INFORMATION RETRIEVAL

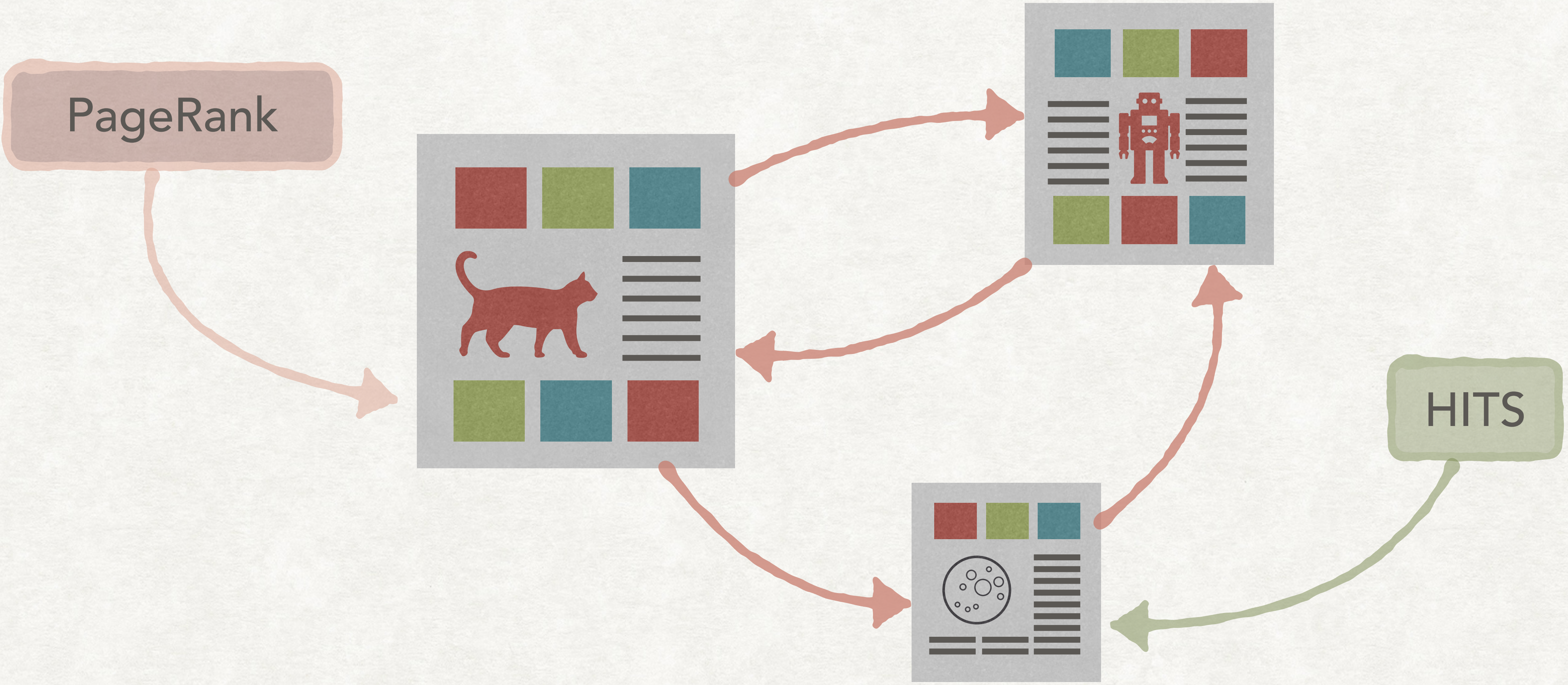
Luca Manzoni

lmanzoni@units.it

Lecture 10

LECTURE OUTLINE

*REQUIRES AT LEAST A 486



PAGERANK

HISTORY

BRIN, PAGE & GOOGLE

- PageRank is (part of) the algorithm used by Google in ranking the pages in its results.
- Developed in 1996 with the first paper on it published in 1998[2]: *"we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext "*
- Or, in other words,*"we take advantage of the link structure of the Web to produce a global "importance" ranking of every web page."*[1]
- The origin of PageRank can be traced back to methods in bibliometrics, sociometry, and possibly other fields.

