Google PageRank with Stochastic Matrix

Md. Shariq

Puranjit Sanyal

Samik Mitra

M.Sc. Applications of Mathematics (I Year)

Chennai Mathematical Institute

PROJECT PROPOSAL

Group Members: Md Shariq, Samik Mitra, Puranjit Sanyal
Title: Google Page Rank using Markov Chains.

Introduction

Whenever we need to know about something the first thing that comes to our mind is Google! Its obvious for a mathematics student to wonder how the pages are ordered after a search. We look at how the pages were ranked by an algorithm developed by Larry Page(Stanford University) and Sergey Brin(Stanford University) in 1998.

In this project we consider a finite number of pages and try to rank them. Once a term is searched, the pages containing the term are ordered according to the ranks.

Motivation

We have many search engines, but Google has been the leader for a long time now. Its strange how it uses Markov Chains and methods from Linear Algebra in ranking the pages.

Project Details

We try to answer how the pages are ranked. We encounter a Stochastic Matrix for which we need to find the eigen vector corresponding to the eigen value 1, and for this we use QR method for solving eigen values and eigen vectors. This can also be achieved by taking the powers of the matrix obtained. And we analyze these two methods. Scilab will be extensively used througout.

In the process we might face hurdles like, if we view calculating eigen vector as solving the linear system, the matrix we arrive at might not be invertible, and hence Gaussian elimination cannot be applied. Even in calculation of eigen vectors the complexity is quite high.

References

- Ciarlet, Philippe G. 1989, *Introduction to Numerical Linear Algebra and Optimisation*, First Edition, University Press, Cambridge.
- *Markov Chain*, <http://en.wikipedia.org/wiki/Markov_chain>
- *Examples of Markov Chains*, <http://en.wikipedia.org/wiki/Examples_of_Markov_chains>

Progress Report I

In order to explore the application of Markov Chain in deciding the page rank, we went through Markov Chains initially.

Definition: *Markov Chain* is a random process described by a physical system which at any given time $t = 1, 2, 3, ...$ occupies one of a finite no of states. At each time t the system moves from state i to state j with probability p_{ij} that does not depend on time t. The quantities p_{ij} are called transition probabilities ie., the next state of the system depends only on the current state and not on any prior states.

A transition Matrix T of a Markov Chain is an $n \times n$ matrix (where n represents the no. of states of the system) and the corresponding entries t_{ij}'s are the transition probabilities ($0 \leq t_{ij} \leq 1$ for all $i, j = 1, 2, , n$)

The state, the system currently in, is represented by the $n \times 1$ matrix: $q^k = (q_1, q_2, \ldots, q_n)'$ called the *state vector*. The initial state of the system is called *initial state vector*.

If the information of the initial state of the system is not known,but the probability of it being in a certain state is, we use an initial probability vector q, where

1. $0 \leq q_i \leq 1$
2. $q_1 + q_2 + ... + q_n = 1$

since the system must be any one of the n states at any given time .

Given an initial probability vector, the k^{th} step probability vector is $q^k = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix}$ where ${q_i}^k$ is the probability of being in state i after k steps.

When dealing with Markov Chains we often are interested in what is happening to the system in the long run or as $k \longrightarrow \infty$.As $k \longrightarrow \infty$, q_k will approach a limiting vector s called a steady state vector .

A matrix A is *regular* when for some positive k all entries A^k are all positive. It can be shown that if the transition matrix is regular then it has a

steady state vector. It can also be shown steady state vector is the eigen vector of the transition matrix corresponding to the eigen value 1.

After describing what Markov Chains are the next question which arises immediately is how this can be used by Google in assigning page ranks.

The importance of a page is decided by the no. of links to and from that page. The relative importance of a page is determined by the no. of inlinks to that page and moreover inlinks from more important pages bear more weight than that from less important pages. Also this weight is distributed proportionately if a particular page carries multiple outlinks.

Page Rank is formally defined by where $r_i = \sum_{j \in l_i} \frac{r_j}{|O_j|}$ where r_i denotes the rank of the page j, l_i the set of pages that have inlinks to i and $|O_j|$ is the no. of pages that have outlinks from page j. An initial rank of $r_i(0) = 1/n$ where n is the total no of pages on the web . The Page rank iterates the ${r_i}^{k+1} = \sum_{j \in l_i} \frac{{r_j}^k}{|O_j|}$ for k=0,1,2,.. and ${r_i}^k$ is the Page rank of page i at the k^{th} iteration. The whole process can be represented by using matrices if q^k be the page rank vector at the k^{th} iteration then $q^{k+1} = T.q^k$ where T is the transition matrix.

If the no of outlinks from page i be O_i and it is equally likely that any outlink can be chosen then

$$t_{ij} = \begin{cases} \frac{1}{|O_j|} & \text{if there is a link from i to j} \\ 0 & \text{otherwise} \end{cases}$$

Henceforth, all of the above mentioned facts can be clearly explained by creating a finite node sample web where we will imagine the world wide web to be a directed graph i.e. a finite set of nodes and a set of ordered pairs of nodes representing directed edges between nodes. Each web page is a node and each hyperlink is a directed edge.

The difficulty which may arise is that a particular page may have no outlinks at all and so the corresponding row of the transition will have all entries as 0. Also, it is not guaranteed that the Markov model corresponding to every stochastic matrix will converge. In coming days we would explore how to circumnavigate these problems.

Progress Report II

Following concepts have been defined, which would be relevant in the build up:

1. discrete time Markov chain
2. column-stochastic matrix
3. essential and inessential states
4. irreducible stochastic matrix
5. irreducible Markov chain
6. spectral radius of a matrix
7. $\infty - norm$ of a vector and a matrix
8. period of a state in a Markov chain
9. aperiodicity of a Markov chain
10. steady-state distribution

In building the theory to obtain a PageRank, following were defined

1. hyperlink matrix
2. dangling node
3. Google matrix

Also proofs of the following have been provided:

1. A stochastic matrix P always has 1 as one of its eigenvalues.
2. If P is a $n \times n$ column-stochastic matrix, then $\|P\| = 1$.
3. If P is a column-stochastic matrix, then $\rho(P) = 1$.
4. *(Theorem:(Perron, 1907; Frobenius, 1912)):* If P is a column-stochastic matrix and P be irreducible, in the sense that $p_{ij} > 0 \quad \forall\ i, j \in S$, then 1 is a simple eigenvalue of P. Moreover, the unique eigenvector can be chosen to be the probability vector $\mathbf{w}$ which satisfies $\lim_{t\to\infty} P^{(t)} = [\mathbf{w}, \mathbf{w}, \ldots, \mathbf{w}]$. Furthermore, for any probability vector $\mathbf{q}$ we have $\lim_{t\to\infty} P^{(t)}\mathbf{q} = \mathbf{w}$.

