# The PageRank Algorithm and Web Search Engines

John Lindsay Orr

University Of Nebraska – Lincoln

April 2010

jorr@math.unl.edu

# What is PageReank?



PageRank is an algorithm for ranking the importance of webpages.

It was developed in the late '90's by Larry Page and Sergey Brin, at that time grad students at Stranford.

# References

Brin and Page, The anatomy of a large-scale hypertextual web search engine, 1998

Page, Brin, Motwani, Rajeev, Winograd, The PageRank citation ranking, 1998

Bonato, A course on the web graph, AMS 2008

Bryan and Leise, The $25,000,000,000 eigenvector, SIAM Review 2006

# The job of a search engine



The job of a search engine is to receive queries and return a usable list of relevant matches, within in a reasonable time.

# The job of a search engine



The job of a search engine is to receive queries and return a usable list of relevant matches, within in a reasonable time.

# What is the web?

The web is a distributed, linked collection of documents.

# What is the web?

The web is a distributed, linked collection of documents.

This isn't as obvious as it sounds:

- HTML or other content types?
- Static or dynamic?
- HTTP(S) or other protocols?
- Public or restricted?

# The web is big

## But how big?

It's hard to tell how big, because estimates vary wildly and are constantly changing.

What counts as a web page: a URL, or the content returned? The "surface web" or the "deep web"?

Google (2008) claimed to have identified 1 trillion URLs, but they only index a fraction of those.

The size of the "indexed web" today is probably measured in the 10's of billions.

# The web is big
## Simple evidence

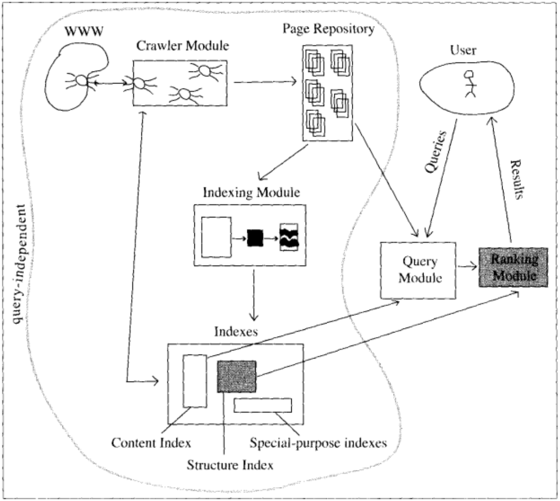
A Google query on `*a*` finds over 25 billion results.

A breadth-first search rooted at `http://www.math.unl.edu` found 21,000 internal pages. What percentage of UNL is the Math Dept? What percentage of the web is UNL? Surely

$$20,000 \times 50 \times 10,000 = 10^{10}$$

is a huge underestimate.

# How does a search engine work?



# The need for ranking



A Google query on "cat" found 591,000,000 results. A search for "PageRank" got 81,000,000.

1. Word/term frequency
2. Word/term context (`h1`, `h2`, `strong`, etc.)
3. Back-link counts

All very vulnerable to SEO spamming.

# Link analysis

PageRank – and other ranking algorithms, e.g., HITS – use global link analysis.

# PageRank: The goal

Let $W$ be the web-graph. Vertices are pages and there is a directed edge from $u$ to $v$ if a hyperlink, `<a href="...">cat</a>`, is found in $u$, pointing to $v$. (Ignore multiple links and loops.)

Let $n = |W|$ ($n \sim 10^{10}$).

Seek a single vector $r \in \mathbb{R}^n$, with

1. $r_i \geq 0$
2. $\|r\|_1 = 1$

(i.e., stochastic), where each $r_i$ represents the relative importance of page $v_i$.

# PageRank: The goal

Let $W$ be the web-graph. Vertices are pages and there is a directed edge from $u$ to $v$ if a hyperlink, `<a href="...">cat</a>`, is found in $u$, pointing to $v$. (Ignore multiple links and loops.)

Let $n = |W|$ $(n \sim 10^{10})$.

Seek a single vector $r \in \mathbb{R}^n$, with

1. $r_i \geq 0$
2. $\|r\|_1 = 1$

(i.e., stochastic), where each $r_i$ represents the relative importance of page $v_i$.

# What's important?

A page is important if a lot of important pages cite it.

# What's important?

A page is important if a lot of important pages cite it.

$$r_i = \sum_{v_j \to v_i} r_j$$

# What's important?

A page is important if a lot of important pages cite it.

$$r_i = \sum_{v_j \to v_i} r_j$$

$$r_i = \sum_{v_j \to v_i} \frac{1}{d_j^+} r_j$$

# What's important?

Let $A$ be the adjacency matrix of the directed graph $W$ (i.e., $a_{i,j} = 1$ if $v_i \to v_j$, otherwise zero).

Let $D = diag(d_1^+, \dots, d_n^+)$.

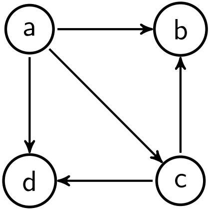
Let $A_0 = D^{-1}A$ (allowing for non-invertibility)

Then

$$r = rA_0$$

In other words, find an eigenvector (*the* eigenvector?) of $A_0$ for $\lambda = 1$.

