CentraleSupelec 2018-2019 MSC DSBA / DATA SCIENCES

Big Data Algorithms, Techniques and Platforms

Distributed Computing with MapReduce and Hadoop: PageRank

Hugues Talbot & Céline Hudelot, professors.

Plan

1. Page Rank Algorithm

Page Rank : the origin

Citation analysis in scientific litterature

- Example : "Miller (2001) has shown that physical activity alters the metabolism of estrogens.'"
- "Miller (2001)" can be seen as an hyperlink between two scientific articles.

Origin of the page rank : *Pinsker and Narin, 1960s*

Page Rank

Page Rank : principle

- Web = oriented graph
- the links are important information
  - a link between two pages = a relation of relevance
  - the anchoring text of a link is a summary of the content of the targeted page : the anchoring text is used during indexation

page rank : an algorithm to compute weighted citations in the web

Page Rank

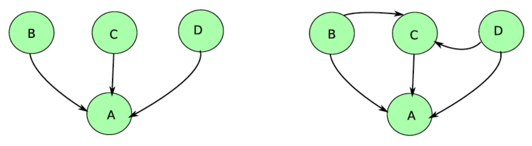
PageRank : idea

- a guy doing a random walk in the web graph
    - It start from a random node (a random page)
    - A each step, he can go out the current page by following an out link (equiprobable)
- At a time, we reach a stationnary state which represents the probability to reach each visited page : this value is the **Page Rank** or **steady state probability** or **long-term visit rate**.
- Modeling as a Markov chain

The pages which are the most visited as the pages which have many in-links

Page Rank

PageRank : idea

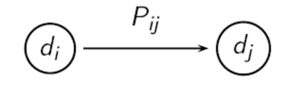


- Page rank PR of a page i = probability that the walker reaches i at a moment.
- Left figure : $PR(A) = PR(B) + PR(C) + PR(D)$ (he will reach A if he reaches B, C or D in time $i - 1$.
- Right figure : $PR(C) = \frac{1}{2}PR(B) + \frac{1}{2}PR(D)$ (probability of the node B (or D) to go to C is $\frac{1}{2}$).

Link analysis

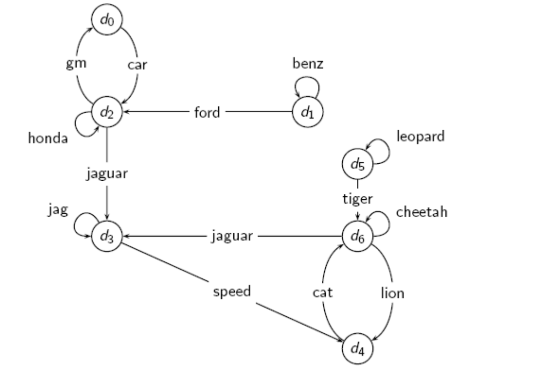
PageRank : Markov chain

- A Markov chain consists in n states and a transition matrix P of size nxn
- a state = a page
- At each step, we are exactly on one page
- For $1 \leq i, j \leq n$, the element P_{ij} of the matrix represents the probability of j to be the next page considering that i is the current page.
- For all i, $\sum_{j=1}^{n} P_{ij} = 1$



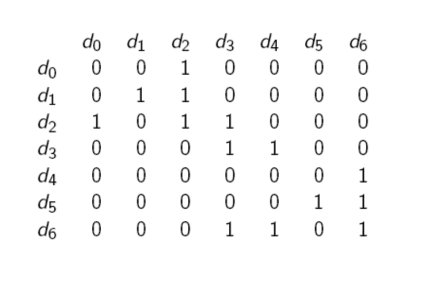
Graph example

Example



Graph example

Example : matrix of links



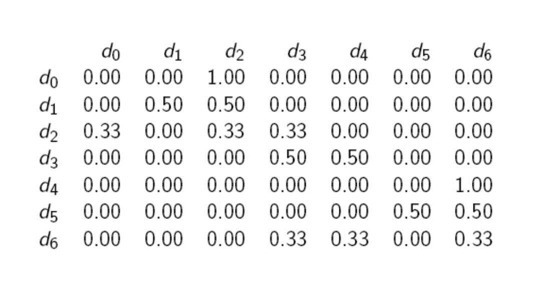
	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0	0	1	0	0	0	0
d_1	0	1	1	0	0	0	0
d_2	1	0	1	1	0	0	0
d_3	0	0	0	1	1	0	0
d_4	0	0	0	0	0	0	1
d_5	0	0	0	0	0	1	1
d_6	0	0	0	1	1	0	1

Graph example

Example : transition matrix

Let $G = (g_{ij})$ be the transition matrix of the graph.