Then, we considered web pages as the states of a Markov chain and the corresponding stochastic matrix was defined to be hyperlink matrix. In a step by step process, a counter example was shown, where the matrix cannot provide a reasonable estimate of PageRank and hence it was modified to a better one.

Finally we arrived at the Google matrix, which satisfies the conditions of the Perron-Frobenious theorem (proof given), and its eigenvector corresponding to the eigenvalue 1 gives us the PageRank.

Power method for calculating the eigenvector was used, since we need eigenvector corresponding to eigenvalue 1, which is the spectral radius. And then a python code is built to calculate the eigenvector (doing upto 100 iterations, which is believed to be sufficient!).

Things yet to be done:

1. organizing the document.
2. making a beamer presentation of it.
3. mentioning the references.
4. (if possible) including the proofs of statements made, but not proved.

Google PageRank with Stochastic Matrix

Md. Shariq, Puranjit Sanyal, Samik Mitra
(M.Sc. Applications of Mathematics)

November 15, 2012

Discrete Time Markov Chain

Let S be a countable set (usually S is a subset of $\mathbb{Z}$ or $\mathbb{Z}^d$ or $\mathbb{R}$ or $\mathbb{R}^d$). Let $\{X_0, X_1, X_2, \ldots\}$ be a sequence of random variables on a probability space taking values in S. Then $\{X_n : n = 0, 1, 2, \ldots\}$ is called a *Markov Chain* with state space S if for any $n \in \mathbb{Z}^{\geq 0}$, any $j_0, j_1, \ldots, j_{n-1} \in S$, any $i, j \in S$ one has

$$\Pr(X_{n+1} = i \mid X_0 = j_0, X_1 = j_1, \ldots, X_n = j) = \Pr(X_{n+1} = i \mid X_n = j).$$

In addition, if $\Pr(X_{n+1} = i \mid X_n = j) = \Pr(X_1 = i \mid X_0 = j)\ \forall\ i, j \in S$ and $n \in \mathbb{Z}^{\geq 0}$ then we say $\{X_n : n \in \mathbb{Z}^{\geq 0}\}$ is a time homogeneous Markov Chain.

Notation: We denote time homogeneous Markov Chain by MC.
Note: The set S is called state space and its elements are called states.

Column-Stochastic Matrix

A *column-stochastic matrix* (or column-transition probability matrix) is a square matrix $P = ((p_{ij}))_{i,j \in S}$ (where S may be a finite or countably infinite set) satisfying:

(i) $p_{ij} \geq 0$ for any $i, j \in S$

(ii) $\sum_{i \in S} p_{ij} = 1$ for any $j \in S$

Similarly, *row-stochastic matrix* can be defined considering $\sum_{j \in S} p_{ij} = 1$ for any $i \in S$.

Consider the MC, $\{X_n : n \in S\}$ on the state space S. Let

$$p_{ij} = \Pr(X_1 = i \mid X_0 = j) \ \ \forall \ \ i, j \in S.$$

Then $P = ((p_{ij}))_{i,j \in S}$ is the column-stochastic matrix. We call P as the stochastic matrix of MC, $\{X_n : n \in S\}$.

Lemma: If A is a $n \times n$ matrix whose rows(or columns) are linearly dependent, then $det(A) = 0$.

Proof:
Let $r_1, r_2, \ldots, r_n$ be the rows of A.
Given, $r_1, r_2, \ldots, r_n$ are dependent, hence

$$\exists\ \alpha_1, \alpha_2, \ldots, \alpha_n \ni \prod_{i=1}^{n} \alpha_i \neq 0 \text{ and } \sum_{i=1}^{n} \alpha_i r_i = \mathbf{0}$$

Consider a matrix A' with rows as $\begin{bmatrix} \alpha_1 r_1 \\ \alpha_2 r_2 \\ \vdots \\ \alpha_n r_n \end{bmatrix}$.

Now, $det(A') = det(A) \prod_{i=1}^{n} \alpha_i$.

$$det(A') = \begin{vmatrix} \alpha_1 r_1 \\ \alpha_2 r_2 \\ \vdots \\ \alpha_n r_n \end{vmatrix} = \begin{vmatrix} \sum_{i=1}^{n} \alpha_i r_i \\ \alpha_2 r_2 \\ \vdots \\ \alpha_n r_n \end{vmatrix} = \begin{vmatrix} \mathbf{0} \\ \alpha_2 r_2 \\ \vdots \\ \alpha_n r_n \end{vmatrix}$$

$\therefore det(A') = 0$ and hence $det(A) = 0$ $\ (\because \prod_{i=1}^{n} \alpha_i \neq 0)$. $\square$

Theorem: A stochastic matrix P always has 1 as one of its eigenvalues.

Proof:
Let $S = \{1, 2, \ldots, n\}$ and $P = ((p_{ij}))_{1 \leq i,j \leq n}$.
Consider the identity matrix I_n,

$$I_n = ((\delta_{ij}))_{1 \leq i,j \leq n} \text{ where } \delta_{ij} \text{ is Kronecker delta.}$$

$$\sum_{i=1}^{n} p_{ij} = 1 \text{ and } \sum_{i=1}^{n} \delta_{ij} = 1$$

$$\sum_{i=1}^{n} (p_{ij} - I_{ij}) = 0 \ \ \forall \ \ 1 \leq j \leq n$$

Consequently, the rows of $P - I_n$ are not linearly independent and hence $det(P - I_n) = 0$ (by the above lemma). $\therefore P$ has 1 as its eigenvalue. $\square$

Definition: $P^{(n)} = ((p_{ij}^{(n)}))_{i,j \in S}$ where $p_{ij}^{(n)} = \Pr(X_n = i \mid X_0 = j)$, $i, j \in S$.
A little work and we can see that $P^{(n)} = P^n \ \ \forall \ n \in \mathbb{Z}^{\geq 1}$.
Also $P^{(n)}$ is a column-stochastic matrix as

$$\sum_{i \in S} \Pr(X_n = i \mid X_0 = j) = 1$$

Classification of states of a Markov Chain

Definition 1: $j \longrightarrow i$ (read as i is accessible from j or the process can go from j to i) if $p_{ij}{}^{(n)} > 0$ for some $n \in \mathbb{Z}^{\geq 1}$.
Note: $j \longrightarrow i \iff \exists \ \ n \in \mathbb{Z}^{\geq 1}$ and $j = j_0, j_1, j_2, \ldots, j_{n-1} \in S$ such that $p_{jj_1} > 0, \ p_{j_1 j_2} > 0, \ p_{j_2 j_3} > 0, \ldots, \ p_{j_{n-2} j_{n-1}} > 0, \ p_{j_{n-1} i} > 0$.