[1] Page, L., Brin, S., Motwani, R. and Winograd, T., *The PageRank Citation Ranking: Bringing Order to the Web,* Stanford InfoLab, 1999

[2] Brin, S. and Page, L., *The anatomy of a large-scale hypertextual Web search engine,* Computer networks and ISDN systems, 30(1-7), pp.107-117, 1998

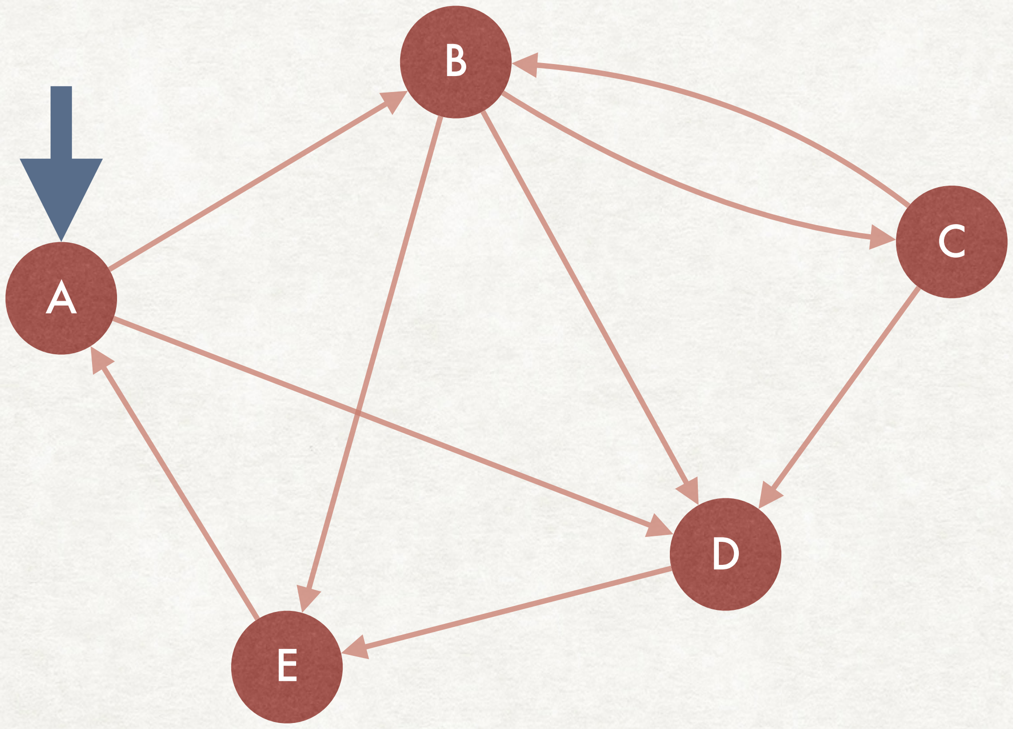
MAIN IDEA

USING LINKS TO GET SCORES

- We want to assign a value to each page that is independent from the query, i.e., a static score.
- We model a user randomly following links across web pages.
- What is the limit distribution of “where the user is” across all the pages?
- A user is without any memory of the page from where he/she came…
- …it seems like a case for using a Markov chain!

