The Theory behind PageRank

Mauro Sozio

Telecom ParisTech

December 5, 2016

The PageRank Algorithm: History and Facts

It was devised at Stanford by Larry Page and Sergei Brin, (the founders of Google) in 1996 as part of a research project on a new search engine.

The paper was published in 1998 [1, 2] and shortly after the authors founded Google. Named after one of the authors L. Page.

It is a *link analysis algorithm*, i.e., only the links between pages (not their content) are considered.

Other link analysis algorithms: community and spam detection, HITS, ...

Ranking Web Pages

Humans cannot make sense of billions of web pages, which have to be ranked according to their "importance".

The web pages `www.stanford.edu` and `www.johnsmith.com` have not the same importance. The former one has more than 25K Web pages linking to it, the latter one only 10.

Simple algorithm: Rank the pages according to the # of links to them.

Are all web pages linking to `www.stanford.edu` equally important? Cornell University homepage more important than that of J. Smith friend.

Ranking Web Pages

Humans cannot make sense of billions of web pages, which have to be ranked according to their "importance".

The web pages `www.stanford.edu` and `www.johnsmith.com` have not the same importance. The former one has more than 25K Web pages linking to it, the latter one only 10.

Simple algorithm: Rank the pages according to the # of links to them.

Are all web pages linking to `www.stanford.edu` equally important? Cornell University homepage more important than that of J. Smith friend.

$\Rightarrow$ Recursive definition of importance.

Computing Importance

Web graph: a directed graph $G = (V, E)$ where nodes represent web pages, while there is a directed edge between u and v if there is a hyperlink between the corresponding web pages.

Importance of v is proportional to the importance of nodes linking to v.

It can be modeled by a system of linear equations ...

System of Linear Equations for PageRank

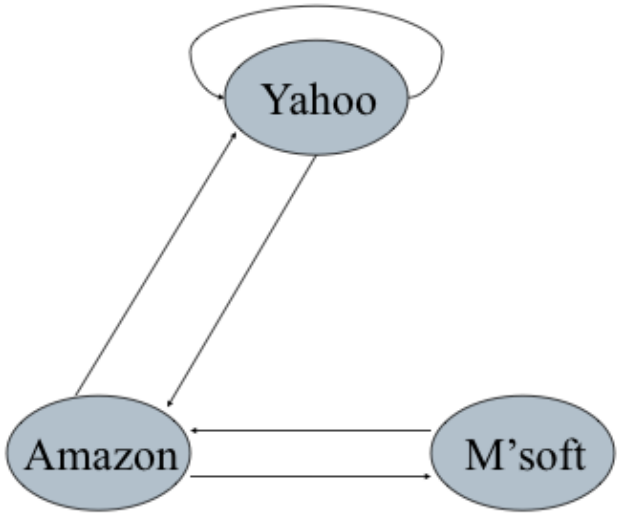
Let $G = (V, E)$ (web graph) be a directed graph, with $V = \{v_1, \ldots, v_n\}$. Let $\delta_{\text{in}}(v)$ be the in-degree of v, i.e. $\delta_{in}(v) = |\{u : (u, v) \in E\}|$, while let $\delta_{\text{out}}(v)$ be its out-degree, i.e. $\delta_{out}(v) = |\{u : (v, u) \in E\}|$.

Let M_G (M for short) be a $n \times n$ matrix with entries in $[0, 1]$ as follows:

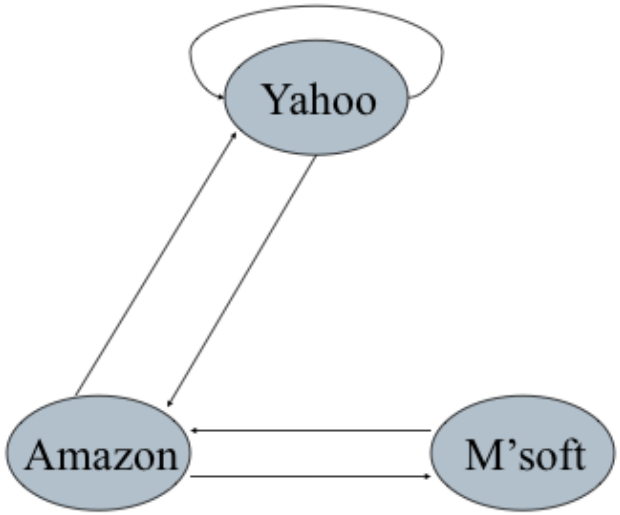
$$M_{ij} = \begin{cases} \frac{1}{\delta_{\text{out}}(v_j)} & if\,(v_j, v_i) \in E \\ 0 & if\,(v_j, v_i) \notin E \end{cases}, \quad \forall i, j \in [1, n].$$

Any π s.t. $\pi = M\pi$ and $\sum_{i=1}^{n} \pi_i = 1$ meets our requirements.

