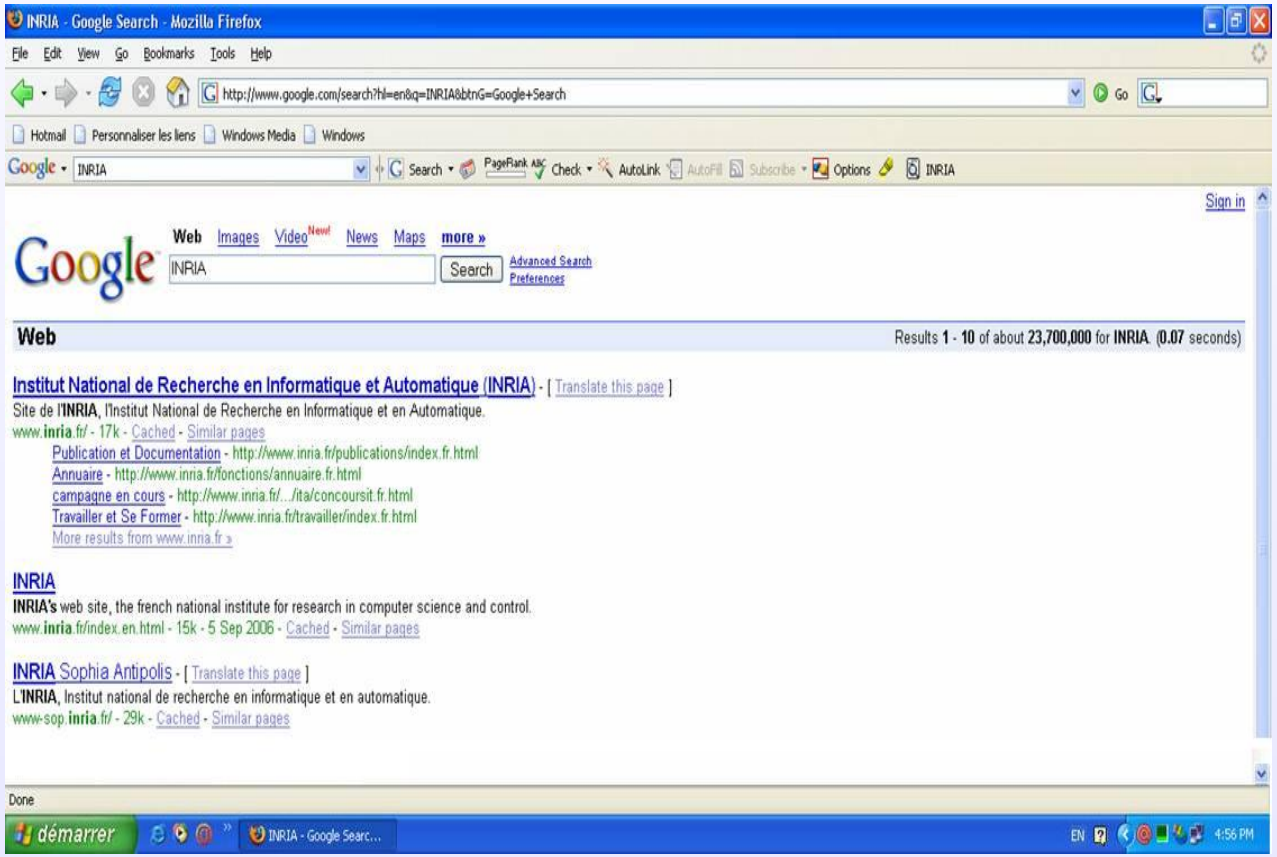
Mathematical Analysis of Google PageRank

Konstantin Avrachenkov

INRIA Sophia Antipolis, France

Ranking Answers to User Query



Ranking Answers to User Query

How a search engine should sort the retrieved answers?

Possible solutions: (a) use the frequency of the searched terms in the Web page, (b) analyse the log files,... These solutions might be not objective.

