PageRank

definition

a way of measuring the importance of website pages.

works by counting the **number** and **quality** of links to a page, assuming that the more important a page is, the more likely it would **receive links** from other website.

numerical weight of element E: PR(E)

influenced by **citation analysis** and **Hyper Search**.

importance

developed in 1996 as part of a research about a new kind of search machine.[1]

order websites in a hierarchy by "link popularity".

initial prototype of the Google search engine.

algorithm

outputs a probability distribution ~ the likelihood that a person randomly clicking on links will ultimately arrive at any particular page.

the distribution is evenly divided at the beginning of the computational process.

require several "iterations" to adjust approximate PR values to a close reflection of theoretical PR value.

- pre-settings
  - ignore self-linking $L_{i \to i}$
  - multiple $L_{i \to j}$ == a single link
  - sink: a page with no outbound links $\rightarrow$ pick another URL at random $\rightarrow$ link out to all other pages.
- PageRank for A,B,C,D
  - initial probability distribution

PR(A)	PR(B)	PR(C)	PR(D)
0.25	0.25	0.25	0.25

  - iteration

    - PageRank trnasfered from a outbound link: the linking document's score divided by number of its outbound links **L( )**.
    - PR(A)=$\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}$
  - damping factor d
    - the probability, at any step, that the random visitor will continue
    - the residual probability is estimated from the frequency that an average surfer uses the bookmark feature.
    - it is generally assumed that d $\approx$ 0.85.[2]
    - PR(A)=$\frac{1-d}{N} + d(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)})$
    - N denotes the number of documents
    - Page and Brin confused this formula in their most popular paper "The Anatomy of a Large-Scale Hypertextual Web Search Engine".
- connection to Markov Chain
  - states $\rightarrow$ pages
  - transition $\rightarrow$ links, all equally probable
  - stochastic row vector $R = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ \vdots \\ PR(p_n) \end{bmatrix}$
  - transition rate matrix

$$R_{new} = \begin{bmatrix} \frac{1-d}{N} \\ \frac{1-d}{N} \\ \vdots \\ \frac{1-d}{N} \end{bmatrix} + d \begin{bmatrix} l(p_1,p_1) & l(p_1,p_2) & \cdots & l(p_1,p_n) \\ l(p_2,p_1) & \ddots & & \vdots \\ \vdots & & l(p_i,p_j) & \\ l(p_n,p_1) & \cdots & & l(p_n,p_n) \end{bmatrix} R$$

  - adjacency function $l_a(p_i,p_j) = \begin{cases} \frac{L_{j\rightarrow i}(j)}{L(j)}, L_{j\rightarrow i}(j) \neq 0 \\ 0, L_{j\rightarrow i}(j) = 0 \end{cases}$
  - $l(p_i,p_j)$ is obtained by normalizing $l_a(p_i,p_j)$, such that

$$\sum_{i=1}^{N} l(p_i,p_j) = 1$$

  - a variant of the eigenvector centrality measure used commonly in network analysis.
  - a large eigengap of the modified adjacency matrix, values of PageRank eigenvector can be approximated to within a high degree of accuracy within only a few iterations.[3]
    time complexity is roughly linear in O(log n).
  - As a result of Markov theory, the probability of arriving at A after a large number of clicks —— PR(A) = t^{-1}

where t is the expectation of the number of clicks (or random jumps) required to get from A back to itself.

- strategies to accelerate the compution of PageRank.[4]
- disadvantage
  - favor old pages, since new pages may not have many links
  - needs further improving search results rankings and monetizing advertising links

computation

- `iterative`
  - t=0, $PR(p_i;0)=\frac{1}{N}$
  - at each step $t \to t+1$
    - $PR(p_i;t+1)=\frac{1-d}{N}+d\sum_{pj\in M(p_i)}\frac{PR(p_j;t)}{L(p_j)}$, $M(p_i)$ is the set of pages that link to p_i.
    - matrix notation R
      $R(t+1)=dMR(t)+\frac{1-d}{N}$ * **1**,
      1 is the column vector of length N containing only ones,

$$M_{ij}=\begin{cases}\frac{1}{L(p_j)}, & \text{if j links to i}\\ 0, & \text{otherwise}\end{cases}$$

  - computation ends when |R(t+1)-R(t)| < ϵ
- `algebraic`
  - $t\to\infty$, $R=dMR+\frac{1-d}{N}$ * **1**
    - R=$(\mathbf{I}-dM)^{-1}\frac{1-d}{N}$ * **1**
    - M is a stochastic matrix $\to$ Perron-Frobenius theorem: $\exists$ eigenvalue = 1 $\to$ the solution exists and is unique for 0 < d < 1.
- `power method`
  - transition probability M : column-stachastic
    if a column of M has only 0 values (L(p_i)=0), replace this column with $\frac{1}{N}\mathbf{1}$.
  - probability distribution R : |R|=1, **E**R=**1**
  - R = (dM+$\frac{1-d}{N}$**E**) R := PR, $\Rightarrow$ R is the principle eigenvector of P.
  - apply power method procedure
    - starting with an arbitrary vector x(0)
    - x(t+1) = ||Px(t)||
    - iterate until |x(t+1)-x(t)| < ϵ
- `verifying anwsers`

$$\mathbf{R}_{power}=||\mathbf{R}_{iterative}||=||\mathbf{R}_{algebraic}||$$

Matlab

```
% Parameter M adjacency matrix where M_i,j represents the link from 'j' to 'i', such th
%     sum(i, M_i,j) = 1
% Parameter d damping factor
% Parameter v_quadratic_error quadratic error for v
% Return v, a vector of ranks such that v_i is the i-th rank from [0, 1]

function [v] = rank2(M, d, v_quadratic_error)

N = size(M, 2); % N is equal to either dimension of M and the number of documents
v = rand(N, 1);
v = v ./ norm(v, 1);   % This is now L1, not L2
last_v = ones(N, 1) * inf;
M_hat = (d .* M) + (((1 - d) / N) .* ones(N, N));

while(norm(v - last_v, 2) > v_quadratic_error)
        last_v = v;
        v = M_hat * v;
        % removed the L2 norm of the iterated PR
end

end %function
```

topics

- identify the most important page (the most likely page, which a random visitor would eventually arrive at) among UMJI official websites.

source

> Page, Larry, "PageRank: Bringing Order to the Web". Archived from the original on May 6, 2002. Retrieved 2016-09-11., Stanford Digital Library Project, talk. August 18, 1997 (archived 2002)
> Brin, S.; Page, L. (1998). "The anatomy of a large-scale hypertextual Web search engine". Computer Networks and ISDN Systems. 30 (1–7): 107–117. CiteSeerX 10.1.1.115.5930. doi:10.1016/S0169-7552(98)00110-X. ISSN 0169-7552. Archived from the original on 2015-09-27.
> Taher Haveliwala & Sepandar Kamvar (March 2003). "The Second Eigenvalue of the Google Matrix". Stanford University Technical Report: 7056. arXiv:math/0307056. Bibcode:2003math.....7056N. Archived from the original on 2008-12-17.
> Gianna M. Del Corso; Antonio Gullí; Francesco Romani (2005). "Fast PageRank Computation via a Sparse Linear System". Internet Mathematics. Lecture Notes in Computer Science. 2. pp.

118–130. CiteSeerX 10.1.1.58.9060. doi:10.1007/978-3-540-30216-2_10. ISBN 978-3-540-23427-2. Archived from the original on 2014-02-09.