_{k=1}^N r_k = N$. The PageRank algorithm corresponds to the Jacobi iteration applied to this system, and it converges linearly with rate at least d.

It is perhaps worth emphasizing that d not only controls the linear convergence rate but also affects the solution. For example, changing from $d = 0.85$ to $d = 0.7$ in the example of Figure 4 compresses the PageRanks to

HOME	1.9020
LECTURE 1	1.6314
LECTURE 2	0.8710
LECTURE 3	0.6048
LECTURE 4	0.5117
LECTURE 5	0.4791

In the extreme, and non-useful, case where d = 0, from any starting vector $\mathbf{r}^{[0]}$ the iteration produces the exact solution, $\mathbf{r}^{[1]} = \mathbf{e}$, in a single step.

A Teleporting Random Walk on the Web

To connect PageRanking to stochastic processes, we first introduce a few basic concepts. A discrete time, finite state, *Markov chain* is determined by a *transition matrix*, $P \in \mathbb{R}^{N\times N}$. Given that the process is at state i at time $n-1$, p_{ij} gives the probability that the process will be at state j at the next time level, n. Because the process must be somewhere at time n, those probabilities must sum to one; $\sum_{k=1}^N p_{ik} = 1$. This means that P is a *stochastic* matrix.

In many cases, the probability of being in state j after a large number of steps settles down to a fixed value π_j, no matter what initial state was chosen. Equivalently, π_j is the proportion of time that we spend at state j in the long time limit. Such a vector $\pi \in \mathbb{R}^N$, which must satisfy $\sum_{k=1}^N \pi_k = 1$, is called an *invariant measure*. If it exists it will satisfy $P^T\pi = \pi$, that is, it will be an eigenvector of P^T corresponding to the eigenvalue 1 [13, 17]. There can be no eigenvalues bigger than 1 in modulus, because a stochastic matrix has $\|P^T\|_1 = 1$, and $\rho(P) \le \|P^T\|_1$.

Consider now the prospect of taking a random walk on the web. Having visited a web page at time $n-1$, we look at the links coming out of that page and choose one of them at random, uniformly, to visit at time n. If we are currently at page i, then the probability of visiting page j on the next step is

$$\begin{cases} 0 & \text{if } w_{ij} = 0, \quad \text{(no link exists)}, \\ \frac{w_{ij}}{\deg_i} & \text{if } w_{ij} = 1, \quad \text{(a link exists)}. \end{cases}$$

The transition matrix for this Markov chain is thus $P = D^{-1}W$. Imagine surfing the web in this way for a few billion years. A measure of the 'well-connectedness' of page j is the proportion of time that we spend there, ultimately. This is precisely the quantity π_j mentioned above. A possible difficulty with this picture is that we may get stuck at a web page with no outgoing links, or more generally, may cycle around an isolated set of pages. To get around this, we may switch to a *teleporting random walk*. At each time, we flip a biased coin that lands heads with probability $1-d$. If the coin shows heads, we jump to a page chosen at random, uniformly, over the whole web. Otherwise we follow a link as described above. The transition matrix for teleporting is F/N, where $F \in \mathbb{R}^{N\times N}$ is a matrix with all entries equal to one. It follows that the transition matrix for the overall teleporting random walk is

$$P = \frac{(1-d)}{N}F + dD^{-1}W, \tag{4}$$

It is intuitively reasonable that the teleporting trick gets around difficulties associated with the construction of an invariant measure. More formally, it ensures that the Markov chain is *ergodic* so that there is a unique, nonnegative solution to the system $P^T\pi = \pi$, where we normalise so that $\sum_{k=1}^N \pi_k = 1$ [13, 17]. Using (4), this linear system is