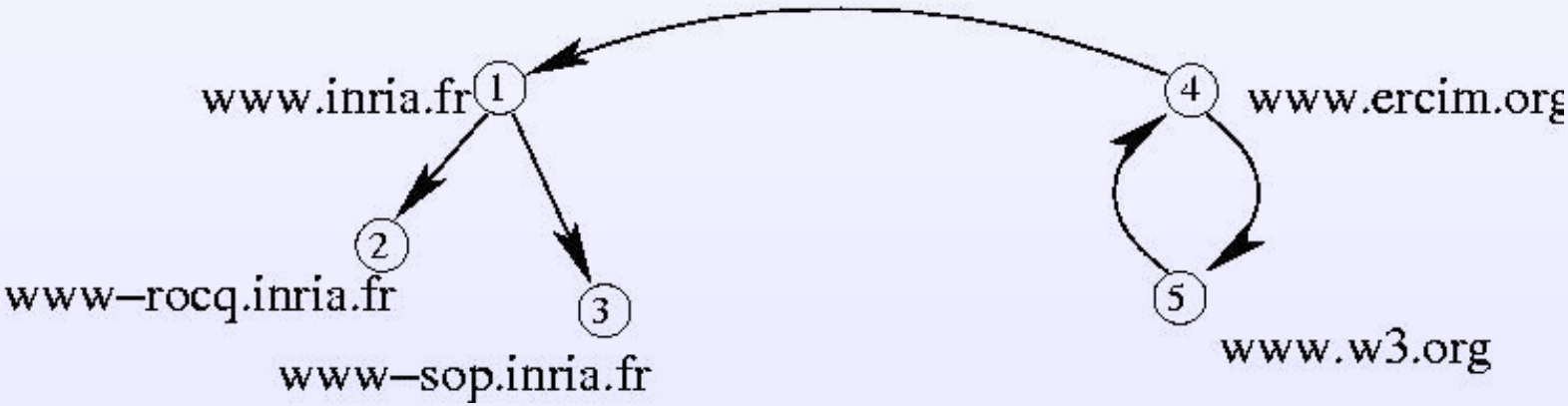
An original idea of Google is based on two observations:

1. The more pages point to a Web page, the more important the page is.
2. If more important Web pages point to the page, the page is even more important.

INRIA SOPHIA-ANTIPOLIS

Web Graph

Consider the Web as a directed graph:



Random surfer PageRank Definition

Consider a random surfer who, with probability c (=0.85) follows a randomly chosen outgoing link, otherwise, with probability $1 - c$ jumps to a completely random page.

Then, PageRank π_i of page i is the long run fraction of time that a random surfer spends on page i.

The dynamics of random surfer can be described using Markov chains.

▸ Markov chains definition

INRIA SOPHIA-ANTIPOLIS

Formal PageRank Definition

Let n be the total number of pages on the Web ($n \approx 8 \times 10^9$).

Define the hyperlink matrix $P = \{p_{ij}\}_{i,j=1}^{n}$ as follows:

- $p_{ij} = 1/d_i$, if j is one of the d_i outgoing links of i,
- $p_{ij} = 1/n$, if $d_i = 0$ (dangling node),
- $p_{ij} = 0$, otherwise.

The transitions of "easily bored surfer" corresponds to the following perturbed Google matrix

$$\tilde{P} = cP + (1 - c)(1/n)E,$$

where E is an $n \times n$ matrix consisting of one's, $c = 0.85$.

INRIA SOPHIA-ANTIPOLIS

Formal PageRank Definition

Then, the PageRank vector is a solution of

$$\pi \tilde{P} = \pi, \quad \pi \underline{1} = 1,$$

or, equivalently, in the component form

$$\pi_i = \sum_{j \to i} \frac{c}{d_j} \pi_j + \frac{1-c}{n}, \quad \sum_i \pi_i = 1.$$

Example

$$P = \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 \\ 0.5 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$\pi = \begin{bmatrix} 0.1982 & 0.1731 & 0.1731 & 0.2573 & 0.1982 \end{bmatrix}$$

Power Method

Even though this is a well kept secret, it seems that Google still uses the simple power iteration method for PageRank computation

$$\pi^{(k+1)} = c\pi^{(k)}P + (1-c)\frac{1}{n}\underline{1}^T, \quad \pi^{(0)} = \frac{1}{n}\underline{1}^T.$$

It can be easily estimated that using the constant $c = 0.85$ Google achieves the tolerance level (measured by the residual $\pi^{(k+1)} - \pi^{(k)}$) of $10^{-3} - 10^{-5}$ for only 50-100 iterations.