A SIMPLE EXAMPLE

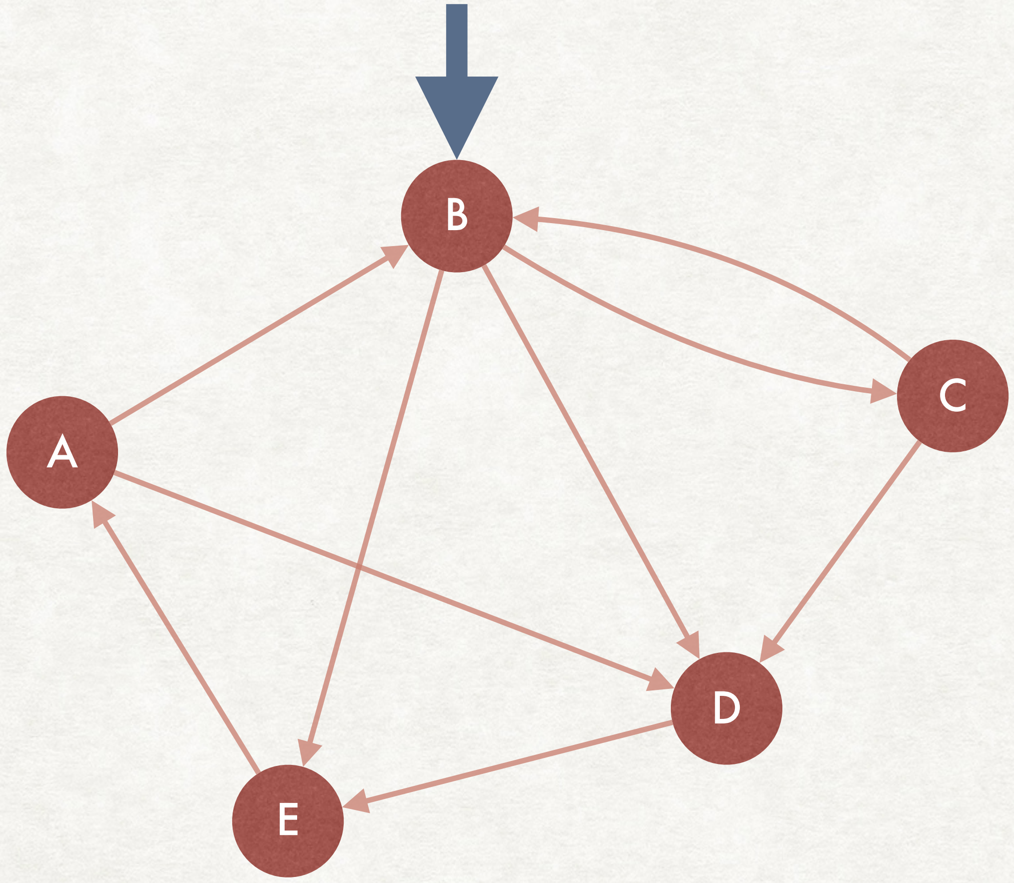
RANDOM WALK ON A GRAPH



We are visiting page A, where can move either to page B or D.
We select where to move uniformly at random.