- $g_{ij} = 0$ if no link between i and the page j.
- $g_{ij} = \frac{1}{n_i}$ else with n_i the number of out-links of the page i.



	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0.00	0.00	1.00	0.00	0.00	0.00	0.00
d_1	0.00	0.50	0.50	0.00	0.00	0.00	0.00
d_2	0.33	0.00	0.33	0.33	0.00	0.00	0.00
d_3	0.00	0.00	0.00	0.50	0.50	0.00	0.00
d_4	0.00	0.00	0.00	0.00	0.00	0.00	1.00
d_5	0.00	0.00	0.00	0.00	0.00	0.50	0.50
d_6	0.00	0.00	0.00	0.33	0.33	0.00	0.33

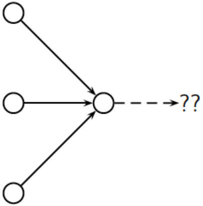
Stochastic matrix : each row has for sum the number 1.

Page Rank

Consequences for the graph

- the page rank is the probability of the walker to be on the page *d* at a given time.
- What are the properties of the graph to have a well-defined page rank ?
- An ergodic markov chain : i.e. each state can be reach from each other state.
- consequence : no dead ends. This is not the case of the web

Page rank : dead ends



Page rank : to solve the dead end problem

Principle

- At each *dead end*, skip to a random page with the probability $\frac{1}{N}$ (N number of nodes).
- At each other step :
  - Skip to a random page with a probability p
  - With the remaining probability $1 - p$ chose one of the out-links of the page with equal probability
- p is the teleportation rate.
- it makes the web graph ergodic
  - There is a path between any two nodes of the graph.

Page rank :how to compute it ?

Some formalisation

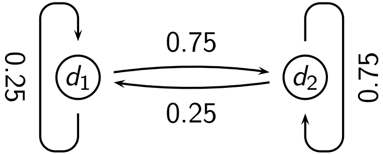
- A vector (row) of probability $\vec{x} = (x_1, ...x_n)$ gives the position of the walker at a time
- We consider that the walker is at the position i with the probability x_i
- Exemple 1 : $\begin{pmatrix} 0 & 0 & 0 & \dots & 1 & \dots & 0 & 0 & 0 \\ 1 & 2 & 3 & \dots & i & \dots & \text{N-2} & \text{N-1} & \text{N} \end{pmatrix}$
- Exemple 2 : $\begin{pmatrix} 0.05 & 0.01 & 0.0 & \dots & 0.2 & \dots & 0.01 & 0.05 & 0.03 \\ 1 & 2 & 3 & \dots & i & \dots & \text{N-2} & \text{N-1} & \text{N} \end{pmatrix}$
- We have $\sum x_i = 1$
- How to go to the next step ?

Page rank :how to compute it ?

- We use the transition matrix which row i informs on where to go after node i
- Next step : $\vec{x}P$
- We have $\vec{\pi} = (\pi_1, ...\pi_n)$ probability vector with π_i the page rank of page i.
- We search for $\vec{\pi} = \vec{\pi}P$ (steady state)

Steady-state : Example

- PageRank / steady state of this example ?