Definition 2: $i \longleftrightarrow j$ (read as i and j communicate) if $i \longrightarrow j$ and $j \longrightarrow i$.

Essential and Inessential States
i is an *essential state* if $\forall \ j \in S \ni i \longrightarrow j$, then $j \longrightarrow i$ (ie., if any state j is accessible from i, then i is accessible from j).
States that are not essential are called *inessential states*.

Let ξ be set of all essential states.
For $i \in \xi$, let $\xi(i) = \{j : i \longrightarrow j\}$ where $\xi(i)$ is the essential class of i. Then $\xi(i_0) = \xi(j_0)$ iff $j_0 \in \xi(i_0)$ (ie., $\xi(i) \bigcap \xi(k) = \phi$ iff $k \notin \xi(i)$).

Definition: A stochastic matrix P having one essential class and no inessential states (ie., $S = \xi = \xi(i) \ \forall \ i \in S$) is called *irreducible*, and the corresponding MC is called *irreducible*.

Let A be a $n \times n$ matrix.

• The *spectral radius* of a $n \times n$ matrix, $\rho(A)$ is defined as

$$\rho(A) = \max_{1 \leq i \leq n} \ \{ \ |\lambda_i| : \ \lambda_i \text{ is an eigenvalue of A} \}$$

- $\infty - norm$ of a vector $\mathbf{x}$ is defined as $\|\mathbf{x}\|_\infty = \max\limits_{\mathbf{1\le i\le n}} |\mathbf{x_i}|$
- $\infty - norm$ of a matrix $\mathbf{A}$ is defined as $\|\mathbf{A}\|_\infty = \max\limits_{\mathbf{1\le i\le n}} (\sum\limits_{\mathbf{j=1}}^{\mathbf{n}} |\mathbf{a_{ij}}|)$.
- Also $\|A\|_2 = \sqrt{\rho(A^*A)} = \sqrt{\rho(AA^*)} = \|A^*\|_2$.
- If V is a finite dimensional vector space, then all norms on V are equivalent.

$$\therefore \quad \|A\|_\infty = \|A\|_2 = \|A^*\|_2 = \|A^*\|_\infty$$

Lemma: If P is a $n \times n$ column-stochastic matrix, then $\|P\| = 1$.

Proof:
If P is column-stochastic, then P' is row-stochastic (ie., $\sum\limits_{i=1}^{n} p_{ji} = 1$).
We know that

$$\|P'\|_\infty = \max_{1\le j\le n} (\sum_{i=1}^{n} |p_{ij}|)$$

$\because P'$ is stochastic

$$\|P'\|_\infty = \max_{1\le j\le n} (\sum_{i=1}^{n} p_{ij})$$

$$\|P'\|_\infty = \max_{1\le j\le n} 1$$

$$\|P'\|_\infty = 1$$

$$\|P\|_\infty = 1$$

Also we know that if V is any finite dimensional vector space, then all norms on V are equivalent. $\therefore \|P\| = 1$ □

Theorem: If P is a stochastic matrix, then $\rho(P) = 1$.

Proof:
Let λ_i be an eigenvalue of $P \ \ \forall\ 1 \le i \le n$.
Then it is also an eigenvalue for P'.
Let x_i be an eigenvector corresponding to the eigenvalue λ_i of P'.

$$P'x_i = \lambda_i x_i$$

$$\|\lambda_i x_i\| = |\lambda_i|\|x_i\| = \|P'x_i\| \le \|P'\|\|x_i\|$$

$$\Rightarrow |\lambda_i| \|x_i\| \leq \|x_i\|$$
$$\Rightarrow |\lambda_i| \leq 1$$

Also we have proved that 1 is always an eigenvalue of P, hence $\rho(P) = 1$. $\square$

Definition: Let $i \in \xi$. Let $A = \{n \geq 1 : p_{ii}^{(n)} > 0\}$. $A \neq \phi$ and the greatest common divisor(gcd) of A is called the *period* of state i.

If $i \longleftrightarrow j$, then i and j have same period. In particular, period is constant on each equivalence class of essential states. If a MC is irreducible, then we can define period for the corresponding stochastic matrix since all the sates are essential.

Definition: Let d be the period of the irreducible Markov chain. The Markov chain is called *aperiodic* if $d = 1$.

• If $\mathbf{q} = (q_1, q_2, \ldots, q_n)'$ is a probability distribution for the states of the Markov chain at a given iterate with $q_i \geq 0$ *and* $\sum_{i=1}^{n} q_i = 1$, then

$$P\mathbf{q} = (\sum_{j=1}^{n} P_{1j}q_j, \sum_{j=1}^{n} P_{2j}q_j, \ldots, \sum_{j=1}^{n} P_{nj}q_j)'$$

is again a probability distribution for the states at the next iterate.

• A probability distribution $\mathbf{w}$ is said to be a *steady-state distribution* if it is invariant under the transition, i.e. $P\mathbf{w} = \mathbf{w}$. Such a distribution must be an eigenvector of P corresponding to the eigenvalue 1.

The existence as well as the uniqueness of the steady-state distribution is guaranteed for a class of Markov chains by the following theorem due to Perron and Frobenius.