A SIMPLE EXAMPLE

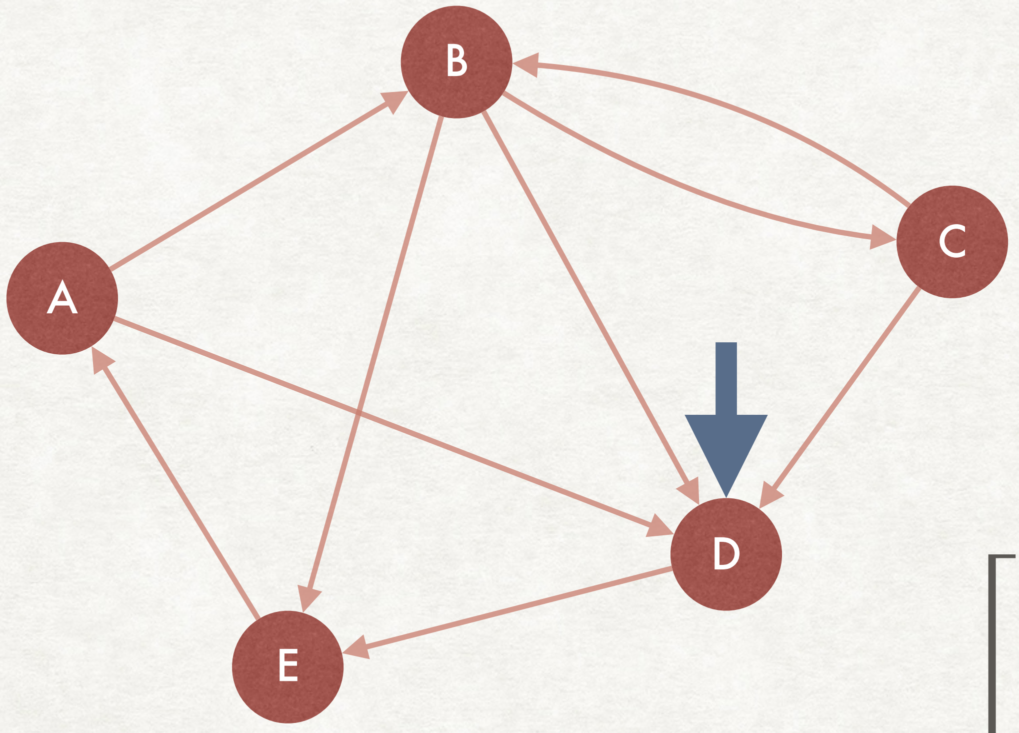
RANDOM WALK ON A GRAPH



We are visiting page B, where can move either to page C, D, or E.
We select where to move uniformly at random.

A SIMPLE EXAMPLE

RANDOM WALK ON A GRAPH



Probability of moving from node A to B

$$\begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We can formalise this random walk by defining a stochastic matrix

FORMALISATION AS A MARKOV CHAIN

AND THE STATIONARY DISTRIBUTION

Finding the probability distribution of the web page out idealised user is in then time tends to infinity

Is equivalent to

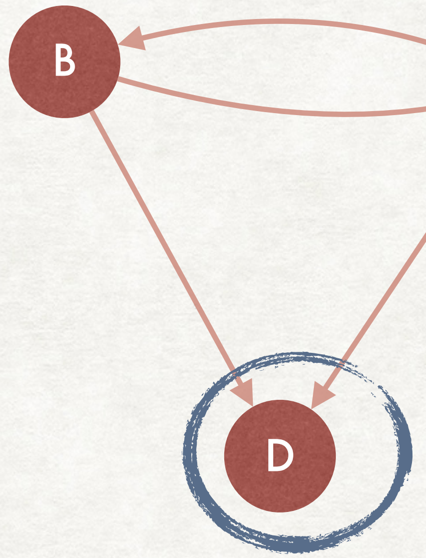
Finding the stationary distribution of the Markov chain with the following transition matrix:

$$R = \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

DANGLING NODES

NODES WITHOUT OUTGOING EDGES

A first problem that can appear in defining the stochastic matrix R is the presence of "dangling nodes"