But even this small number of iterations takes Google about a week to update the PageRank...

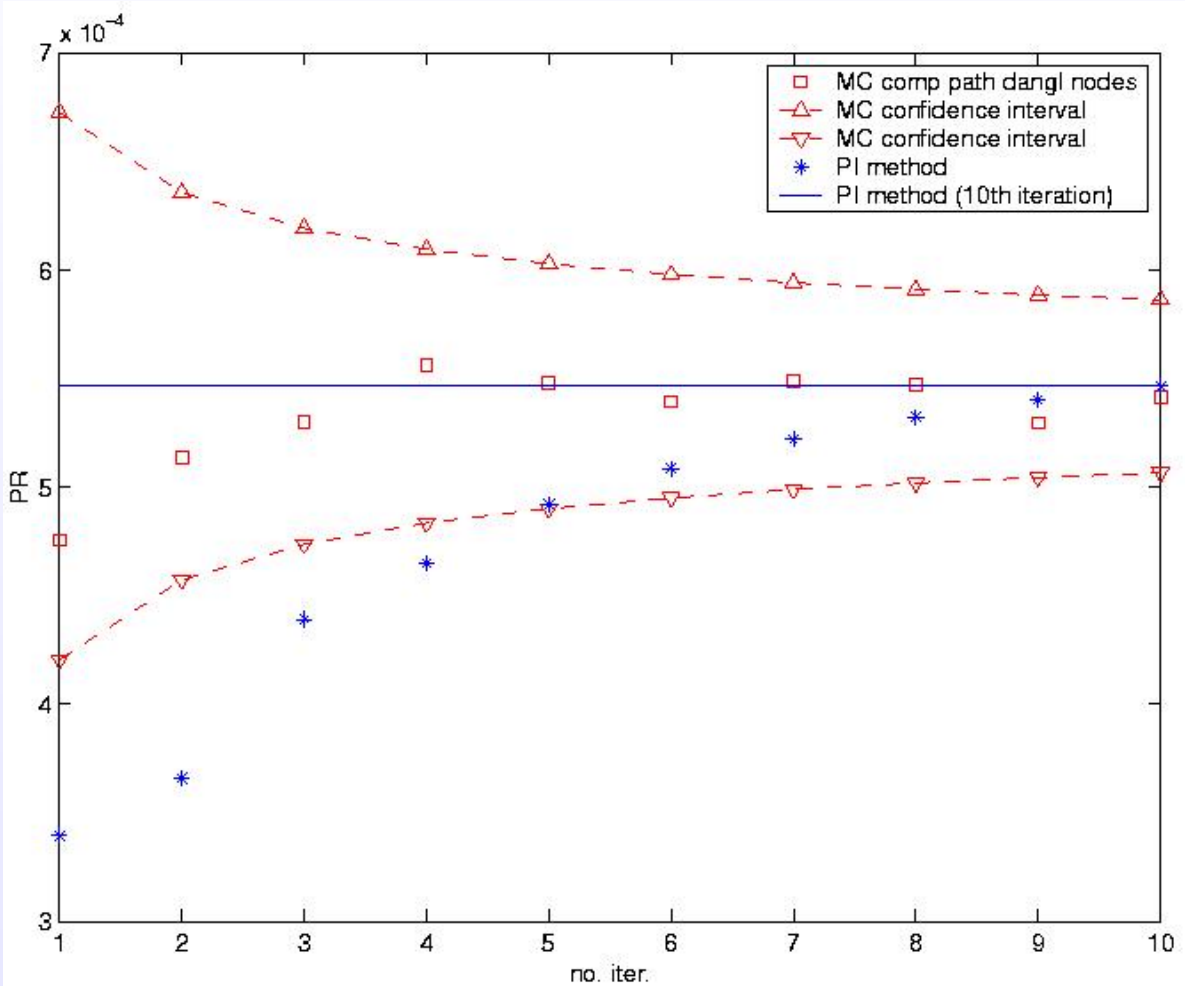
INRIA SOPHIA-ANTIPOLIS

Monte Carlo Method

Run random surfer process $(X_t)_{t\geq 0}$, m times from each page, terminating at each step with probability $1-c$. Evaluate π_j as $\bar{\pi}_j =$[fraction of time spent in j].