Theorem:(Perron, 1907; Frobenius, 1912) If P is a stochastic matrix and P be irreducible, in the sense that $p_{ij} > 0 \ \ \forall\, i, j \in S$, then 1 is a simple eigenvalue of P. Moreover, the unique eigenvector can be chosen to be the probability vector $\mathbf{w}$ which satisfies $\lim_{t\to\infty} P^{(t)} = [\mathbf{w}, \mathbf{w}, \ldots, \mathbf{w}]$. Furthermore, for any probability vector $\mathbf{q}$ we have $\lim_{t\to\infty} P^{(t)}\mathbf{q} = \mathbf{w}$.

Proof:
Claim: $\lim_{t\to\infty} p_{ij}^{(t)} = w_i$
Proof:
$\because P = ((p_{ij}))_{i,j\in S} \ni p_{ij} > 0 \ \ \forall\ i,j \in S$ we have, $\delta = \min_{i,j\in S} p_{ij} > 0$

$$(P^{(t+1)})_{ij} = (P^{(t)}P)_{ij}$$

$$p_{ij}^{(t+1)} = \sum_{k\in S} p_{ik}^{(t)} p_{kj}$$

Let $m_i^{(t)} = \min_{j\in S} p_{ij}^{(t)}$ and $M_i^{(t)} = \max_{j\in S} p_{ij}^{(t)}$

$$0 < m_i^{(t)} \leq M_i^{(t)} < 1$$

Now,

$$m_i^{(t+1)} = \min_{j\in S} \sum_{k\in S} p_{ik}^{(t)} p_{kj} \geq m_i^{(t)} \sum_{k\in S} p_{kj} = m_i^{(t)}$$

$\therefore$ the sequence $(m_i^{(t)})$ is non-decreasing.
Also,

$$M_i^{(t+1)} = \max_{j\in S} \sum_{k\in S} p_{ik}^{(t)} p_{kj} \leq M_i^{(t)} \sum_{k\in S} p_{kj} = M_i^{(t)}$$

$\therefore$ the sequence $(M_i^{(t)})$ is non-increasing.
Hence, $\lim_{t\to\infty} m_i^{(t)} = m_i \leq M_i = \lim_{t\to\infty} M_i^{(t)}$ exist.

We now try to prove that $m_i = M_i$.

Consider $M_i^{(t+1)} - m_i^{(t+1)}$

$$= \max_{j\in S} \sum_{k\in S} p_{ik}^{(t)} p_{kj} - \min_{l\in S} \sum_{k\in S} p_{ik}^{(t)} p_{kl}$$

$$= \max_{j,l\in S} \sum_{k\in S} p_{ik}^{(t)} (p_{kj} - p_{kl})$$

$$= \max_{j,l\in S} \ [\sum_{k\in S} p_{ik}^{(t)} (p_{kj} - p_{kl})^+ + \sum_{k\in S} p_{ik}^{(t)} (p_{kj} - p_{kl})^-]$$

$$\leq \max_{j,l\in S} \ [M_i^{(t)} \sum_{k\in S} (p_{kj} - p_{kl})^+ + m_i^{(t)} \sum_{k\in S} (p_{kj} - p_{kl})^-]$$

where $\sum_{k\in S}(p_{kj}-p_{kl})^+$ means the summation of only the positive terms ($p_{kj}-p_{kl}>0$) and similarly $\sum_{k\in S}(p_{kj}-p_{kl})^-$ means the summation of only the negative terms ($p_{kj}-p_{kl}<0$).

Let $\sum_{k\in S}^{+}(p_{kj}-p_{kl})=\sum_{k\in S}(p_{kj}-p_{kl})^+$ and $\sum_{k\in S}^{-}(p_{kj}-p_{kl})=\sum_{k\in S}(p_{kj}-p_{kl})^-$
Consider $\sum_{k\in S}(p_{kj}-p_{kl})^-$

$$= \sum_{k\in S}^{-}(p_{kj}-p_{kl})$$

$$= \sum_{k\in S}^{-}p_{kj} - \sum_{k\in S}^{-}p_{kl}$$

$$= 1 - \sum_{k\in S}^{+}p_{kj} - (1 - \sum_{k\in S}^{+}p_{kl})$$

$$= \sum_{k\in S}^{+}(p_{kl}-p_{kj})$$

$$= -\sum_{k\in S}(p_{kj}-p_{kl})^+$$

$$\therefore\ M_i^{(t+1)} - m_i^{(t+1)} \le (M_i^{(t)} - m_i^{(t)}) \max_{j,l\in S} \sum_{k\in S}(p_{kj}-p_{kl})^+.$$

If $\max_{j,l\in S} \sum_{k\in S}(p_{kj}-p_{kl})^+ = 0$, then $M_i^{(t)} = m_i^{(t)}$.

If $\max_{j,l\in S} \sum_{k\in S}(p_{kj}-p_{kl})^+ \neq 0$, for the pair j,l that gives the maximum, let r be the number of terms in $k\in S$ for which $p_{kj}-p_{kl}>0$, and s be the number of terms for which $p_{kj}-p_{kl}<0$. Then, $r\ge 1$ and $\tilde{n}=r+s\ge 1$ as well as $\tilde{n}\le n$.
ie.,

$$\sum_{k\in S}(p_{kj}-p_{kl})^+ = \sum_{k\in S}^{+}p_{kj} - \sum_{k\in S}^{+}p_{kl}$$

$$= 1 - \sum_{k\in S}^{-}p_{kj} - \sum_{k\in S}^{+}p_{kl}$$

$$\le 1 - s\delta - r\delta = 1 - \tilde{n}\delta$$

$$\le 1 - \delta < 1.$$

Hence the estimate of $M_i^{(t+1)} - m_i^{(t+1)}$ is

$$M_i^{(t+1)} - m_i^{(t+1)} \leq (1-\delta)(M_i^{(t)} - m_i^{(t)}) \leq (1-\delta)^t (M_i^{(1)} - m_i^{(1)}) \to 0$$

as $t \to \infty$.

$$\therefore M_i = m_i$$

Let $w_i = M_i = m_i$. But,

$$m_i^{(t)} \leq p_{ij}^{(t)} \leq M_i^{(t)} \Rightarrow \lim_{t\to\infty} p_{ij}^{(t)} = w_i \quad \forall\ j \in S$$

$$\lim_{t\to\infty} P^{(t)} = [\mathbf{w}, \mathbf{w}, \ldots, \mathbf{w}]$$

$$\lim_{t\to\infty} P^{(t)} = [\mathbf{w}, \mathbf{w}, \ldots, \mathbf{w}] = P \lim_{t\to\infty} P^{(t-1)} = P[\mathbf{w}, \mathbf{w}, \ldots, \mathbf{w}] = [P\mathbf{w}, P\mathbf{w}, \ldots, P\mathbf{w}]$$

Hence, $\mathbf{w}$ is the eigenvector corresponding to the eigenvalue $\lambda = 1$.