$$\left(\frac{(1-d)}{N}F + d(D^{-1}W)^T\right)\pi = \pi,$$

which, since $F\pi = \mathbf{e}$ and $(D^{-1}W)^T = W^TD^{-1}$, may be written

$$(I - dW^TD^{-1})\pi = \frac{(1-d)}{N}\mathbf{e}. \tag{5}$$

Comparing (5) and (3), we make the connection that $\pi = \mathbf{r} / N$: the PageRank solution, when scaled by the factor N, is precisely the invariant measure for the teleporting random walk. We can take the connection further. Given a transition matrix we may apply an unscaled version of the *power method* ([4]) in an attempt to compute the invariant measure by regarding it as the eigenvector of the largest eigenvalue of P^T. Having chosen $\mathbf{x}^{[0]}$, with $x_i^{[0]} \geq 0$ and $\sum_{k=1}^{N} x_k^{[0]} = 1$, this gives $\mathbf{x}^{[n]} = P^T \mathbf{x}^{[n-1]}$. Using (4), this iteration becomes

$$\mathbf{x}^{[n]} = \left(\frac{1-d}{N} F + dD^{-1}W \right)^T \mathbf{x}^{[n-1]},$$

but since $F^T \mathbf{x}^{[n]} = \mathbf{e}$ for all n, it simplifies to

$$\mathbf{x}^{[n]} = \frac{(1-d)}{N} \mathbf{e} + dW^T D^{-1} \mathbf{x}^{[n-1]}.$$

This is precisely the original PageRank algorithm (1) with $N\mathbf{x}^{[n]}$ playing the role of $\mathbf{r}^{[n]}$.
In summary

> the normalized PageRank vector $\mathbf{r} / N$ is the unique invariant measure for a surfer who takes a teleporting random walk on the web, with transition matrix P in (4). The probability of a teleporting jump on each step of this process is $1 - d$. The PageRank algorithm corresponds to applying the power method to compute the invariant measure as the dominant eigenvector of P^T.

Ranking Mathematics Classes

The PageRank algorithm need not be confined to web pages. It could conceivably be let loose on any set of objects that have pairwise attachments. We note that the algorithm (1) continues to make sense when the w_{ij} are real-valued, that is, in the case of *weighted edges*, and, furthermore, the weights could also be negative. Examples that spring to mind where objects could be PageRanked according to their overall inuence are

- Mathematical articles: if paper i cites paper j then set $w_{ij} = 1$.
- Financial assets: if stock i moves up/down a short time after stock j moves up/down then set $w_{ij} = 1$.
- League tables: if Manchester City beat Manchester United 2-1 at home and 5-3 away, then set $w_{\text{united, city}} = (2+5) - (1+3)$, etc.

In this section, we show how PageRank can be used on a graph whose nodes are classes offered to undergraduate mathematics students at the University of Strathclyde. The 'link' information is provided by the pre-requisite structure between classes. We took data from

http://www.maths.strath.ac.uk/ungrad/index.html

in July 2003. (A pre-requisite was taken to be a specification on the class web page of either 'Essential' or 'Desirable'.) We looked at two interpretations.

Rank A: If class i is a pre-requisite for class j, then set $w_{ij} = 1$. Here, votes of confidence feed in to classes that use previously covered material, so PageRanking can be regarded as finding the 'hardest' or 'deepest' classes.

Rank B: If class i is a pre-requisite for class j, then set $w_{ji} = 1$. Here, votes of confidence are given to classes that get used in later classes, so PageRanking can be regarded as finding the most 'useful' or 'fundamental' classes.

Table 2: Mathematical classes subjected to PageRanking

Index	Code	Class
1	11 602	Introduction to Mechanics
2	11 606	Discrete Maths
3	11 611	IT Skills and Maths Software
4	11 612	Maths 1A
5	11 613	Maths 2A
6	11 700	Multivariable Calculus
7	11 710	Finite Dimensional Vector Spaces
8	11 711	Sequences and Series
9	11 712	Newtonian Mechanics
10	11 713	Numerical Analysis 2
11	11 721	Real Variable Calculus
12	11 722	Rigid Body Mechanics
13	11 725	Algebraic Structures
14	11 731	Differential Equations
15	11 741	Vector Calculus
16	11 801	Mathematical Analysis 2
17	11 802	Advanced Newtonian Mechanics
18	11 811	Mathematical Analysis 1
19	11 812	Advanced Rigid Body Mechanics
20	11 825	Abstract Algebra
21	11 835	Vector Spaces
22	11 841	Complex Analysis 1
23	11 843	Numerical Analysis 3
24	11 851	Complex Analysis 2
25	11 852	Fluid Mechanics 1
26	11 853	Numerical Analysis 4
27	11 861	Partial Differential Equations 1
28	11 862	Fluid Mechanics 2
29	11 871	Partial Differential Equations 2
30	11 891	Laplace Transforms and Linear Systems
31	11 921	Applications of Spectral Theory
32	11 922	Applied Functional Analysis
33	11 923	Continuum Mechanics 1
34	11 924	Continuum Mechanics 2
35	11 928	Numerical Solution of Initial Value PDEs
36	11 929	Modern Methods for Differential Equations
37	11 932	Numerical Approximation
38	11 937	One Dimensional Dynamical Systems
39	11 938	Higher Dimensional Dynamical Systems
40	11 939	Numerical Solution of Boundary Value PDEs
41	11 946	Waves
42	11 947	Coding Theory
43	11 948	Calculus of Variations
44	11 949	Mathematics of Financial Derivatives
45	11 951	Mathematics in Medicine