Steady-state : Example

	x_1	x_2		
	$P_t(d_1)$	$P_t(d_2)$		
			$P_{11} = 0.25$	$P_{12} = 0.75$
			$P_{21} = 0.25$	$P_{22} = 0.75$
t_0	0.25	0.75	0.25	0.75
t_1	0.25	0.75	(convergence)	

PageRank vector $= \vec{\pi} = (\pi_1, \pi_2) = (0.25, 0.75)$

$$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$$

$$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$$

Page rank computation

The power approach

- We start by an uniform distribution $\vec{x}$
- After 1 step, we are in $\vec{x}P$
- After 2 steps, we are in n $\vec{x}P^2$
- ...
- We multiply $\vec{x}$ by power of P until a stationary state is reached.

Power method.

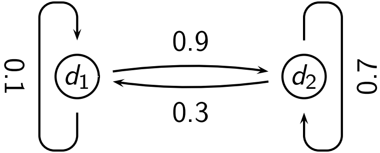
Page rank computation

The power method

- With two nodes : $\vec{x} = (0.5, 0.5)$, $P = \begin{pmatrix} 0.25 & 0.75 \\ 0.25 & 0.75 \end{pmatrix}$
- $\vec{x}P = (0.25, 0.75)$
- $\vec{x}P^2 = (0.25, 0.75)$
- Convergence from the first iteration

Example

page rank ?



We have for page rank 0.25 for d_1 and 0.75 for d_2.

Page rank computation : the power method

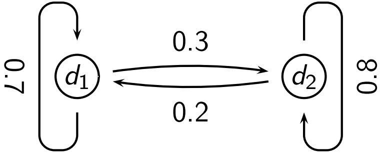
	x_1	x_2			
	$P_t(d_1)$	$P_t(d_2)$			
			$P_{11} = 0.1$	$P_{12} = 0.9$	
			$P_{21} = 0.3$	$P_{22} = 0.7$	
t_0	0	1	0.3	0.7	$= \vec{x}P$
t_1	0.3	0.7	0.24	0.76	$= \vec{x}P^2$
t_2	0.24	0.76	0.252	0.748	$= \vec{x}P^3$
t_3	0.252	0.748	0.2496	0.7504	$= \vec{x}P^4$
				$\dots$	$\dots$
t_∞	0.25	0.75	0.25	0.75	$= \vec{x}P^\infty$

PageRank vector $= \vec{\pi} = (\pi_1, \pi_2) = (0.25, 0.75)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Other example



Solution

	x_1 $P_t(d_1)$	x_2 $P_t(d_2)$		
			$P_{11} = 0.7$	$P_{12} = 0.3$
			$P_{21} = 0.2$	$P_{22} = 0.8$
t_0	0	1	0.2	0.8
t_1	0.2	0.8	0.3	0.7
t_2	0.3	0.7	0.35	0.65
t_3	0.35	0.65	0.375	0.625
				...
t_∞	0.4	0.6	0.4	0.6

PageRank vector $= \vec{\pi} = (\pi_1, \pi_2) = (0.4, 0.6)$

$P_t(d_1) = P_{t-1}(d_1) * P_{11} + P_{t-1}(d_2) * P_{21}$

$P_t(d_2) = P_{t-1}(d_1) * P_{12} + P_{t-1}(d_2) * P_{22}$

Page Rank : in brief

Processing

- Building of the matrix P
- Teleportation principle
- From the modified matrix, we can compute $\vec{\pi}$
- π_i is the page rank of the page i

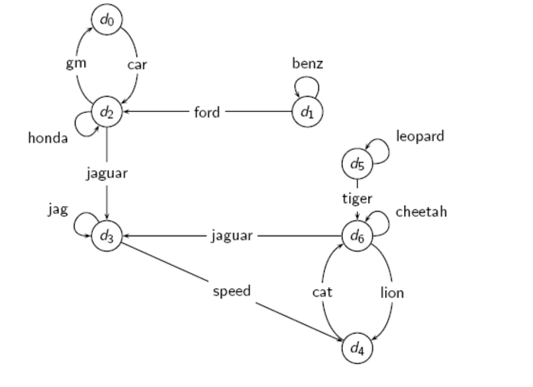
Page rank : teleportation

Principle : building of the matrix

- From the adjacency matrix, we obtain the transition matrix
- Teleportation : multiplication of the matrix by $1 - \alpha$
- We add $\frac{\alpha}{N}$ to the matrix to obtain P.
- We have $R(u) = (1 - \alpha) \sum_{v \in B_u} \frac{R(v)}{|N_v|} + \frac{\alpha}{N}$ with u, v some web pages, B_u the set of the pages that go to u, N_v the set of out-links from v