Let $\mathbf{x}(\neq 0)$ be an eigenvector corresponding to the eigenvalue $\lambda = 1$.

$$\Rightarrow P\mathbf{x} = \mathbf{x} \qquad \Rightarrow P^{(t)}\mathbf{x} = \mathbf{x}$$

$$\lim_{t\to\infty} P^{(t)}\mathbf{x} = [\mathbf{w}, \mathbf{w}, \ldots, \mathbf{w}]\mathbf{x} = (w_1(\sum_{i\in S} x_i), w_2(\sum_{i\in S} x_i), \ldots, w_n(\sum_{i\in S} x_i))' = (\sum_{i\in S} x_i)\mathbf{w}.$$

But, $\lim_{t\to\infty} P^{(t)}\mathbf{x} = \mathbf{x}$

$$\Rightarrow\ \mathbf{x} = (\sum_{i\in S} x_i)\mathbf{w} \qquad (\sum_{i\in S} x_i \neq 0 \quad \because \mathbf{x} \neq 0)$$

Hence, eigenvector corresponding to eigenvalue 1 is unique upto a constant multiple.
Finally, for any probability vector $\mathbf{q}$, the above result shows that

$$\lim_{t\to\infty} P^{(t)}\mathbf{q} = (w_1(\sum_{i\in S} q_i), w_2(\sum_{i\in S} q_i), \ldots, w_n(\sum_{i\in S} q_i))' = \mathbf{w}.$$

□

Let $\mathbf{q}$ be a probability distribution vector. Define

$$\mathbf{q}^{(i+1)} = P\mathbf{q}^{(i)} \ \ \forall\ \ i \in \mathbb{Z}^{\geq 0}\ where\ \mathbf{q}^{(0)} = \mathbf{q}$$

$$\therefore\ \ \mathbf{q}^{(t)} = P^{(t)}\mathbf{q}^{(0)} = P^{(t)}\mathbf{q}\ \ \forall\ \ t \in \mathbb{Z}^{\geq 1}$$

From the above theorem

$$\lim_{t\to\infty} P^{(t)}\mathbf{q} = \mathbf{w}\ \Rightarrow\ \lim_{t\to\infty} \mathbf{q}^{(t)} = \mathbf{w}$$

Google Page Rank

There are approximately 45.3 *billion* web pages according to the website *www.worldwidewebsize.com*. Now it's not absurd to believe that some information you might need, exists in atleast one of the 45.3 *billion* web pages. One would think of organizing these web pages, otherwise its like searching for a document/book in a huge unorganized library with no librarians.

This organizing and finding is done by search engines, of course there are many, but Google is the pioneer. In this article we will look into how Google organizes the web world.

Most search engines, including Google, continually run an army of computer programs that retrieve pages from the web, index the words in each document, and store this information in an efficient format. Each time a user asks for a web search using a search phrase, such as *abc xyz*, the search engine determines all the pages on the web that contains the words in the search phrase (perhaps additional information such as the space between the words *abc* and *xyz* will be noted as well) and then displays those pages in a particular indexed way. Google claims to index 25 billion pages as per March 2011. The problem is: Roughly 95% of the text in web pages is composed from a mere 10,000 words. This means that, for most searches, there will be a huge number of pages containing the words in the search phrase. We need to sort these pages such that important pages are at the top of the list.

Google feels that the value of its service is largely in its ability to provide unbiased results to search queries and asserts that, "the heart of our software is PageRank." As we'll see, the trick to sorting or ranking is to ask the web itself to rank the importance of pages.

The outline is, when a user gives an input for the search, Google gets hold to all the pages that conatin the search input. And now, in the search result, these pages are displayed in the order of their ranking.

History:

PageRank was developed at Stanford University by Larry Page (hence the name PageRank) and Sergey Brin in 1996 as part of a research project about a new kind of search engine. Sergey Brin had the idea that information on the web could be ordered in a hierarchy by "link popularity": a page is ranked higher as there are more links to it. It was co-authored by

Rajeev Motwani and Terry Winograd. The first paper about the project, describing PageRank and the initial prototype of the Google search engine, was published in 1998 shortly after, Page and Brin founded Google Inc., the company behind the Google search engine. While just one of many factors that determine the ranking of Google search results, PageRank continues to provide the basis for all of Google's web search tools.

PageRank has been influenced by citation analysis, early developed by Eugene Garfield in the 1950s at the University of Pennsylvania, and by Hyper Search, developed by Massimo Marchiori at the University of Padua. In the same year PageRank was introduced (1998), Jon Kleinberg published his important work on HITS. Google's founders cite Garfield, Marchiori, and Kleinberg in their original paper.

Generating importance of pages:

A web page generally has links to other pages that contain valuable, relaible information related to (or may be not) to the web page. This tells us that, the web pages to which there are links in a particular web page are of considerable importance. It is said that Google assigns importance to all the web pages each month.

The importance of a page is judged by the number of pages linking to it as well as the importance of the linked pages. Let $I(P)$ be the measure of importance for each web page P, let it be called the *PageRank*. At various web sites, we may find an approximation of a page's PageRank. (For instance, the home page of The American Mathematical Society currently has a PageRank of 8 on a scale of 10). This reported value is only an approximation since Google declines to publish actual PageRanks.

Suppose that a page P_j has l_j links. If one of those links is to page P_i , then P_j will pass on $\frac{1}{l_j}$ of its importance to P_i. Let the set of all the pages linking to P_i be denoted by B_i.

Hence the PageRank of P_i is given by

$$I(P_i) = \sum_{P_j \in B_i} \frac{I(P_j)}{l_j}$$

This looks wierd, because determining the PageRank of a page involves the PageRank of the pages linking to it. Is it the chicken or the egg?

We now formulate it into a more mathematically familiar problem.