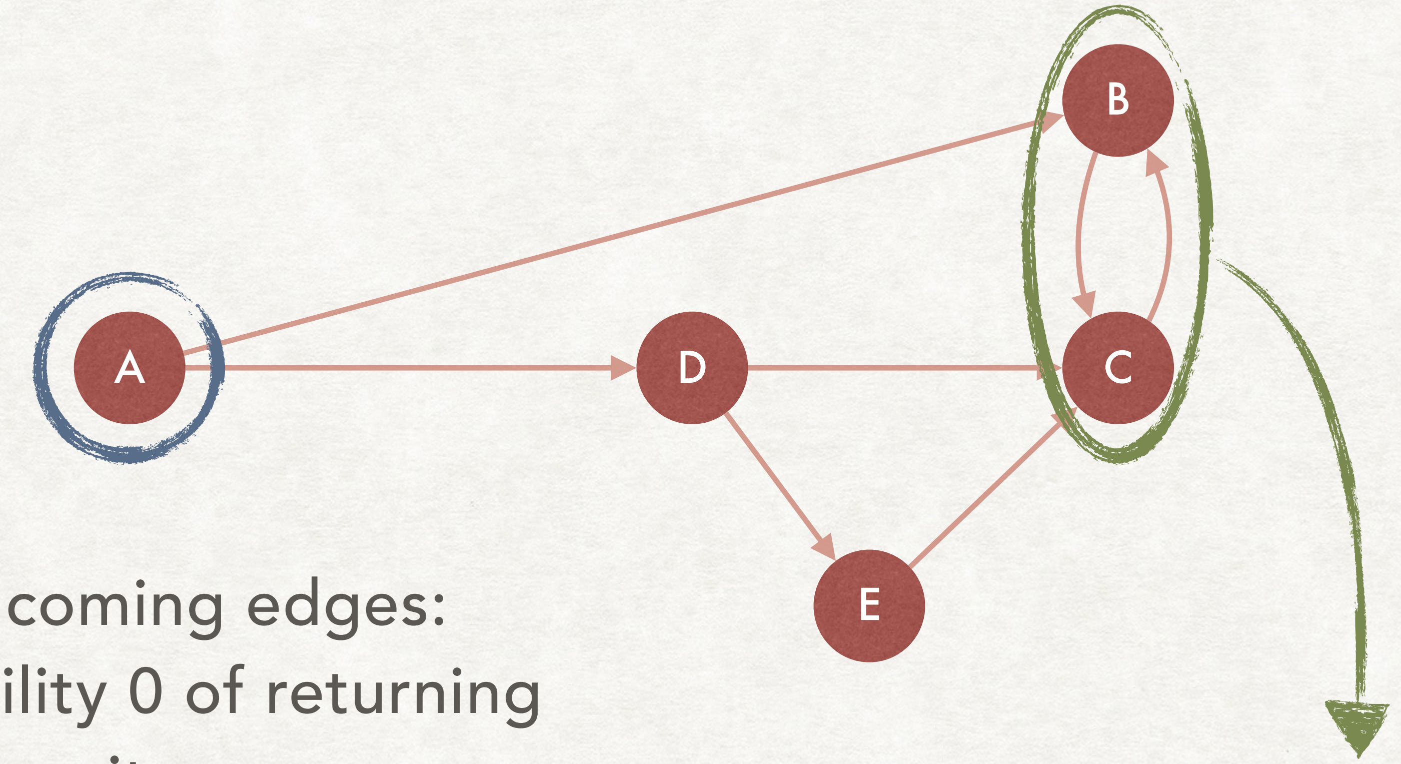


Node without outgoing edges: how to assign probabilities to go to another page?

A simple fix is to suppose that the user will go somewhere else uniformly at random: $\begin{bmatrix} \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \end{bmatrix}$

PROBLEMS

PAGES WITHOUT INCOMING OR OUTGOING LINKS



Node without incoming edges: we have probability 0 of returning to it once we leave it

The same problem is also present for nodes D and E

Group of nodes without outgoing edges: we can never leave them once entered

TELEPORTING

HOW TO MAKE THE USER SMARTER

- It is common to have “sinks” where it is impossible to exit by only following the links…
- …or pages that we cannot go back to.
- This produces an imbalance in our scores, that can potentially be exploited.
- In fact our idealised user can be a little bit smarter. At every page it can:
  - Move following one of the links in the page…
  - …or go to a random page

TELEPORTING

AND THE TRANSITION MATRIX

- Move to a linked page with probability $1-\alpha$
- Move to random page with probability α
- $\alpha > 0$ can be considered a "damping factor" or "probability that our user decides to go to another website"

$$P = \alpha \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix} + (1-\alpha) \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

TELEPORTING

AND THE TRANSITION MATRIX

- We assign a probability of $\frac{1}{N}$ of landing on any particular page.

- The previous matrix can also be written as:

$$P = \alpha \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \underbrace{\begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix}}_{\text{“Jump vector”}} + (1-\alpha) \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

TELEPORTING

AND THE TRANSITION MATRIX

$$P = \alpha \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{bmatrix} + (1 - \alpha) \begin{bmatrix} 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We usually write it as $P = \alpha \overrightarrow{1}^T \overrightarrow{J} + (1 - \alpha)R$

Can we find something about P that helps us in computing the PageRank of all pages (i.e., the stationary distribution)?
Can we have a solution that is independent from any initial guess that we might have to perform?

TELEPORTING

AND THE STATIONARY DISTRIBUTION

- With this “teleporting” trick we can now go to any other web page in one step.

- Which means that all entries of P are positive.