Graph example

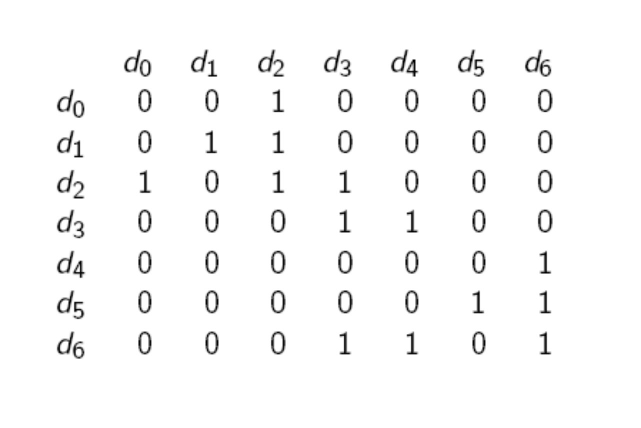
Example



page rank of d_2 ¡ page rank of d_6. Why ?

Graph example

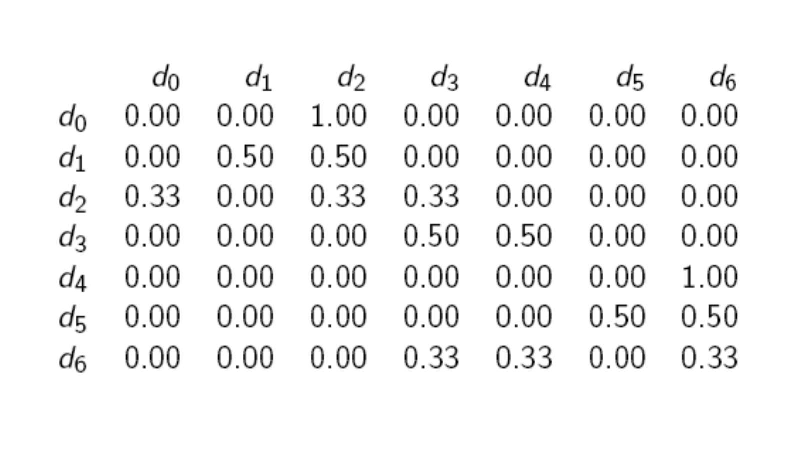
Example : adjacency matrix



$$\begin{array}{cccccccc}
 & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\
d_0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
d_1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
d_2 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
d_3 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
d_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
d_5 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
d_6 & 0 & 0 & 0 & 1 & 1 & 0 & 1
\end{array}$$

Graph example

Example : transition matrix

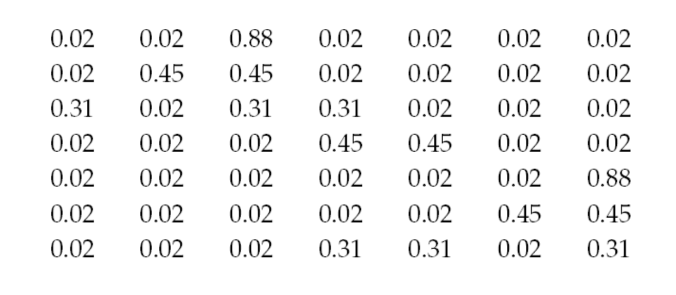


	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0.00	0.00	1.00	0.00	0.00	0.00	0.00
d_1	0.00	0.50	0.50	0.00	0.00	0.00	0.00
d_2	0.33	0.00	0.33	0.33	0.00	0.00	0.00
d_3	0.00	0.00	0.00	0.50	0.50	0.00	0.00
d_4	0.00	0.00	0.00	0.00	0.00	0.00	1.00
d_5	0.00	0.00	0.00	0.00	0.00	0.50	0.50
d_6	0.00	0.00	0.00	0.33	0.33	0.00	0.33

