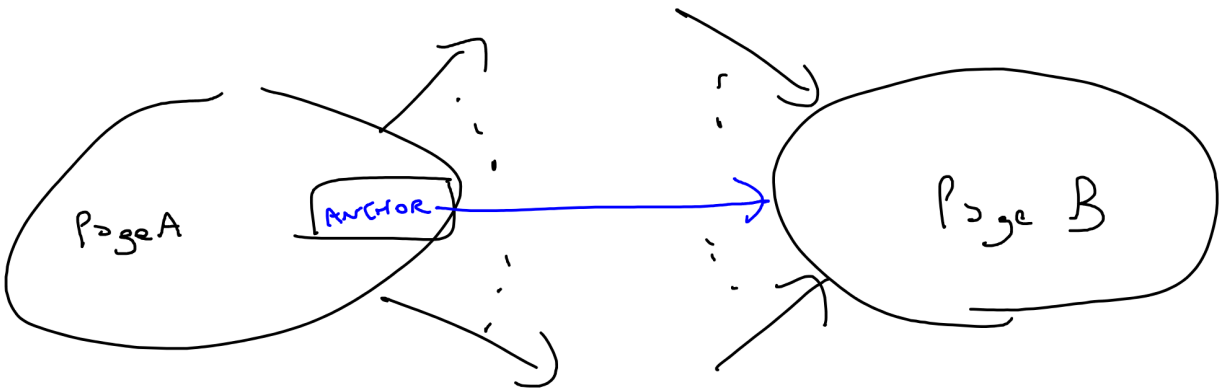
Link-based ranking

Second generation, used by modern web search engines, which ranks a document (actually, since we are talking about the web, we talk about web nodes) based on the hyperlinks that represent the web graph.

The assumptions are that an hyperlink between two pages (web nodes) denotes a perceived relevance by the author (QUALITY SIGNAL) and that the text in the anchor of the hyperlink describes the target page (TEXTUAL CONTEXT)



USS

When indexing a webpage, you should also include the anchor texts pointing to it, because sometimes the publishers don't provide an accurate description of themselves (marketing strategy?)

EXAMPLE

The IBM homepage doesn't contain the term "computer" anywhere, but the pages citing it may contains more useful informations

OSS of USS

You should also add a weight representing the trust factor of the anchor page's website, because it might use wrong descriptions to associate the cited page with bad terms

Under the assumptions shown above, a random walker that traverses the web graph would step over some nodes more often than others, based on the number of links pointing to them: The idea of web ranking functions such as PageRank is that the pages visited more often in this walk are more important

How do we model this random walk on a graph?

MARKOV CHAINS

Given a Graph

- Adjacency matrix A = represent the graph in memory

$$a_{ij} \begin{cases} 0 & \text{no link } i \to j \\ 1 & \text{link } i \to j \end{cases}$$

- Transition probability matrix P = represent the probability of moving from a state (node) to another

$$P_{ij} = Pr(j|i) = \text{probability of moving } i \to j \in [0,1]$$

NOTE = the total transition probability from a node i to all the other nodes must be 1

$$\forall i \quad \sum_j P_{ij} = 1$$

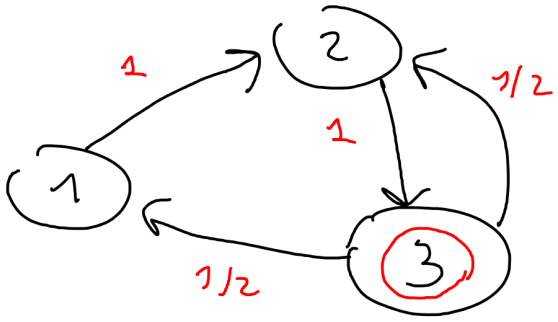
P IS STOCHASTIC MATRIX

The probability distribution of the next state (node) depends only on the current state (node) and not on how the Markov chain (graph) arrived at that current state (node)

- $X_t(i)$ = probability that the random surfer is at node i at time t

- $X_{t+1}(i) = \sum_j X_t(j) \cdot P_{ji} = X_t \cdot P =$
$= X_{t-1} \cdot P \cdot P =$
$= \dots =$
$= X_0 \cdot P^{t+1}$

Example



I'm at node 3 at time t (X_t known), compute X_{t+1}

$$X_{t+1} = X_t \cdot P = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1/2 & 1/2 & 0 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 1/2 \\ 0 \end{bmatrix}$$