Table 2 lists the 45 classes that we considered. The class codes reveal the year of study; 11.6XX, 11.7XX. 11.8XX and 11.9XX

denoting years 1,2,3, and 4 respectively. To illustrate how the connectivity structure is defined, the class 11.891, Laplace Transforms and Linear Systems, which is number 30 on our list, is specified as a pre-requisite for the class 11.938, Higher Dimensional Dynamical Systems, which is number 39 on our list. Thus $w_{30,39} = 1$ for Rank A, and $w_{39,30} = 1$ for Rank B.

The PageRank results, normalized so that $\|\mathbf{r}\|_1 = 1$, are shown in Figure 5. Not surprisingly, final year classes, located at the right-hand end of Figure 5, score well when we use Rank A. With this viewpoint, importance tends to filter up through the years and the final year has the greatest opportunity to reap the benefit of incoming links. For what it is worth, Continuum Mechanics 2, our honours-level elasticity class, gets the highest Rank A score.

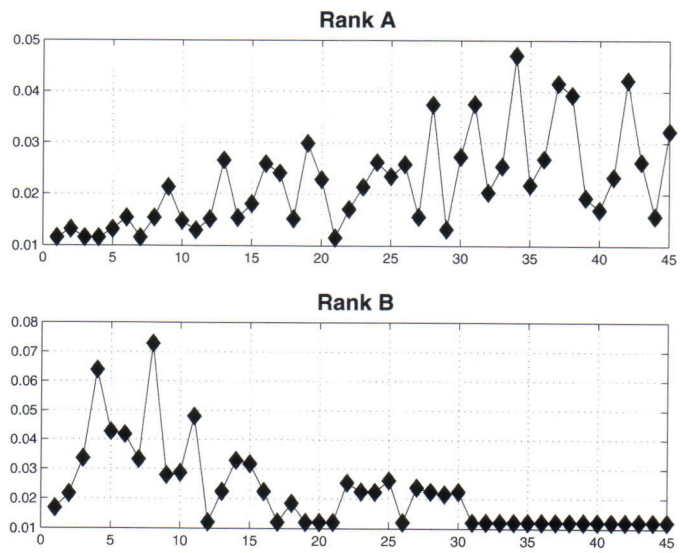


Figure 5: PageRank results for classes in Table 2

The picture reverses when Rank B is used. Here, the earlier year classes get credit for laying the foundations. However, the top 'usefulness' ranking goes not to a first year class but to the second year analysis course Sequences and Series. The epsilon-delta proof is vindicated.

We are not, of course, proposing PageRank in isolation as an infallible means to judge mathematics classes (especially since the two classes taught by one of the authors were deemed neither outstandingly useful nor deep!), rather we simply wish to show that the algorithm has wide applicability. Any serious implementation of PageRank would undoubtedly require some application-specific issues to be addressed (e.g., in our case, it may be prudent to treat compulsory classes in a special way), along with the 'dangling page' dilemma.