Consider a matrix $H = ((H_{ij}))$ called the *hyperlink matrix* where

$$H_{ij} = \begin{cases} \frac{1}{l_j} & \text{if } P_j \in B_i \\ 0 & \text{otherwise} \end{cases}$$

Note : H is a column-stochastic matrix

$$\because \quad \sum_i H_{ij} = 1$$

Also define $I = [I(P_i)]$, then the equation of page rank can be written as

$$I = HI \ .$$

The vector I is the eigenvector corresponding to the eigenvalue 1 of the matrix H.

Consider the web shown in the figure $A1$.

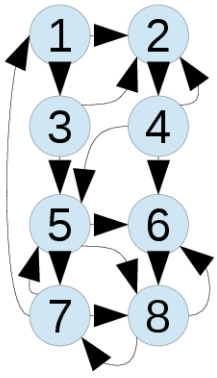

Figure - A1

It is a collection of 8 web pages with the links shown by arrows. The hyperlink

matrix for this web is

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 1 & \frac{1}{3} & 0 \end{bmatrix} \quad \text{with} \quad I = \begin{bmatrix} 0.0600 \\ 0.0675 \\ 0.0300 \\ 0.0675 \\ 0.0975 \\ 0.2025 \\ 0.1800 \\ 0.2950 \end{bmatrix}$$

Hence, page 8 is most popular, followed by page 6 and so on.

Calculating the eigenvector I

There are different ways of calculating the eigenvectors. But the challenge here is that the hyperlink matrix, H is a 45.3 *billion* $\times$ 45.3 *billion* matrix! Studies show that on an average a web page has 10 links going out, meaning almost all but 10 entries in each column are 0.

Let us consider the power method for calculating the eigenvector. In this method, we begin by choosing a vector $I^{(0)}$ (which is generally considered to be $(1, 0, 0, \ldots, 0)'$) as a candidate for I and then produce a sequence of vectors $I^{(k)}$ such that

$$I^{(k+1)} = HI^{(k)}.$$

There are issues regarding the convergence of the sequence of vectors $(I^{(n)})$. Matrix under consideration must satisfy certain conditions.

For the web described in figure $A1$, if $I^{(0)} = (1, 0, 0, 0, 0, 0, 0, 0)'$ power method shows that

$$I^{(0)} = (1, 0, 0, 0, 0, 0, 0, 0)'$$
$$I^{(1)} = (0, 0.5, 0.5, 0, 0, 0, 0, 0)'$$
$$I^{(2)} = (0, 0.25, 0, 0.5, 0.25, 0, 0, 0)'$$
$$I^{(3)} = (0, 0.1667, 0, 0.25, 0.1667, 0.25, 0.0833, 0.0833)'$$
$$\vdots$$
$$I^{(60)} = (0.06, 0.0675, 0.03, 0.0675, 0.0975, 0.2025, 0.18, 0.295)'$$
$$I^{(61)} = (0.06, 0.0675, 0.03, 0.0675, 0.0975, 0.2025, 0.18, 0.295)'$$

These numbers give us the relative measures for the importance of pages. Hence we multiply all the popularities by a fixed constant so as to get the sum of popularities equal to 1.

Consider the web shown in the figure $A2$.

Figure - A2

with hyperlink matrix

$$H = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$$

The algorithm defined above applies as

$$I^{(0)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad I^{(1)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad I^{(2)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad I^{(3)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

In this web, the measure of importance of both pages is zero, indicating nothing about the relative importance of these pages. Problem arises as page 2 has no links going out. Consequently, page 2 takes some of the importance from page page 1 in each iterative step but does not pass it on to any other page, draining all the importance from the web.

Pages with no links are called *dangling nodes*, and there are, of course, many of them in the real web. We'll now modify H.

A probabilistic interpretation of H

Assume that we are on a particular web page, and we randomly follow one of its links to another page ie., if we are on page P_j with l_j links, one of which takes us to page P_i , the probability that we next end up on page P_i is then $\frac{1}{l_j}$.

As we surf randomly, let T_j be the fraction of time that we spend on page P_j. Then, the fraction of time that we spend on page P_i coming from its link in page P_j is $\frac{T_j}{l_j}$. If we end up on page P_i, then we must have come from

some page linking to it, which means

$$T_i = \sum_{P_j \in B_i} \frac{T_j}{l_j}$$

From the equation we defined for PageRank rankings, we see that $I(P_i) = T_i$ which can be understood as a web page's PageRank is the fraction of time a random surfer spends on that page.

Notice that, given this interpretation, it is natural to require that the sum of the entries in the PageRank vector I be 1, since we are considering fraction of times spent on each page.

There is a problem with the above description, if we surf randomly, then at some point we might end up at a dangling node. To overcome this, we pretend that a dangling node has a link to all the pages in the web.

Now, the hyperlink matrix H is modified by replacing the column of zeroes (if any) with a column in which each entry is $\frac{1}{n}$ where n is the total number of web pages. Let this matrix be denoted by S.

Again, consider the web

Figure - A2

$$\text{where} \quad S = \begin{bmatrix} 0 & \frac{1}{2} \\ 1 & \frac{1}{2} \end{bmatrix} \text{ and } \quad I = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \end{bmatrix}$$

meaning P_2 has twice the measure of importance of P_1, which seems reasonable now.

Note: S is also a column-stochastic matrix. Let A be a matrix (with size same as of H) whose all entries are zero except for the columns corresponding to the dangling nodes, in which each entry is $\frac{1}{n}$, then $S = H + A$.

Now, consider the web shown below

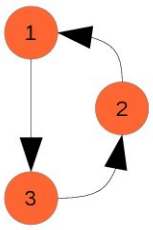

Figure - A3

where $S = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ and let $I^{(0)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ using power method, we see that

$$I^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad I^{(2)} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad I^{(3)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad I^{(4)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \ldots$$

In this case power method fails because, 1 is not a simple eigenvalue of the matrix S.