It turns out the one iteration of Monte Carlo method ($m=1$) is sufficient to estimate well the PageRank of important pages.

Monte Carlo Method



Advantages of MC Method in respect to PI Method

- Monte Carlo method has natural parallel implementation;
- Monte Carlo method provides good estimation of the PageRank for important pages already after one iteration;
- Monte Carlo method allows one to perform continuous update of the PageRank as the structure of the Web changes.

INRIA SOPHIA-ANTIPOLIS

Decomposition based on SCC

It is known that the Web Graph consistes of many disjoint Strongly Connected Components (SCCs). This fact implies that the hyperlink matrix has the following form

$$P = \begin{bmatrix} P_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & P_N \end{bmatrix},$$

where the elements of diagonal blocks P_I, $I = 1, ..., N$, correspond to links inside the I-th SCC. Denote by n_I the size of the I-th SCC.

Decomposition based on SCC

For each block I, define the Google matrix
$\tilde{P}_I = cP_I + (1-c)(1/n_I)E$,
and let vector π_I be the PageRank of SCC I such that

$$\pi_I \tilde{P}_I = \pi, \quad \pi_I \underline{1} = 1.$$

Then the following theorem holds.

Theorem

The PageRank π is given by

$$\pi = ((n_1/n)\pi_1, (n_2/n)\pi_2, \ldots, (n_N/n)\pi_N). \qquad (1)$$

INRIA SOPHIA-ANTIPOLIS

To what extend a page can control its PageRank?

Let us give a rough estimation by how much a page can control its PageRank by modifying its outgoing links.

Define a discrete-time absorbing Markov chain $\{X_t, t = 0, 1, \ldots\}$ with the state space $\{0, 1 \ldots, n\}$, where transitions between the states $1, \ldots, n$ are conducted by the matrix cP, and the state 0 is absorbing.

Let N_j be the number of visits to state $j = 1, \ldots, n$ before absorption. Then, denote $z_{ij} := \mathbb{E}(N_j | X_0 = i)$.

To what extend a page can control its PageRank?

Let q_{ji} be the probability to reach the state i before absorption if the initial state is j.

We have the following decomposition result:

Theorem

The PageRank of page $i = 1, \ldots, n$ is given by

$$\pi_i = \frac{1-c}{n} z_{ii} \left(1 + \sum_{\substack{j=1 \\ j \neq i}}^{n} q_{ji} \right), \quad i = 1, \ldots, n. \tag{2}$$

▸ Proof

To what extend a page can control its PageRank?

The decomposition formula (2) represents the PageRank of page i as a product of three multipliers where only the term z_{ii} depends on the outgoing links of page i.

Hence, by changing the outgoing links, a page can control its PageRank up to a multiple factor

$$z_{ii} = 1/(1 - q_{ii}) \in [1, 1/(1 - c^2)],$$

where $q_{ii} \in [0, c^2]$ is a probability to return back to i starting from i before absorption.

Note that the upper bound $1/(1 - c^2)$ (approximately 3.6 for $c = .85$) is hard or rather not possible to achieve...

To what extend a page can control its PageRank?

We note that even a threefold increase of the PageRank might not be considered as a significant improvement, since Google measures the PageRank on a logarithmic scale.

Next we show how a Web page should use its scarce resources to increase its PageRank.

Optimal Linking Strategy

Let us show that there exists in fact an optimal linking strategy.

Consider a page $i = 1, \ldots, n$ and assume that i has links to the pages $i_1, \ldots, i_k$ where $i_l \neq i$ for all $l = 1, \ldots, k$.

Then for the mean return time, we have

$$\mu_{ii} = 1 + \frac{c}{k} \sum_{l=1}^{k} \mu_{i_l i} + \frac{1}{n}(1-c) \sum_{\substack{j=1 \\ j \neq i}}^{n} \mu_{ji}, \qquad (3)$$

where μ_{ij} is the mean first passage time from page i to page j and c is the Google constant.