Example

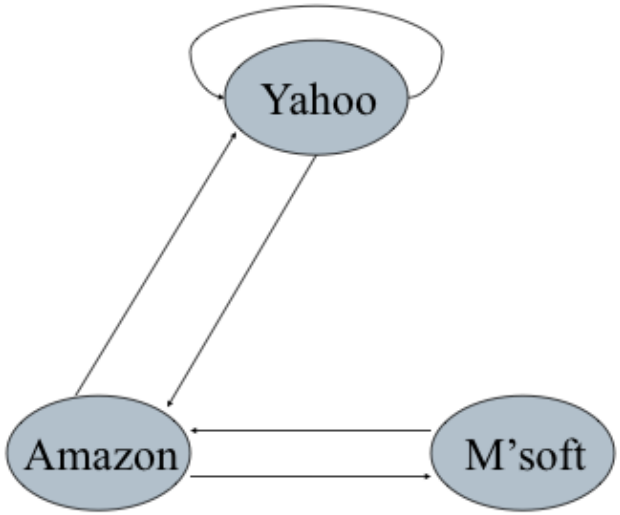


Example

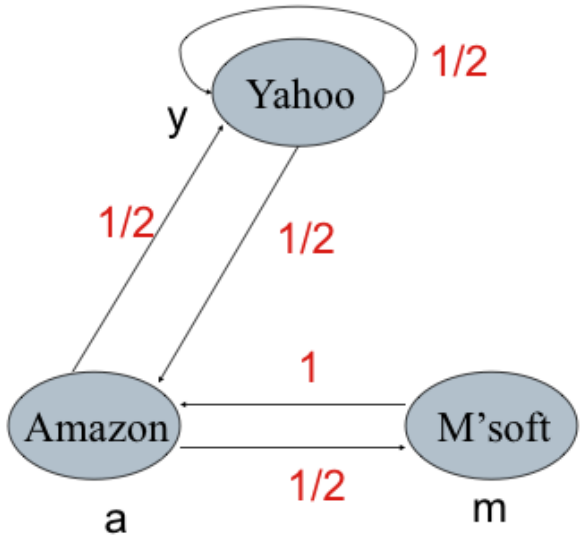


$$M_G = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}$$

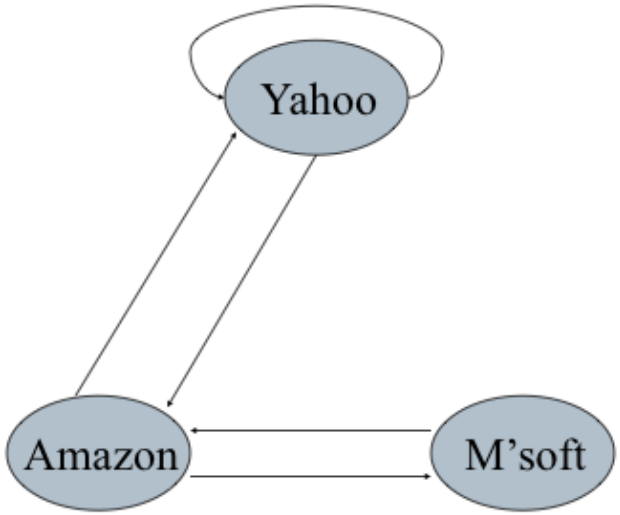
Example



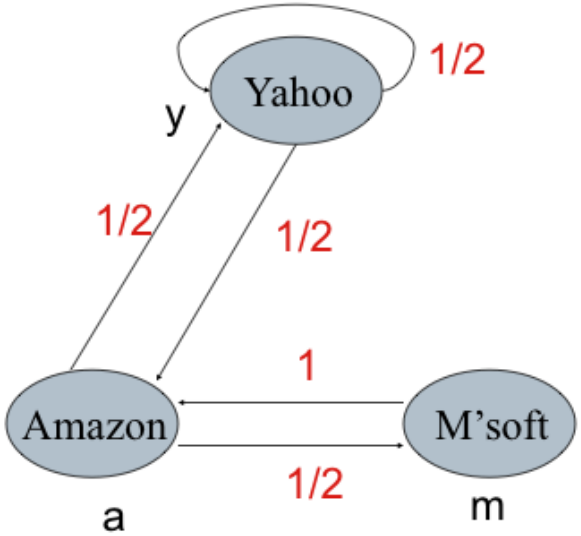
$$M_G = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix}$$



Example



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



$$\begin{aligned} y &= \frac{y}{2} + \frac{a}{2} \\ a &= \frac{y}{2} + m \\ m &= \frac{a}{2} \\ y+a+m &= 1 \end{aligned}$$

PageRank

The importance of a web page can be computed by solving the corresponding system of linear equations.

However there are two main issues:

- The solution might not be unique!
- It is expensive to solve large system of linear equations. *Gaussian elimination* requires $\Omega(n^3)$ operations.

PageRank: Eigenvector Computation

Definition 1

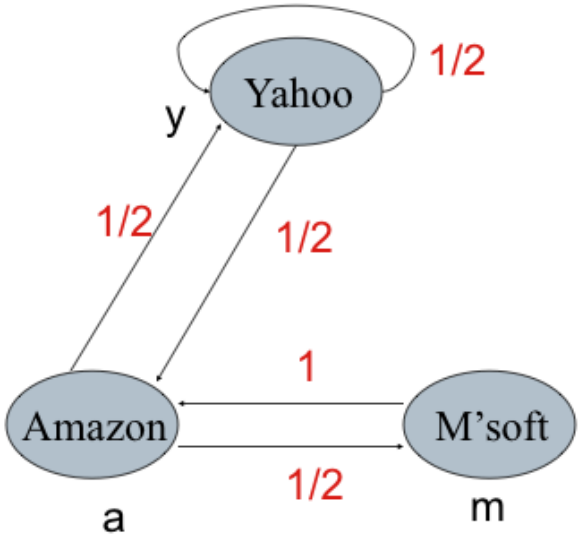
The vector x is an eigenvector of the matrix A with eigenvalue λ if

$$Ax = \lambda x.$$

Therefore, one could compute the importance of web pages by computing an eigenvector with eigenvalue 1 of M_G, which is also very expensive!

PageRank: Random Surfer

A *random surfer* starts surfing the web from a random page (step 0). At step t, let u be the web page currently visited by the random surfer. At step $t+1$, the random surfer visits a web page v being picked uniformly at random from the $\delta_{\text{out}}(u)$ neighbors.



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