Consider the web shown below

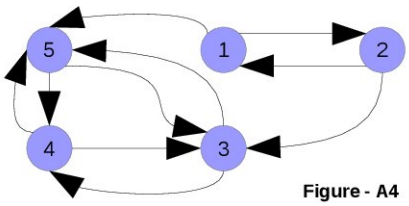

Figure - A4

$$S = \begin{bmatrix} 0 & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{2} & 0 \end{bmatrix} \text{ and let } I^{(0)} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \text{ now, using power method}$$

$$I^{(1)} = \begin{bmatrix} 0 \\ 0.5 \\ 0 \\ 0 \\ 0.5 \end{bmatrix} \qquad I^{(2)} = \begin{bmatrix} 0.25 \\ 0 \\ 0.5 \\ 0.25 \\ 0 \end{bmatrix} \qquad I^{(3)} = \begin{bmatrix} 0 \\ 0.125 \\ 0.125 \\ 0.25 \\ 0.5 \end{bmatrix}$$

$$\ldots I^{(13)} = \begin{bmatrix} 0 \\ 0.0001 \\ 0.3325 \\ 0.3332 \\ 0.3341 \end{bmatrix} \qquad I^{(14)} = \begin{bmatrix} 0 \\ 0 \\ 0.3337 \\ 0.3333 \\ 0.3328 \end{bmatrix} \qquad I^{(15)} = \begin{bmatrix} 0 \\ 0 \\ 0.3331 \\ 0.3333 \\ 0.3335 \end{bmatrix}$$

Hence, $I = \begin{bmatrix} 0 \\ 0 \\ 0.333 \\ 0.333 \\ 0.333 \end{bmatrix}$ where PageRanks assigned to page 1 and page 2 are zero, which is unsatisfactory as page 1 and page 2 have links coming in and going out of them. The problem here is that the web considered has a smaller web in it, ie., pages $3, 4, 5$ are a web of themselves. Links come into this sub web formed by pages $3, 4, 5$, but none go out. Just as in the example of the dangling node, these pages form an "importance sink" that drains the importance out of the other two pages. In mathematical terms power method doesn't work here as S is not irreducible.

One last modification:

We will modify S to get a matrix which is irreducible and has 1 as a simple eigenvalue. As it stands now, our movement while surfing randomly is determined by S ie., either we follow one of the links on the current page or, if we are at a page with no links, we randomly choose any other page to move to. To make our modification, we will first choose a parameter $\alpha \quad \ni \quad 0 < \alpha < 1$. Now, suppose we move in a slightly different way. With probability α we are guided by S, and with probability $1 - \alpha$ we choose the

next page at random.

Now we obtain the *Google Matrix*

$$G = \alpha S + (1 - \alpha)\frac{1}{n}J$$

where J is a matrix, all of whose entries are 1.

Note: G is a column-stochastic matrix. Also, G is a positive matrix, hence by Perron's theorem G has a unique eigenvector I corresponding to the eigenvalue 1, which can be found using the power method.

Parameter α:

The role of the parameter α is important. If $\alpha = 1$ then, $G = S$ which means we are dealing with the unmodified version. If $\alpha = 0$ then $G = \frac{1}{n}J$ which means the web we are considering has a link between any two pages and we have lost the original hyperlink structure of the web. Since, α is the probability by which we are guided by S, we would like to choose α closer to one, so that the PageRanks are weighted heavily into the calculations.

But, the convergence of the power method is geometric with ratio $|\frac{\lambda_2}{\lambda_1}|$, where λ_1 is the eigenvalue with maximum magnitude and λ_2 is the eigenvalue closest in magnitute to the magnitude of λ_1. Hence power method converges slowly if λ_2 is close to λ_1.

Theorem:(Taher & Sepandar) Let P be a $n \times n$ row-stochastic matrix. Let c be a real number such that $0 \leq c \leq 1$. Let E be a $n \times n$ row-stochastic matrix $E = ev^T$, where e is the n-vector whose elements are all $e_i = 1$, and v is an n-vector that represents a probability distribution. Let $A = (cP + (1 - c)E)^T$, then its second eigenvalue $|\lambda_2| \leq c$.

Theorem:(Taher & Sepandar) Further, if P has at least two irreducible closed subsets (which is the case for the hyperlink matrix), then the second eigenvalue of A is given by $\lambda_2 = c$.

Hence for the Google matrix, $|\lambda_2| = \alpha$, which means when α is close to 1, the power method converges slowly.

With all these considerations on the parameter, it is believed that (not known!), Larry Page and Serge Brin chose $\alpha = 0.85$.

Computations:

In the theory mentioned above, matrices under consideration are of the order 45.3 *billion* $\times$ 45.3 *billion*. Remember $S = H + A$ and hence Google matrix has the form

$$G = \alpha H + \alpha A + \frac{(1-\alpha)}{n} J$$

$$\therefore \quad GI^{(k)} = \alpha H I^{(k)} + \alpha A I^{(k)} + \frac{(1-\alpha)}{n} J I^{(k)}$$

Recall that, most of the entries in H are zero, hence evaluating $HI^{(k)}$, on an average requires only ten nonzero terms for each entry in the resultant vector. Also, rows of A are all identical as are the rows of J. Therfore, evaluating $AI^{(k)}$ and $JI^{(k)}$ amount to adding the current importance rankings of the dangling nodes or of all web pages. This only needs to be done once.

It is guessed that, Google believes that with $\alpha = 0.85$, $50 - 100$ iterations are required to obtain a sufficiently good approximation to I. For, 45.3 *billion* web pages, the calculations are expected to take a few days to complete. The web is continually changing, pages might be created or deleted, and links in or to the pages also might be added or removed. It is rumored that Google recomputes the PageRank vector I roughly every month. Since the PageRank of pages can be observed to fluctuate considerably during this computations, it is known to some as the *Google Dance*!