And I would be able to compute it even if I was given only the starting position (X_0)

We talk of stationary probability distribution when the process (the random walk) goes on for a long time and the distribution does not change

Left eigenvector of eigenvalue 1

$$X_{t+1} = X_t \implies X_t P = 1 \cdot X_t$$

Therefore, it exists a stationary state probability vector π, such that

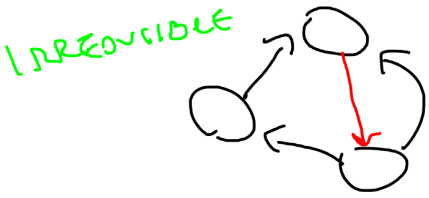
$$\lim_{t \to +\infty} x_0 \cdot P^t = \pi$$

Does a stationary distribution always exist? Is it unique?

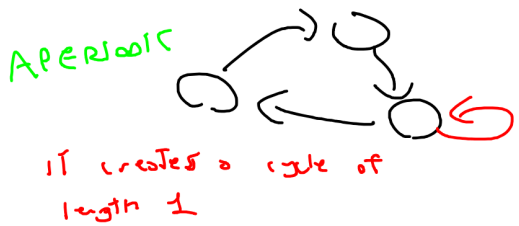
Yes, but only if the graph is "well-behaved", which means that the Markov chain is irreducible and aperiodic

- irreducible = A graph is irreducible if there is a path from every node to every other node

EXAMPLES = strongly connected direct graph, connected undirected graph



- Aperiodic = A graph where the length of every cycle is not divisible by $K > 1$ (The Greatest Common Divisor is 1)



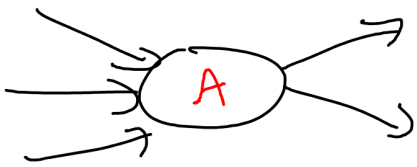
We can finally talk about how we can score web pages using links

1° generation

Simply count the links to measure the popularity of a webpage. In particular there are two types of popularity:

- undirected popularity, where each page gets a score given by the number of in-links plus the number of out-links

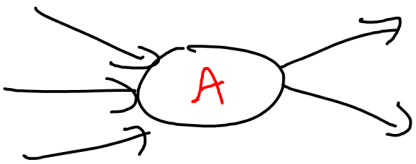
EXAMPLE



$r(A) = 5$

- directed popularity, where the score is given only on the number of in-links

EXAMPLE



$r(A) = 3$

IT'S VERY EASY TO TRICK THIS RANKING METHOD: JUST CREATES PAGES FULL OF LINKS TO MAKE IT RANK HIGHER. ALSO, IT DOESN'T EXPLOIT THE GRAPH STRUCTURE EXPLAINED ABOVE

2° Generation: PageRank

To overcome the problem, every link is assigned with an importance value, which depends on how much is "valuable" the pointing web page, which is determined by how many nodes link to it, and the pagerank will be computed using them.

As you can see, we can apply the random walk idea and model the web graph as a Markov chain

HOWEVER WE ARE INTERESTED IN THE STATIONARY PROBABILITY DISTRIBUTION, BECAUSE WE DON'T WANT OUR RANKING TO CHANGE AT EVERY TIME STEP

But does the web guarantees it?

Actually NO, because the web has OUT nodes with no hyperlinks in it, therefore it is not irreducible

BUT GOOGLE SOLVED THE PROBLEM INTRODUCING THE RANDOM JUMPS

This way we transformed the web graph in a strongly connected graph: from anywhere you might go everywhere.

OSS: If you think about it, if you are "stuck" in a page with no hyperlinks, you can just type a new URL in the URL box and get away with it

$$r(i) = \alpha \cdot \sum_{j \in B(i)} \frac{r(j)}{\#out(j)} + (1-\alpha)\frac{1}{N}$$

- $r(i)$ = pagerank of i
- N = total number of pages in the web graph
- $B(i)$ = set of pages linking to i
- $\#out(j)$ = number of out-links from j
- α = damping factor

RANDOM JUMP (OR TELEPORTATION STEP)

NOTE IF WE IMAGINE THAT EACH NODE CONTAINS A TEXT CONTENT, WE MIGHT WANT TO USE THE SIMILARITY $Sim(T_i, T_j)$ INSTEAD OF $\#out(j)$ TO REPRESENT THE RELEVANCE OF node i