M_{ij} is the probability that the random surfer moves from page u_j to u_i.

PageRank = probability of visiting web pages

We would like to compute the importance of a web page u as the probability that the random surfer visits u at step $t > 0$ for some large t.

Issues:

- Which value of t?
- The probability of visiting u might depend on the starting page.
- Can such a probability be computed efficiently?
- So called *spider traps* and dead-ends might give non-interesting results. A spider trap is a set of web pages S not containing any link to $V \setminus S$. *Dead-ends* are web pages not having any link to any other web page. We assume for simplicity that there are no dead-ends.

Random Surfer 2.0

The random surfer starts from one page chosen uniformly at random. Then, at step t it either 1) follows one random link from the current page with prob. β or 2) "jumps" to any of the n pages uniformly at random.

Common values for β are in $[0.8, 0.9]$.

Let A be the $n \times n$ matrix where $A_{ij} = \beta M_{ij} + \frac{1-\beta}{n}$, $i, j \in [1, n]$. A_{ij} denotes the probability that the random surfer 2.0 moves from page u_j to page u_i.

PageRank: Interpretation

Given a web graph $G = (V, E)$ the following facts hold:

- Let $P(X_t = u)$ be the probability that the random surfer visits page u at step t. The PR of page u is equal to $\lim_{t\to\infty} P(X_t = u)$.
- The PR vector π is the (unique) eigenvector of A with eigenvalue 1.
- The PageRank vector π satisfies $\pi = A\pi$ and $\sum_{i=1}^{n} \pi_i = 1$.