Since $\pi_i = 1/\mu_{ii}$, the objective now is to choose k and $i_1, \ldots, i_k$ such that μ_{ii} becomes as small as possible.

Optimal Linking Strategy

From (3) one can see that μ_i is a linear function of μ_{ji}'s.
Moreover, outgoing links from i do not affect μ_{ji}'s.

Thus, the best what one can do is to link only to one Web page j^* such that

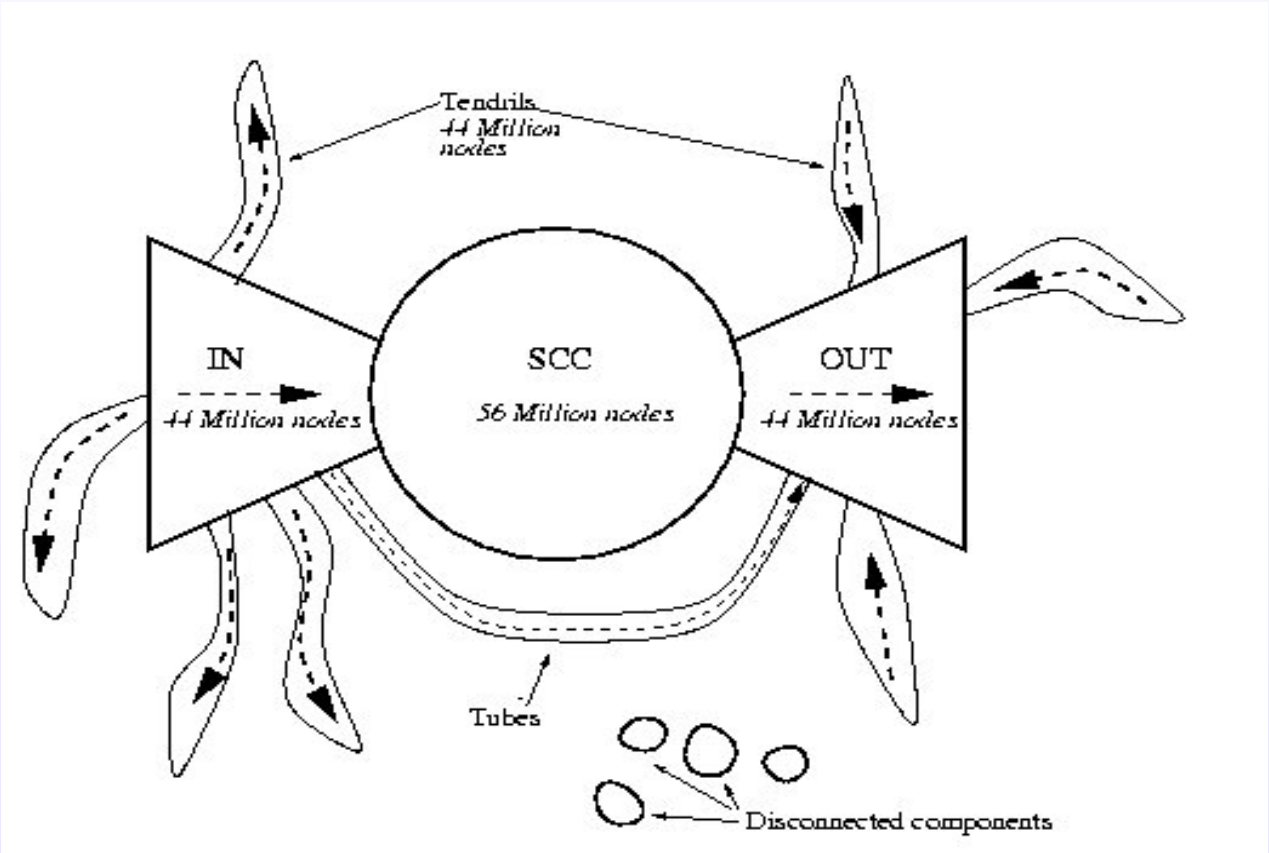
$$\mu_{j^*i} = \min_j \{\mu_{ji}\}.$$

Note that (surprisingly) the PageRank of j^* plays no role here.