NOTE WE ASSUME RANDOM JUMP WITH UNIFORM RANDOM PROBABILITY $p(\text{"jump to node } i") = \frac{1}{\#nodes} = \frac{1}{N}$

principal eigenvector $r = [\alpha \Pi^T + (1-\alpha) e e^T] \cdot r$

where $\Pi_{i,j} = \begin{cases} 1/\#out(i) & \text{if } i \to j \\ 0 & \text{else} \end{cases}$

MANCA ROBA PER CAPIRE ESATTAMENTE CHE SUCCEDE

PageRank in web searches

- PRE-PROCESSING PHASE = Given the Web graph, build the transition prob matrix P and then compute its principal eigenvector π:

  $\pi(i)$ is the pagerank of page i.

  Since computing π is extremely costly, it is usually approximated as $\boxed{\pi = e^* P^t}$,

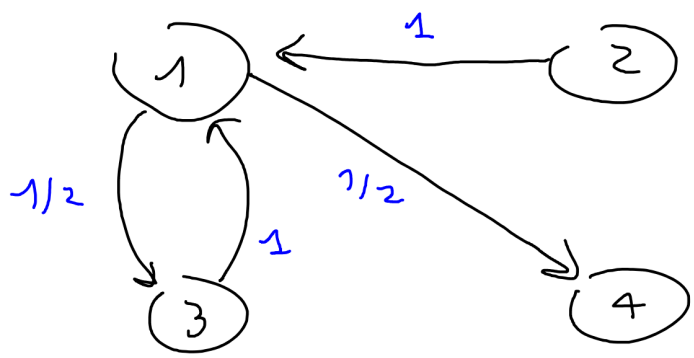
  where $t = 0, 1, \ldots$ is the number of steps and $e = \left[\frac{1}{N} \; \frac{1}{N} \; \ldots\right]$ is the starting prob.

  The personalized PageRank let you decide the prefered jump pages by substituting e with a preference vector: this way you bias the jumps towards certain pages.

- QUERY PHASE = Retrieve the pages containing the query terms and rank them by their Pagerank (the ranking is completely independent from the query!)

Exercise 4

Given a directed graph G with edges $\{(1,3), (3,1), (1,4), (2,1)\}$, simulate the execution of one step of PR algorithm, starting with PR = 1 (unnormalized) every node and assuming a teleportation step that jumps only to node 3.



normalized

$R(1) = 1/4$

$R(2) = 1/4$

$R(3) = 1/4$

$R(4) = 1/4$

e = teleportation vector = $(0\ 0\ 1\ 0)$

$\alpha = 1/2$ (since the text doesn't say anything)

$$R(1) = \frac{1}{2}\left(\frac{\frac{1}{4}}{1} + \frac{\frac{1}{4}}{1}\right) + \frac{1}{2}\cdot 0 = \frac{1}{4}$$

$$R(2) = \frac{1}{2}\cdot 0 + \frac{1}{2} = 0$$

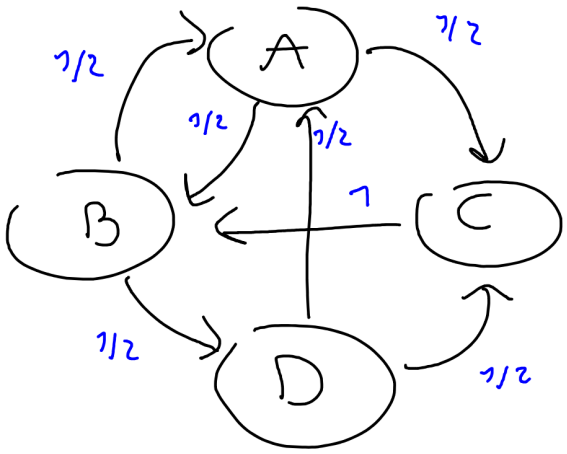
$$R(3) = \frac{1}{2}\left(\frac{\frac{1}{4}}{2}\right) + \frac{1}{2}\cdot 1 = \frac{1}{16} + \frac{1}{2} = \frac{9}{16}$$

$$R(4) = \frac{1}{2}\left(\frac{\frac{1}{4}}{2}\right) + \frac{1}{2}\cdot 0 = \frac{1}{16}$$

Exercise 2

Given the directed graph G of edges $\{(B,A),(A,C),(D,C),(A,B),(C,B),(D,A),(B,D)\}$: compute two steps of PageRank assuming uniform starting probability and $\alpha = 3/4$ (teleportation step with 1/4); compute one step of personalized PR for the node A assuming starting probability $[1,0,0,0]$ and $\alpha = 1/2$.