To be proved later...

The PageRank algorithm

Input: A directed graph G with n nodes (Web pages), $0 < \beta < 1, \epsilon > 0$.
Output: The PageRank vector r of the web pages in G.

1: Remove *dead ends* iteratively from G;
2: Build the stochastic matrix M_G (M for short);
3: Let $\pi^{(0)} = [\frac{1}{n}, \ldots \frac{1}{n}]^T$
4: **while** (true) **do**
5: $\quad t = t + 1$;
6: $\quad \pi^{(t)} = A\pi^{(t-1)}$;
7: $\quad$ If $||\pi^{(t)} - \pi^{(t-1)}||_1 < \epsilon$ **break**;
8: **end while**
9: **return** $\pi^{(t)}$.

Efficiency issues

A is a dense $n \times n$ matrix. $n \gg 10^9$ which implies A contains $\gg 10^{18}$ non-zero entries. Lemma 2 allows us to deal with M_G which is sparse.

Lemma 2

Let $x \in \mathbb{R}$ and let $[x]_n$ be the vector with n entries equal to x. Let $G = (V, E)$ a directed graph, let $\beta > 0$, let M_G and A be the $n \times n$ matrices as defined above. For any π with $||\pi||_1 = 1$, we have:

$$A\pi = \beta M_G \pi + \left[\frac{1-\beta}{n}\right]_n$$

Proof of Lemma 2

Proof.

For any $i \in [1, n]$:

$$\begin{aligned}\sum_{j=1}^{n} A_{ij}\pi_j &= \sum_{j=1}^{n} \left(\beta M_{ij} + \frac{1-\beta}{n}\right)\pi_j \\ &= \beta \sum_{j=1}^{n} M_{ij}\pi_j + \frac{1-\beta}{n}\sum_{j=1}^{n}\pi_j \\ &= \beta \sum_{j=1}^{n} M_{ij}\pi_j + \frac{1-\beta}{n}.\end{aligned}$$

The PageRank algorithm (improved)

Input: A directed graph G with n nodes (Web pages), $0 < \beta < 1, \epsilon > 0$.
Output: The PageRank vector π of the web pages in G.

1: Remove *dead ends* iteratively from G;
2: Build the stochastic matrix M_G (M for short);
3: Let $\pi^{(0)} = [\frac{1}{n}, \ldots \frac{1}{n}]^T$
4: **while** (true) **do**
5: $\quad t = t + 1$;
6: $\quad \pi^{(t)} = \beta M \pi^{(t-1)} + [\frac{1-\beta}{n}]_n$;
7: $\quad$ If $||\pi^{(t)} - \pi^{(t-1)}||_1 < \epsilon$ **break**;
8: **end while**
9: **return** $\pi^{(t)}$.

Events and Probability

Consider a stochastic process (e.g. throw a dice, pick a card from a deck)

- Each possible outcome is a *simple event*.
- The sample space Ω is the set of all possible simple events.
- An event is a set of simple events (a subset of the sample space).
- With each simple event E we associate a real number $0 \leq P(E) \leq 1$, which is the probability that event E happens.

Probability Space

Definition 3

A *probability space* has three components:

- A *sample space* Ω, which is the set of all possible outcomes of the random process modeled by the probability space;
- A family of sets $\mathcal{F}$ representing the allowable events, where each set in $\mathcal{F}$ is a subset of the sample space in Ω;
- a *probability function* $P : \mathcal{F} \to \mathbb{R}$, satisfying the definition below (next slide).

Probability Function

Definition 4

A *probability function* is any function $P : \mathcal{F} \to \mathbb{R}$ that satisfies the following conditions:

- for any event E, $0 \leq P(E) \leq 1$;
- $P(\Omega) = 1$;
- for any finite or countably infinite sequence of pairwise mutually disjoint events $E_1, E_2, E_3, \ldots$

$$P\left(\cup_{i \geq 1} E_i\right) = \sum_{i \geq 1} P(E_i). \tag{1}$$

The Union Bound

Theorem 5

$$P(\cup_{i=1}^{n} E_i \leq \sum_{i=1}^{n} P(E_i)). \tag{2}$$

Example: roll a dice:

- let $E_1 =$ "result is odd"
- let $E_2 =$ "result is ≤ 2"