Still, as was already mentioned, we need to admit that a Web page owner has very limited control of his/her PageRank.

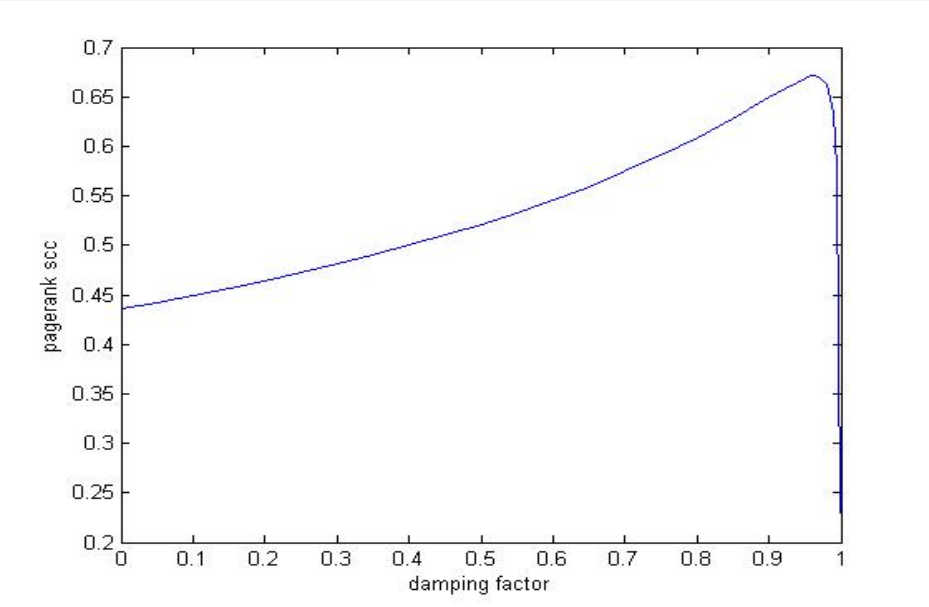
The Bowtie structure of the Web graph

A. Broder *et al.* 2000 and R. Kumar *et al.* 2000 have observed that the Web Graph has a Bowtie structure.



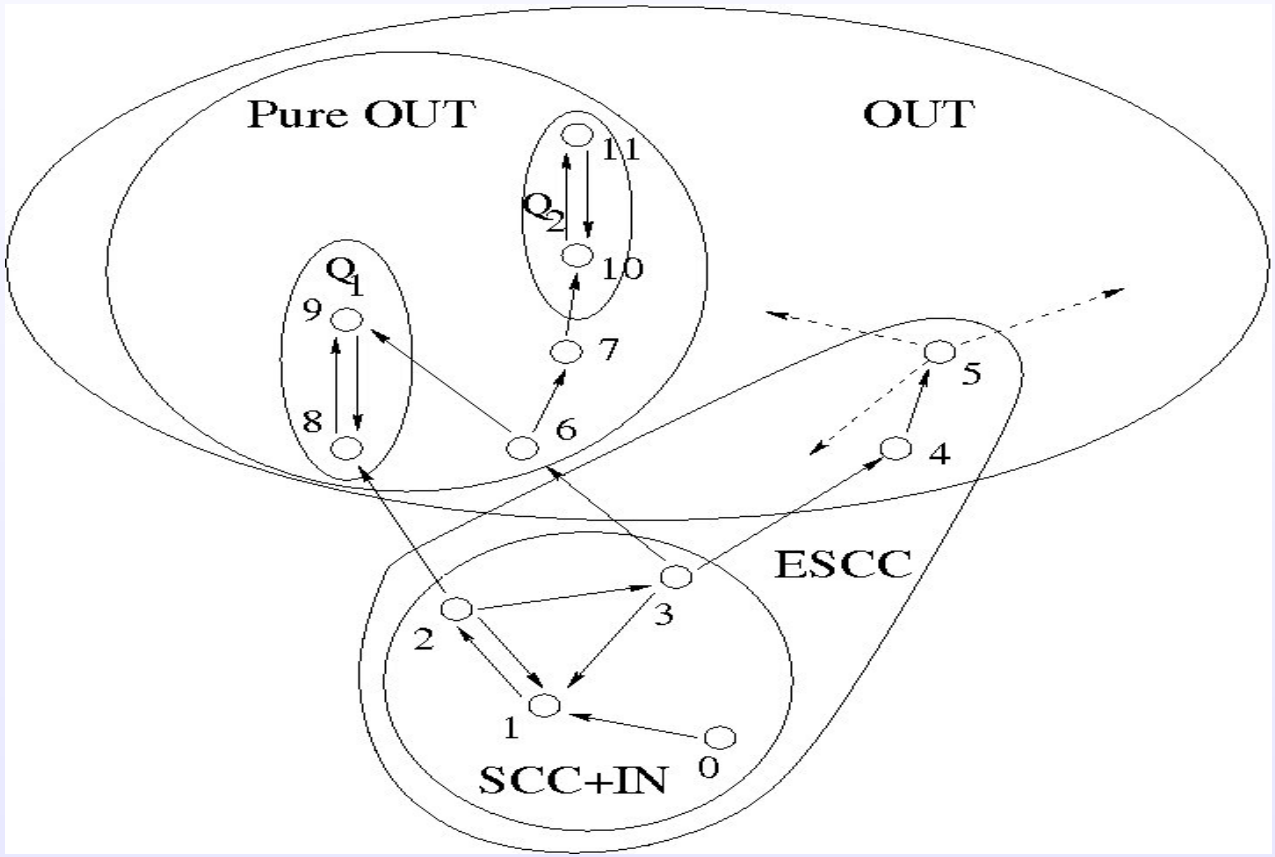
Maximizing the mass of SCC

One can choose the damping factor c to maximize the total PageRank mass of SCC:



However, the factor c becomes to close to one. Is it good?

More detailed structure of the Web graph