(1)



$r(A) = 1/4$

$r(B) = 1/4$

$r(C) = 1/4$

$r(D) = 1/4$

step 1

$$r(A) = \frac{3}{4}\left(\frac{\frac{1}{4}}{2} + \frac{\frac{1}{4}}{2}\right) + \frac{1}{4}\cdot\frac{1}{4} = \frac{3}{4}\cdot\frac{1}{4} + \frac{1}{16} = \frac{1}{4}$$

$$r(B) = \frac{3}{4}\left(\frac{\frac{1}{4}}{2} + \frac{\frac{1}{4}}{1}\right) + \frac{1}{4}\cdot\frac{1}{4} = \frac{3}{4}\cdot\frac{3}{8} + \frac{1}{16} = \frac{11}{32}$$

$$r(C) = \frac{3}{4}\left(\frac{\frac{1}{4}}{2} + \frac{\frac{1}{4}}{2}\right) + \frac{1}{4}\cdot\frac{1}{4} = \frac{1}{4}$$

$$r(D) = \frac{3}{4}\left(\frac{\frac{1}{4}}{2}\right) + \frac{1}{4}\cdot\frac{1}{4} = \frac{3}{4}\cdot\frac{1}{8} + \frac{1}{16} = \frac{5}{32}$$

step 2

$$\pi(A) = \frac{3}{4}\left(\frac{\frac{11}{32}}{2} + \frac{\frac{5}{32}}{2}\right) + \frac{1}{16} = \frac{3}{4}\cdot\frac{1}{4} + \frac{1}{16} = \frac{1}{4}$$

$$\pi(B) = \frac{3}{4}\left(\frac{\frac{1}{4}}{2} + \frac{\frac{1}{4}}{1}\right) + \frac{1}{16} = \frac{11}{32}$$

$$\pi(C) = \frac{3}{4}\left(\frac{\frac{1}{4}}{2} + \frac{\frac{5}{32}}{2}\right) + \frac{1}{16} = \frac{3}{4}\cdot\frac{13}{64} + \frac{1}{16} = \frac{55}{256}$$

$$\pi(D) = \frac{3}{4}\left(\frac{\frac{11}{32}}{2}\right) + \frac{1}{16} = \frac{3}{4}\cdot\frac{11}{64} + \frac{1}{16} = \frac{99}{256}$$

(2)

SAME GRAPH, BUT

$\pi(A) = 1$

$\pi(B) = \pi(C) = \pi(D) = 0$

THEREFORE THE TELEPORTATION STEP CAN ONLY REACH NODE A (AT THE START)

step 1

$$\pi(A) = \frac{1}{2}\left(\frac{0}{2} + \frac{0}{2}\right) + \frac{1}{2}\cdot 1 = \frac{1}{2}$$

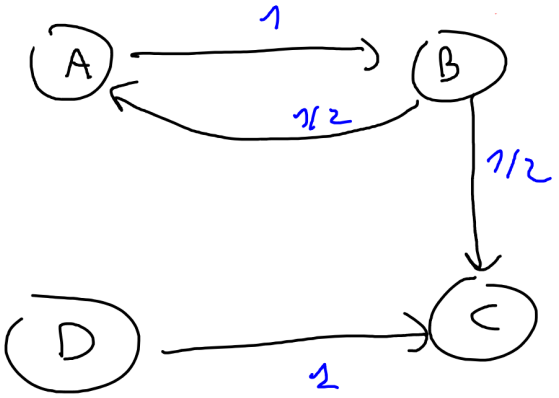
$$\pi(B) = \frac{1}{2}\left(\frac{1}{2} + \frac{0}{2}\right) + \frac{1}{2}\cdot 0 = \frac{1}{4}$$

$$\pi(C) = \frac{1}{2}\left(\frac{1}{2} + \frac{0}{2}\right) + \frac{1}{2}\cdot 0 = \frac{1}{4}$$

$$\pi(D) = \frac{1}{2}\left(\frac{0}{2}\right) + \frac{1}{2}\cdot 0 = 0$$

Exercise 3

Given the following graph



How can you estimate the similarity of B with all the other nodes, using Personalized PageRank? Apply your alg. for one step with $\alpha = \frac{1}{2}$.