Independent Events

Definition 6

Two events E_1 and E_2 are *independent* if and only if

$$P(E_1 \cap E_2) = P(E_1) \cdot P(E_2) \tag{3}$$

Conditional Probability: Example

What is the probability that a random student at Telecom ParisTech was born in Paris?
E_1 = the event "born in Paris".
E_2 = the event "student at Telecom ParisTech".
The conditional probability that a a student at Telecom ParisTech was born in Paris is written:

$$P(E_1|E_2).$$

Conditional Probability: Definition

Definition 7

The *conditional probability* that event E_1 occurs given that event E_2 occurs is

$$P(E_1|E_2) = \frac{P(E_1 \cap E_2)}{P(E_2)} \tag{4}$$

The conditional probability is only well-defined if $P(E_2) > 0$.

By conditioning on E_2 we restrict the sample space to the set E_2.

Law of Total Probability

Theorem 8

Let $B_1, \ldots, B_k$ be a partition of the sample space Ω, with $P(B_i) > 0$, $i = 1, \ldots, k$. Then, for any event $A \subseteq \Omega$:

$$P(A) = \sum_{i=1}^{k} P(A \cap B_i) = \sum_{i=1}^{k} P(A|B_i)P(B_i). \tag{5}$$

Random Variable

Definition 9

A *random variable* X on a sample space Ω is a function on Ω; that is, $X : \Omega \to \mathbb{R}$.

A *discrete random variable* is a random variable that takes on only a finite number of values.

Examples

In practice, a random variable is some random quantity that we are interested in:

- I roll a die, $X =$ result. E.g. $X = 6$.
- I pick a card, $X = 1$ if card is an Ace, 0 otherwise.
- I roll a dice two times. $X_1 =$ result of the first experiment, $X_2 =$ result of the second experiment. What is $P(X_1 + X_2 = 2)$?

Stochastic Processes

Definition 10

A stochastic process in discrete time $n \in \mathbb{N}$ is a sequence of random variables $X_0, X_1, X_2 \ldots$ denoted by $\mathbf{X} = \{X_n\}$.

We refer to the value X_n as the *state* of the process at time n, with X_0 denoting the initial state.

The set of possible values that each random variable can take is denoted by S. Here, we shall assume that S is finite and $S \subseteq \mathbb{N}$.

Markov Chains

Definition 11

A *stochastic process* $\{X_n\}$ is called a *Markov chain* if for any $n \geq 0$ and any value $j_0, j_1, \ldots, i, j \in S$,

$$P(X_{n+1} = i | X_n = j, X_{n-1} = j_{n-1}, \ldots, X_0 = j_0) = P(X_{n+1} = i | X_n = j),$$

which we denote by P_{ij}.

This can be stated as *the future is independent of the past given the present state*. In other words, the probability of moving to the next state does not depend on what happened in the past. Note that $P_{ij} \neq P_{ji}$.

One-step Transition Matrix

P_{ij} denotes the probability that the chain, whenever in state j, moves next into state i.

The square matrix $\mathbf{P} = (P_{ij})$, $i, j \in S$, is called the *one-step transition matrix*. Note that for each $j \in S$ we have:

$$\sum_{i \in S} P_{ij} = 1. \tag{6}$$

n-step Transition Matrix

The *n-step transition matrix* $\mathbf{P}^{(n)}$, $n \geq 1$, where

$$P_{ij}^{n} = P(X_n = i | X_0 = j) = P(X_{m+n} = i | X_m = j), \quad \forall m \tag{7}$$

denotes the probability that n steps later the Markov chain will be in state i given that at step m is in state j.

Theorem 12

$$\mathbf{P}^{(n)} = \mathbf{P}^n = \mathbf{P} \times \mathbf{P} \times \cdots \times \mathbf{P}, n \geq 1.$$

Stationary Distribution

Definition 13

A probability distribution π over the states of the Markov chain ($\sum_{j \in S} \pi_j = 1$) is called a *stationary distribution*[a] if

$$\pi = P\pi. \tag{8}$$

[a] in literature the transpose of P is often used, in that case $\pi = \pi P$.

Irreducible Markov Chains

Definition 14

A Markov chain is called *irreducible*[a] iff for any $i, j \in S$, there is $n \geq 1$ s.t.

$$P_{ij}^n > 0. \tag{9}$$

[a] definition slightly different when S is not finite.