Page rank : teleportation

Principle : building of the matrix with $\alpha = 0.14$

$$(1-\alpha)P + \frac{\alpha}{N}$$



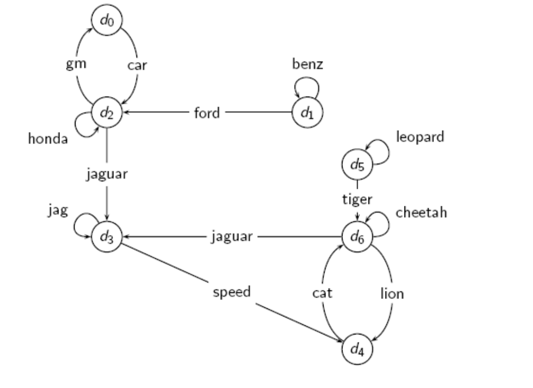
$$\begin{matrix}
0.02 & 0.02 & 0.88 & 0.02 & 0.02 & 0.02 & 0.02 \\
0.02 & 0.45 & 0.45 & 0.02 & 0.02 & 0.02 & 0.02 \\
0.31 & 0.02 & 0.31 & 0.31 & 0.02 & 0.02 & 0.02 \\
0.02 & 0.02 & 0.02 & 0.45 & 0.45 & 0.02 & 0.02 \\
0.02 & 0.02 & 0.02 & 0.02 & 0.02 & 0.02 & 0.88 \\
0.02 & 0.02 & 0.02 & 0.02 & 0.02 & 0.45 & 0.45 \\
0.02 & 0.02 & 0.02 & 0.31 & 0.31 & 0.02 & 0.31
\end{matrix}$$

Power method $\vec{x}P^k$

	$\vec{x}$	$\vec{x}P^1$	$\vec{x}P^2$	$\vec{x}P^3$	$\vec{x}P^4$	$\vec{x}P^5$	$\vec{x}P^6$	$\vec{x}P^7$	$\vec{x}P^8$	$\vec{x}P^9$	$\vec{x}P^{10}$	$\vec{x}P^{11}$	$\vec{x}P^{12}$	$\vec{x}P^{13}$
d_0	0.14	0.06	0.09	0.07	0.07	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05
d_1	0.14	0.08	0.06	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
d_2	0.14	0.25	0.18	0.17	0.15	0.14	0.13	0.12	0.12	0.12	0.12	0.11	0.11	0.11
d_3	0.14	0.16	0.23	0.24	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.25	0.25	0.25
d_4	0.14	0.12	0.16	0.19	0.19	0.20	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
d_5	0.14	0.08	0.06	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
d_6	0.14	0.25	0.23	0.25	0.27	0.28	0.29	0.29	0.30	0.30	0.30	0.30	0.31	0.31

Graph example

Example



page rank of d_2 ¡ page rank of d_6

Page rank : from the web scale

Some principles

- Sparse transition matrix : efficient representation of the matrix.
- Distributed computing, partition into blocks of the matrix.
- Google Pregel [a], Apache Giraph [b]

[a] https://kowshik.github.io/JPregel/pregel_paper.pdf

[b] http://giraph.apache.org/

Page rank : some ressources

- Article of Sergey Brin and Lawrence Page : The Anatomy of a Large-Scale Hypertextual Web Search.
  `http://infolab.stanford.edu/pub/papers/google.pdf`
- some web sites :
  - `http://www.ams.org/samplings/feature-column/fcarc-pagerank`
  - `http://www.sirgroane.net/google-page-rank/`
  - Page rank computing :
    `http://www.webworkshop.net/pagerank_calculator.php`
- Some articles :
  - Kurt Bryan, Tanya Leise, The $25,000,000,000$ eigenvector. The linear algebra behind Google. SIAM Review, 48 (3), 569-81. 2006
    `http://www.rose-hulman.edu/~bryan/google.html`
  - Taher Haveliwala, Sepandar Kamvar, The second eigenvalue of the Google matrix. (`http://kamvar.org/publications`)

Page rank : with map reduce ?