Discussion

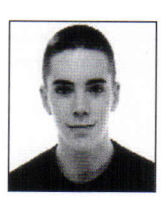
Reasonable estimates suggest that the web currently contains over 3×10^9 pages. This puts PageRanking in the big league of scientific computing [11]. However, while a user's query must be dealt with in real time, PageRanking is a behind-the-scenes operation. In practice, Google updates its archive about once a month, a period during which the 'Google Dance' takes place: type that phrase in to Google to learn more. As we saw in section 4, the PageRanking task boils down to a large, sparse, linear solve; a problem for which there are many rival methods. Identifying promising alternatives to Jacobi is a current research theme [9, 10]. Iterative methods are natural, since last month's rankings offer a good starting approximation for this month's, but issues such as the optimal choice of d and the effect of the terminating criterion must be clarified before concrete comparisons are to be made.

The original article [15] refers to an alternative, personalized, version of PageRank, where F in (4) becomes a more general rank-one matrix. Here, the teleporting jumps do not land uniformly across the web, but are biased towards a user's preferences. From teleporting it is but a tiny hop to *small world networks* [5, 6, 8, 12, 18], another fascinating area where graph theory has refused to stay in its box.❑

REFERENCES

1 Tara Calishain and Rael Dornfest (Editors). *Google Hacks: 100 Industrial-Strength Tips & Tools*. O'Reilly & Associates, 2003.
2 Anthony Cox. The war on the web. *The Guardian*, July 10, 2003. http://www.guardian.co.uk/online/story/0,3605,994676,00.html
3 Phil Craven. Google's PageRank explained: and how to make the most of it. Technical report, WebWorkshop Document, 2002. http://webworkshop.net/pagerank.html
4 James W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, 1997.
5 Paul Glendinning. View from the Pennines: Making connections. *Mathematics TODAY*, 39:26–27, 2003.
6 Peter Grindrod. Range-dependent random graphs and their application to modeling large small-world proteome datasets. Physical Review E, 66:066702, 2002.
7 Ben Hammersley. Making the web pay. *The Guardian*, August 7, 2003. http://www.guardian.co.uk/online/story/0,3605,1013313,00.html
8 Desmond J Higham. A matrix perturbation view of the small world phenomenon. *SIAM Journal on Matrix Analysis and Applications*, to appear,2003.
9 Sepandar D. Kamvar, Taher H. Haveliwala, Christopher D. Manning, and Gene H. Golub. Exploiting the block structure of the web for computing PageRank. Technical report, Stanford University (Preprint), 2003.
10 Sepandar D. Kamvar, Taher H. Haveliwala, Christopher D. Manning, and Gene H. Golub. Extrapolation methods for accelerating pagerank computations. *Proceedings of the Twelfth International World Wide Web Conference*, 2003.
11 Cleve Moler. The world's largest matrix computation. *MATLAB News and Notes*, October, 2002. http://www.mathworks.ch/company/newsletter/clevescorner/oct02_cleve.shtml
12 M. E. J. Newman. The stucture and function of complex networks. *SIAM Review*, 45:167–256, 2003.
13 J. R. Norris. *Markov Chains*. Cambridge University Press, 1997.
14 James M. Ortega. *Numerical Analysis: A Second Course*. SIAM, Philadelphia, 1990.
15 Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998. http://www-diglib.stanford.edu/diglib/pub/projectdir/google.html
16 Ian Rogers. Page rank explained. The Google Pagerank algorithm and how it works. Technical report, PR Computing Ltd, 2002. http://www.iprcom.com/papers/pagerank/
17 Sheldon M. Ross. *A First Course in Probability*. Prentice Hall, 5th edition, 1997.
18 D. J. Watts and S. H. Strogatz. Collective dynamics of `small-world' networks. *Nature*, 393:440–442, 1998.

Des Higham is a professor of mathematics at the University of Strathclyde. His research interests revolve around numerical analysis, especially stochastic computation. He is the author of a forthcoming undergraduate text on mathematical finance for Cambridge University Press and currently holds a Research Fellowship from the Leverhulme Trust. Sadly, his homepage is not ranked highest when 'Higham' is entered in to Google.



Alan Taylor is a third year undergraduate at the University of Strathclyde and is studying a joint degree in mathematics and computer science. He was awarded a Vacation Scholarship in June 2003 from the Carnegie Trust for the Universities of Scotland. Outside mathematics, his interests include squash, film and music. His primary areas of interest in mathematics are numerical analysis and differential equations.