That is, we can move from any state i to any state j, in one or more steps. If a Markov chain is irreducible then there must be n such that $P_{ii}^n > 0$.

Irreducible Markov Chains

Definition 14

A Markov chain is called *irreducible*[a] iff for any $i, j \in S$, there is $n \geq 1$ s.t.

$$P_{ij}^n > 0. \tag{9}$$

[a] definition slightly different when S is not finite.

That is, we can move from any state i to any state j, in one or more steps. If a Markov chain is irreducible then there must be n such that $P_{ii}^n > 0$.

Theorem 15

If a Markov chain is irreducible, there is a unique stationary distribution.

Aperiodic Markov Chains

A state i has period k if any return to i occurs at step $k \cdot l$, for some $l > 0$. Formally,

$$k = gcd\{n : P(X_n = i | x_0 = i) > 0\} \tag{10}$$

where gcd denotes the *greatest common divisor*. If $k = 1$ then state i is said to be *aperiodic*.

Definition 16

A Markov chain is called *aperiodic* if every state is aperiodic.

Main Theorem

Theorem 17

If a Markov chain is irreducible and aperiodic[a]*, then the Markov chain converges to its (unique) stationary distribution, that is,*

$$\pi_j = \lim_{n\to\infty} P(X_n = j) = \lim_{n\to\infty} P(X_n = j | X_0 = i), \quad \forall i, j \in S. \tag{11}$$

[a] in this case the Markov chain is called *ergodic*

Note: Equation (11) holds for any initial state i of the Markov chain.

Markov chains and the Random Surfer

Consider the Markov chain (MC) of the random surfer.

- What is $P(X_n = j | X_0 = i)$?

Markov chains and the Random Surfer

Consider the Markov chain (MC) of the random surfer.

- What is $P(X_n = j | X_0 = i)$? The probability that at step n the random surfer visits page j given that at step 0 he/she visited page i.
- If there are no random jumps, is the MC irreducible/aperiodic ?

Markov chains and the Random Surfer

Consider the Markov chain (MC) of the random surfer.

- What is $P(X_n = j | X_0 = i)$? The probability that at step n the random surfer visits page j given that at step 0 he/she visited page i.
- If there are no random jumps, is the MC irreducible/aperiodic ? Not necessarily (because of spider traps/dead ends).

Markov chains and the Random Surfer

Consider the Markov chain (MC) of the random surfer.

- What is $P(X_n = j | X_0 = i)$? The probability that at step n the random surfer visits page j given that at step 0 he/she visited page i.
- If there are no random jumps, is the MC irreducible/aperiodic ? Not necessarily (because of spider traps/dead ends).
- What if we add random jumps?

Markov chains and the Random Surfer

Consider the Markov chain (MC) of the random surfer.

- What is $P(X_n = j | X_0 = i)$? The probability that at step n the random surfer visits page j given that at step 0 he/she visited page i.
- If there are no random jumps, is the MC irreducible/aperiodic ? Not necessarily (because of spider traps/dead ends).
- What if we add random jumps? It is both irreducible and aperiodic, which implies that the PageRank vector converges to the unique stationary distribution π of MC (see next slide).

The Random Surfer and its Stationary Distribution

Fact: The stationary distribution π of the MC is the PageRank vector!

The Random Surfer and its Stationary Distribution

Fact: The stationary distribution π of the MC is the PageRank vector!

Sketch: At each step n of the PageRank algorithm we compute $\pi^{(n)} = P\pi^{(n-1)} = P^n\pi^{(0)} = P^{(n)}\pi^{(0)}$. From the law of total probability (Theorem 8) and from the fact that $\pi^{(n)} = P^{(n)}\pi^{(0)}$ it follows that $\pi_j^{(n)} = P(X_n = j)$, $\forall j$, which converges to π (Theorem 17).

Therefore, the PageRank algorithm computes for every page j the probability that the random surfer visits page j, i.e. $\lim_{n\to\infty} P(X_n = j)$.

References I

Page, Lawrence and Brin, Sergey and Motwani, Rajeev and Winograd, Terry.

The PageRank citation ranking: bringing order to the web

Stanford InfoLab, (1999).

Brin, Sergey, and Lawrence Page.

The anatomy of a large-scale hypertextual web search engine.

Computer networks, 56.18 (2012): 3825-3833.