Consider the following web again, look at its PageRank vectors

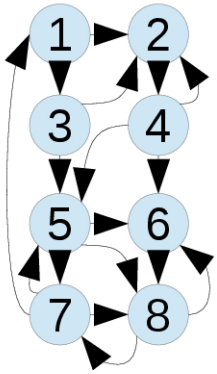

Figure - A1

$$\text{with} \quad H = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{3} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & \frac{1}{3} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 0 & \frac{1}{3} & 1 & \frac{1}{3} & 0 \end{bmatrix}$$

$$\alpha = 0.65,\ m \geq 16, \quad I = \begin{bmatrix} 0.0734 \\ 0.1153 \\ 0.0676 \\ 0.1187 \\ 0.1210 \\ 0.1623 \\ 0.1366 \\ 0.2051 \end{bmatrix} \quad ; \quad \alpha = 0.75,\ m \geq 17, \quad I = \begin{bmatrix} 0.0675 \\ 0.1054 \\ 0.0566 \\ 0.1102 \\ 0.1163 \\ 0.1727 \\ 0.1452 \\ 0.2261 \end{bmatrix}$$

$$\alpha = 0.85,\ m \geq 17, \quad I = \begin{bmatrix} 0.0632 \\ 0.0925 \\ 0.0455 \\ 0.0974 \\ 0.1101 \\ 0.1839 \\ 0.1564 \\ 0.2510 \end{bmatrix}$$

A python code, which takes as input the matrix $H + A$, α and m = no. of iterations to be considered, and calculates the PageRank with m iterations is given below:

```
def matmult(m1,m2):
    m3 = [[0 for q in range(len(m2[0]))] for p in range(len(m1))]
    for i in range(len(m1)):
        for k in range(len(m2)):
            for j in range(len(m2[0])):
                m3[i][j] = m3[i][j] + m1[i][k]*m2[k][j]
    return(m3)

def scalmatmult(c,M):
    m = [[0 for q in range(len(M[0]))] for p in range(len(M))]
    for i in range(len(M)):
        for j in range(len(M[0])):
            m[i][j] = m[i][j] + c*M[i][j]
    return(m)

def matadd(m,M):
    matsum = [[0 for q in range(len(M[0]))] for p in range(len(M))]
    for i in range(len(M)):
        for j in range(len(M[0])):
            matsum[i][j] = matsum[i][j] + m[i][j] + M[i][j]
    return(matsum)

def pagerank(S,alpha,m):
    I = [[0] for i in range(len(S))]
    I[0][0] = 1
    J = [[1 for q in range(len(S[0]))] for p in range(len(S))]
    G = matadd(scalmatmult(alpha,S),scalmatmult(((1-alpha)/len(S)),J))
    for j in range(0,m):
        I = matmult(G,I)
    return(I)
```

Advances:

Google Panda is a change to the Google's search results ranking algorithm that was first released in February 23, 2011. The change aimed to lower the rank of low-quality sites or thin sites, and return higher-quality sites near the top of the search results. CNET reported a surge in the rankings of news websites and social networking sites, and a drop in rankings for sites containing large amounts of advertising. This change reportedly affected the rankings of almost 12 percent of all search results. Soon after the Panda rollout, many websites, including Google's webmaster forum, became filled with complaints of scrapers/copyright infringers getting better rankings than sites with original content. At one point, Google publicly asked for data points to help detect scrapers better. Google's Panda has received several updates since the original rollout in February 2011, and the effect went global in April 2011. To help affected publishers, Google published an advisory on its blog, thus giving some direction for self-evaluation of a website's quality. Google has provided a list of 23 bullet points on its blog answering the question of "What counts as a high-quality site?" that is supposed to help webmasters step into Google's mindset.

Google Panda was built through an algorithm update that used artificial intelligence in a more sophisticated and scalable way than previously possible. Human quality testers rated thousands of websites based on measures of quality, including design, trustworthiness, speed and whether or not they would return to the website. Google's new Panda machine-learning algorithm, made possible by and named after engineer Navneet Panda, was then used to look for similarities between websites people found to be high quality and low quality.

Google Penguin is a code name for a Google algorithm update that was first announced on April 24, 2012. The update is aimed at decreasing search engine rankings of websites that violate Googles Webmaster Guidelines by using black-hat SEO techniques, such as keyword stuffing, cloaking, participating in link schemes, deliberate creation of duplicate content, and others. Penguin update went live on April 24, 2012.

By Googles estimates, Penguin affects approximately 3.1% of search queries in English, about 3% of queries in languages like German, Chinese, and Arabic, and an even bigger percentage of them in highly-spammed languages. On May 25th, 2012, Google unveiled the latest Penguin update, called Penguin 1.1. This update, was supposed to impact less than one-tenth of a percent

of English searches. The guiding principle for the update was to penalise websites using manipulative techniques to achieve high rankings. Penguin 3 was released Oct. 5, 2012 and affected 0.3% of queries.

In January 2012, so-called page *layout algorithm* update was released, which targeted websites with little content above the fold. The strategic goal that Panda, Penguin, and page layout update share is to display higher quality websites at the top of Googles search results. However, sites that were downranked as the result of these updates have different sets of characteristics. The main target of Google Penguin is spamdexing (including link bombing).

References:

- Austin, David 2006, 'How Google Finds Your Needle in the Web's Haystack', *Mathematical Society Feature Column* <http://www.ams.org/samplings/feature-column/fcarc-pagerank>
- Williams, Lance R. 2012, *CS 530: Geometric and Probabilistic Methods in Computer Science*, Lecture notes, University of New Mexico, Albuquerque.
- Ramasubramanian, S. 2012, *Probability III (Introduction to Stochastic Processes)*, Lecture notes, Indian Statistical Institute, Bangalore.
- Karlin, Samuel & Taylor, Howard M. 1975, 'Markov Chains', *A first course in Stochastic Processes*, Second Edition, Academic Press, New York, pp. 45-80.
- Ciarlet, Philippe G. 1989, *Introduction to Numerical Linear Algebra and Optimisation*, First Edition, University Press, Cambridge.
- Deng, Bo 2010, *Math 428: Introduction to Operations Research*, Lecture notes, University of Nebraska-Lincoln, Lincoln.
- Page, Lawrence & Brin, Serge 1998, The antaomy of a large-scale hypertextual Web search engine, *Computer Networks and ISDN Systems*, 33, pp. 107-17 <http://infolab.stanford.edu/pub/papers/google.pdf>
- Atherton, Rebecca 2005, 'A Look at Markov Chains and Their Use in Google', Master's thesis, Iowa State University, Ames.

- Haveliwala, Taher H. & Kamvar, Sepandar D. 2003, The second eigenvalue of the Google matrix, *Stanford University Technical Report*.

- *PageRank*, modified 06.11.2012, Wikipedia, viewed 12.11.2012, <http://en.wikipedia.org/wiki/PageRank>

- *Google Panda*, modified 09.11.2012, Wikipedia, viewed 12.11.2012, <http://en.wikipedia.org/wiki/Google_Panda>

- *Google Penguin*, modified 19.10.2012, Wikipedia, viewed 12.11.2012, <http://en.wikipedia.org/wiki/Google_Penguin>