By renumbering the nodes, the transition matrix P can be then transformed to the following form

$$P = \begin{bmatrix} Q & 0 \\ R & T \end{bmatrix}, \tag{4}$$

where

the block T corresponds to the Extended SCC,

the block Q corresponds to the part of the OUT component without dangling nodes and their predecessors,

and the block R corresponds to the transitions from ESCC to the nodes in block Q.

As was observed by Moler 2003, the PageRank vector can be expressed by the following formula

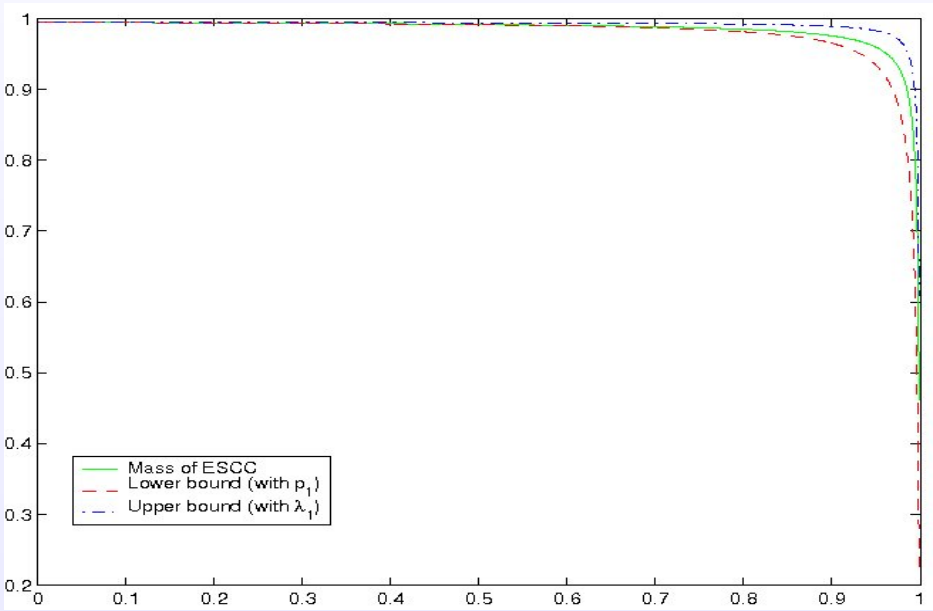
$$\pi = \frac{1-c}{n}\mathbf{1}^T[I - cP]^{-1}. \qquad (5)$$