Since we are not given a starting distribution, put it equal to the jump vect. $e = (0 \ 1 \ 0 \ 0)$

$$r(A) = \frac{1}{2}\left(\frac{1}{2}\right) + \frac{1}{2}\cdot 0 = \frac{1}{4}$$

$$r(B) = \frac{1}{2}\left(\frac{0}{1}\right) + \frac{1}{2}\cdot 1 = 1/2$$

$$r(C) = \frac{1}{2}\left(\frac{0}{1} + \frac{1}{2}\right) + \frac{1}{2}\cdot 0 = 1/4$$

$$r(D) = \frac{1}{2}\cdot 0 + \frac{1}{2}\cdot 0 = 0$$

The most similar pages are A and C

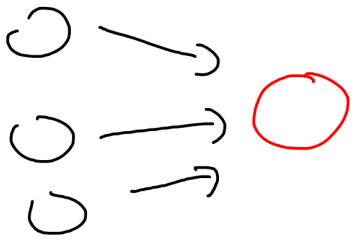
(To find out which one, do another PR step)

HITS = Hypertext Induced Topic Search

A different approach of scoring, where the score depends on the query.

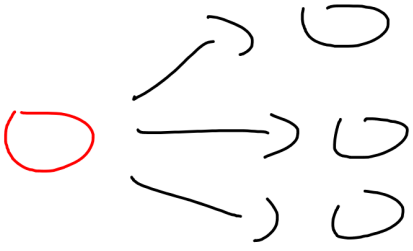
Actually the HITS score is made of two values:

- Authority score $a(i)$ = this value is high if page i has a good authority for a certain topic, which means that many good hubs for that topic point to him



$$a(u) = \sum_{v \to u} h(v)$$

- Hub score $h(i)$ = this value is high if page i is a good hub for a topic, which means that it points to many authoritative pages for that topic



$$h(u) = \sum_{u \to v} a(v)$$

Formally

$$\left.\begin{aligned} a &= A^T h \\ h &= A a \end{aligned}\right\} \Longrightarrow \begin{aligned} a &= A^T A a \\ h &= A A^T h \end{aligned}$$

where A is the adjacency matrix

→ h is an eigenvector of AA^t

→ a is an eigenvector of A^tA

Note

We could add some weights to the scores to valorize more some links

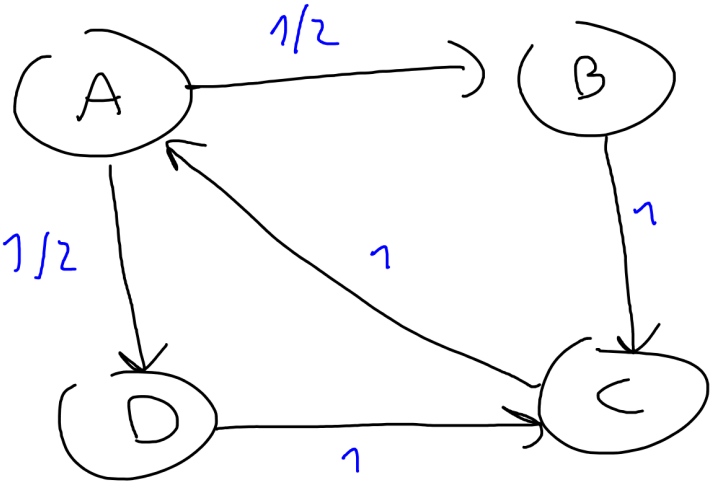
$$h(u) = \sum_{u \to v} w(u,v)\, a(v)$$

$$a(u) = \sum_{v \to u} w(u,v)\, h(v)$$

THIS RANKING METHOD IS TOO COSTLY AT QUERY TIME

Exercise

Given the graph



And the following initial scores ($\langle h, a \rangle$)

$A \langle ?, ? \rangle$, $B \langle \frac{1}{4}, \frac{1}{2} \rangle$, $C \langle \frac{1}{4}, 0 \rangle$, $D \langle \frac{1}{4}, 0 \rangle$

Do the computation for node A

$$h(A) = \frac{1}{2} + 0 = \frac{1}{2} \qquad a(B) \quad a(D)$$

$$a(A) = \frac{1}{4} \qquad h(C)$$