# Example




$$A_0 = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

There are sure to be sinks in $W$.

If $W$ is a chain then

$$A_0 = \begin{bmatrix} 0 & 1 & 0 & \cdots & & & 0 \\ 0 & 0 & 1 & 0 & \cdots & & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & & \ddots & & \\ 0 & 0 & \cdots & & & & \end{bmatrix}$$

which is nilpotent and so $sp(A_0) = \{0\}$

I.e., solutions to $rA_0 = r$ do not exist.

$W$ is not strongly connected or even connected.

$$A_0 = \begin{bmatrix} A' & * \\ 0 & A'' \end{bmatrix}$$

The multiplicity of $\lambda = 1$ is greater than 1.

I.e., solutions to $rA_0 = r$ are not unique.

# Random surfer model

Imagine a (finite state, discrete time, time-homogenous) Markov Process on $W$.

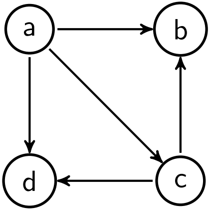
At each step the surfer clicks a link uniformly at random from the links on her current page.

If the page has no outlinks, pick a page uniformly at random from $W$. The transition probabilities for this process are

$$A_1 = A_0 + \frac{1}{n} z^T \mathbf{1}$$

where $z$ is the indicator vector for the sinks ($z_i = 1$ if $d_i^+ = 0$ and is 0 otherwise), and $\mathbf{1} = (1, 1, \ldots, 1)$.

# Example




$$A_1 = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 \end{bmatrix} + \frac{1}{4} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} [1, 1, 1, 1] = \begin{bmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}$$

# Random surfer model



The transition matrix

$$\begin{aligned} A_1 &= A_0 + \frac{1}{n} z^T \mathbf{1} \\ &= D^{-1} A + \frac{1}{n} z^T \mathbf{1} \end{aligned}$$

is a row-stochastic matrix.

## Random surfer model

The stationary distribution of the process is the long-term proportion of the time that the surfer will spend on each page.

If $p = (p_i)$ is the stationary distribution then

$$p = pA_1$$

and so we are still seeking an eigenvector for $\lambda = 1$, but now of our modified matrix, $A_1$.

# Stochastic matrices

**Lemma**

*If $S$ is a (row) stochastic matrix then $\lambda = 1$ is an eigenvalue.*

**Proof.**

$S\mathbf{1}^T = \mathbf{1}^T$. □

In other words, $\mathbf{1}^T$ is a right eigenvector, and so there must exist left eigenvectors too.

# Perron's Theorem

**Theorem**

*Let $P > 0$ and let $\rho$ be the spectral radius of $P$. Then...*

1. *...$\rho$ is positive and is an eigenvalue of $P$,*
2. *...$\rho$ has left and right eigenvectors with positive entries,*
3. *...$\rho$ has algebraic & geometric multiplicity 1, and*
4. *...all the other eigenvalues are less than $\rho$ in magnitude.*

**Proof.**

Find a fixed point of $Px/\|Px\|_1$ on $x_i \geq 0$, $\sum x_i = 1 \ldots$ $\square$

So if $P$ is a positive row-stochastic matrix, and $x$ is a positive left eigenvector for $\rho$, then

$$\|x\|_1 = x\mathbf{1}^T = x(P\mathbf{1}^T) = (xP)\mathbf{1}^T = \rho x\mathbf{1}^T = \rho\|x\|_1$$

and so

$$\rho = 1$$

# But there's still a problem...

Our transition matrix

$$A_1 = D^{-1}A + \frac{1}{n} z^T \mathbf{1}$$

isn't positive.

(If $A_1$ were irreducible we could use the Perron-Frobenius Theorem.)

It's the same issue as before; failure of (strong) connectedness.

# Adapt the random surfer model

Imagine now at each step that the random surfer either. . .

clicks a link uniformly at random from the links on her current page

. . . or else . . .

with probability $\alpha$ jumps to a new page chosen uniformly at random from $W$.

The probability $\alpha$ is called the teleportation constant.

# Adapt the random surfer model



The new transition matrix is

$$A_2 = (1-\alpha)(D^{-1}A + \frac{1}{n}z^T\mathbf{1}) + \alpha\frac{1}{n}\mathbf{1}^T\mathbf{1}$$

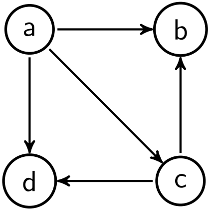
This is often called the Google Matrix.

Clearly this is positive, stochastic.

Brin & Page (1998) report using $\alpha = 0.15$ in early Google.

# Example






$$A_2 = \begin{bmatrix} 0.0375 & 0.3208 & 0.3208 & 0.3208 \\ 0.2500 & 0.2500 & 0.2500 & 0.2500 \\ 0.0375 & 0.4625 & 0.0375 & 0.4625 \\ 0.2500 & 0.2500 & 0.2500 & 0.2500 \end{bmatrix}$$

$$p = \begin{bmatrix} 0.1683 & 0.3078 & 0.2160 & 0.3078 \end{bmatrix}$$

# Computation: Computing the eigenvector
## Computational obstacles

We need to solve

$$pA_2 = p \qquad \text{or} \qquad p(A_2 - I) = 0$$

Gauss-Jordan elimination is $O(n^3)$, or $\sim 10^{30}$.