If we substitute the expression (4) for the transition matrix P into (5), we obtain the following formula for the part of the PageRank vector corresponding to the nodes in ESCC:

$$\pi_T = \frac{1-c}{n}\mathbf{1}^T[I - cT]^{-1} = \alpha(1-c)u_T[I - cT]^{-1}, \qquad (6)$$

where $\alpha = n_T/n$ and n_T is the number of nodes in ESCC, and where u_T is the uniform distribution over all ESCC nodes.

First, we note that since matrix T is substochastic, the inverse $[I - T]^{-1}$ exists and consequently $\pi_T \to 0$ as $c \to 1$.



Clearly, it is not good to take the value of c too close to one.

It follows that the value of c should not be chosen in the critical region where the PageRank mass of the ESCC component is rapidly decreasing.

Luckily, the shape of the function $||\pi_{\mathcal{T}}(c)||_1$ is such that it decreases drastically only when c is really close to one, which leaves a lot of freedom for choosing c.

In particular, the famous Google constant $c = 0.85$ is small enough to ensure a reasonably large PageRank mass of ESCC.

However, as we have observed in numerical experiments, even moderately large values of c result in an unfairly large PageRank mass of the Pure OUT component.

Now, our goal is to find the values of c that lead to a "fair" distribution of the PageRank mass between the Pure OUT and the ESCC components.

Let v be some probability vector over ESCC. We would like to choose $c = c^*$ that satisfies the condition

$$||\pi_T(c)|| = ||vT||, \tag{7}$$

that is, starting from v, the probability mass preserved in ESCC after one step should be equal to the PageRank of ESCC.

Reasonable choices of v:

1. $\hat{\pi}_T$, the quasi-stationary distribution of T,
2. the uniform vector u,
3. the normalized PageRank vector $\pi_T(c)/||\pi_T(c)||$.

All three criteria indicate that $c = 1/2$ seems to be quite a good choice.

INRIA SOPHIA-ANTIPOLIS

Experiments with the log files

c	PR rank w/o link	PR rank with link	rank by no. of clicks
Node A			
0.5	1648	2307	2588
0.85	731	2101	2588
0.95	226	2116	2588
Node B			
0.5	1648	4009	3649
0.85	731	3279	3649
0.95	226	3563	3649

Table: Comparison between PR and click based rankings.

Recommendations for Web Page Design

Now we can suggest the following recommendations for Web Page Design:

1. The more pages a Web site has, the better.
2. Link all pages inside a Web site to the main page. This way the main page will have a significant weight.
3. Give hyperlinks to the Departement and Institution Web sites.
4. Do not make inappropriate links.

And, of course, one should not forget that content still matters for Google. There is really no substitute for good original content...

Thank you!

INRIA SOPHIA-ANTIPOLIS

Markov Chain Definition

Discrete-time discrete-state Markov chain is a stochastic process $\{X_n\}_{n=0}^{\infty}$ on the set of states $S = \{1, 2, ..., |S|\}$ such that

$$P\{X_{n+1} = j\} = \sum_{i \in S} P\{X_{n+1} = j | X_n = i\} P\{x_n = i\}.$$

We denote $p_{ij} := P\{X_{n+1} = j | X_n = i\}$ and call $\{p_{ij}\}_{i,j=1}^{|S|}$ the matrix of transition probabilities.

return

To what extend a page can control its PageRank?

Proof: It follows from (**??**) that

$$\pi_i = \frac{1-c}{n}\underline{1}^T[I - cP]^{-1}e_i = \frac{1-c}{n}\sum_{j=1}^{n} z_{ji}. \qquad (8)$$

Next, we note that for any $i, j = 1, \ldots, n$; $i \neq j$, we have

$$z_{ji} = q_{ji} z_{ii},$$

and consequently, substituting the last equation in (8) we obtain (2). Q.E.D.

return