Moreover, it requires storage of the entire array, $O(n^2)$, or $\sim 10^{20}$ bytes (1 petabyte $\simeq 10^{12}$ bytes)

Let

$$p_0 = \frac{1}{n}\mathbf{1}$$
$$p_{k+1} = p_k A_2$$

so that $p_k = p_0 A_2^k$.

Since $p_k$ is a product of row stochastic matrices, it is row stochastic.

Thus, if $p_k$ converges, it converges to the normalized eigenvector (a.k.a., stationary distribution)

By Perron's Theorem, $A_2$ is similar to a block Jordan matrix

$$\begin{bmatrix} 1 & & & \\ & J_{\lambda_2}^{(m_2)} & & \\ & & J_{\lambda_3}^{(m_3)} & \\ & & & \ddots \end{bmatrix}$$

where the eigenvalues of $A_2$ are

$$1 > \lambda_2 > \lambda_3 > \cdots > \lambda_N$$

each with multiplicity $m_i$. (In particular, $m_1 = 1$.)

The powers of the Jordan blocks, $(J_{\lambda_i}^{(m_i)})^k$ converge to $0_{m_i \times m_i}$ and the rate of convergence is $O(\lambda_i^k)$.

Thus

1. $A_2^k$ converges to $\mathbf{1}^T p$
2. $p_k$ converges to $p$, (independent of $p_0$, in fact) and
3. the rate of convergence is $O(\lambda_2^k)$.

# Power method
## Complexity



$$\begin{aligned} p_{k+1} &= p_k A_2 \\ &= (1-\alpha) p_k D^{-1} A + \underbrace{\frac{1-\alpha}{n} p_k z^T \mathbf{1}}_{O(n)} + \underbrace{\frac{\alpha}{n} p_k \mathbf{1}^T \mathbf{1}}_{O(n)} \end{aligned}$$

Most pages can be expected to contain a bounded number of outlinks. Empirical studies suggest the average number of outlinks per page is around 10. Thus $A$ is sparce, and computing $p_k D^{-1} A$ is also $O(n)$.

Each iteration is $O(n)$ operations. All operations are matrix-vector and from the form of the vectors (diagonal, rank-1, and sparce) storage is also $O(n)$.

# Power method

## Rate of convergence

Brin & Page (1998) report that 52 iterations yield “reasonable tolerance” on a 322 million link database.

The following analysis casts light on the rapid convergence. . .

The PageRank Algorithm

John Orr

Introduction

PageRank

Computation

Further issues

**Theorem (Haveliwala & Kamavar, 2003)**

*If the eigenvalues of the stochastic matrix $A_1$ are*

$$\{1, \lambda_2, \lambda_3, \ldots, \lambda_n\}$$

*then the eigenvalues of*

$$A_2 = (1-\alpha)A_1 + \frac{\alpha}{n}\mathbf{1}^T\mathbf{1}$$

*are*

$$\{1, (1-\alpha)\lambda_2, (1-\alpha)\lambda_3, \ldots, (1-\alpha)\lambda_n\}$$

**Corollary**

*The power method computation of the PageRank vector converges $O((1-\alpha)^k)$.*

Proof (Langeville & Meyer, 2005)

Observe

$$A_1\mathbf{1}^T = \mathbf{1}^T \text{ and } \frac{1}{n}(\mathbf{1}^T\mathbf{1})\mathbf{1}^T = \mathbf{1}^T$$

and so, wrt a basis that starts with $\mathbf{1}$,

$$\begin{aligned} A_2 &= (1-\alpha)A_1 + \frac{\alpha}{n}\mathbf{1}^T\mathbf{1} \\ &= (1-\alpha)\begin{bmatrix} 1 & * \\ 0 & B \end{bmatrix} + \alpha\begin{bmatrix} 1 & * \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 1 & * \\ 0 & (1-\alpha)B \end{bmatrix} \end{aligned}$$

# Stability

The web is constantly changing, and so rankings are not useful unless they are stable under small perturbations of $W$.

**Theorem (Ng, Zheng, Jordan 2001)**

*Let $G$ be the PageRank matrix defined on a directed graph $W$ and let $p$ be its stationary distribution. Suppose $W'$ is obtained by changing the outlinks of vertices $i_1, i_2, \ldots, i_k$, and let $G'$ and $p'$ be the corresponding perturbations of $G$ and $p$. Then*

$$\|p' - p\|_1 \leq \frac{2\sum_{j=1}^{k} p_{i_j}}{\alpha}$$

# Variants of PageRank

"Intelligent surfer" transition matrix, $A_1'$ with values computed from server logs.

"Personalized teleportation vector", $v$, gives

$$(1-\alpha)A_1' + \frac{\alpha}{n}\mathbf{1}^T v$$

The complexity of the calculation makes genuinely personalized vectors impractical.