- Which means that we can apply the Perron-Frobenius theorem (actually one reformulation of it):

If P is a positive row (or column) stochastic matrix then:

1. The eigenvalue 1 is the largest eigenvalue and has multiplicity 1

2. There is a unique stochastic eigenvector for the eigenvalue 1

COMPUTING PAGERANK EXACTLY

USING THE PERRON-FROBENIUS THEOREM

The PageRank vector of the transition matrix P is the **unique** stochastic eigenvector corresponding to the eigenvalue 1

$$\overrightarrow{\pi} P = \lambda \overrightarrow{\pi}$$

In out case $\lambda = 1$, thus:

$$\overrightarrow{\pi} P = \overrightarrow{\pi}$$

Which is a linear system, we know how to solve it…

…except that P is a square matrix with a few billions of rows.

COMPUTING PAGERANK ITERATIVELY

A PRACTICAL APPROACH

- Usually we do not solve exactly the PageRank for a set of web pages.
- We use an iterative methods that, in fact, converges quite rapidly.
- The main idea is that, if we start from a stochastic vector $\overrightarrow{x}$, maybe giving equal probability to each page…
- …then $\overrightarrow{x}P^t$ for a large enough t would be a good approximation of the exact solution $\overrightarrow{\pi}$.

See also: Pavel Berkhin, A Survey on PageRank Computing, Internet Mathematics Vol. 2, No. 1: 73-120, 2005

COMPUTING PAGERANK ITERATIVELY

A PRACTICAL APPROACH

In pseudocode this could be expressed as:

```
$\vec{x}_0 = \text{random}()$
do
    $\vec{x}_t = \vec{x}_{t-1}\left(\alpha \vec{1}^T \vec{J} + (1-\alpha)R\right)$
while $|\vec{x}_t - \vec{x}_{t-1}|_1 \ge \varepsilon$
```

Start with a random probability distribution

Update the vector by multiplying it by P

Until the difference between the vectors in two consecutive iterations is below $\varepsilon > 0$

TOPIC-SPECIFIC PAGERANK

USING PAGERANK FOR SPECIFIC TOPICS

- In addition to computing PageRank scores for all pages we can limit the computation to single topics.
- How?
- Simply change the probability distribution for the “teleportation”, i.e., the “jump vector”.
- Start with a (non-empty) set S of pages specific to a certain topic.
- Your jumps can only be inside S.

TOPIC-SPECIFIC PAGERANK

USING PAGERANK FOR SPECIFIC TOPICS

Given a set of pages S, we consider a topic-specific jump vector $\overrightarrow{J_S}$ in the equation:

$$P = \alpha \overrightarrow{1}^T \overrightarrow{J_S} + (1 - \alpha) R$$

With the elements of $\overrightarrow{J_S}$ now defined as:

$$\overrightarrow{J_{S_i}} = \begin{cases} \frac{1}{|S|} & \text{if } i \in S \\ 0 & \text{otherwise} \end{cases}$$

We will find a set $Y \supseteq S$ of pages with positive PageRank, thus obtaining the solution $\overrightarrow{\pi_S}$ of “topic specific PageRank for S”

PERSONALISED PAGERANK

FOR DIFFERENT USERS

- We might want to add a special PageRank score for every user, depending on the topics he/she is interested in.
- For example, based on a set of favorite web pages.
- However, performing the PageRank computation for every user is too expensive.
- We can use the *linearity of PageRank*.

PERSONALISED PAGERANK

AND LINEARITY OF PAGERANK

Let S_1 and S_2 be two disjoints of “topic specific” pages.

Suppose that the corresponding PageRank scores are $\overrightarrow{\pi_1}$ and $\overrightarrow{\pi_2}$.

For a user that is interested in the first topic with weight $w_1 \geq 0$ and in the second topic with weight $w_2 \geq 0$, with $w_1 + w_2 = 1$ we can compute the corresponding PageRank scores as

$$w_1 \overrightarrow{\pi_1} + w_2 \overrightarrow{\pi_2}$$

Hence we can compute personalised PageRank scores with a weighted sum of pre-computed scores.

MANIPULATION OF PAGERANK

AND REL=NOFOLLOW

- There is an implicating conflict between the indexing (especially the one performed by Google) and the people managing the websites.
- Google needs to keep the search results relevant to the user.
- Normal and spam websites wants to rank high in the search results.
- To mitigate some of the problems, the “rel=nofollow” attribute was added to HTML.
- A link like:

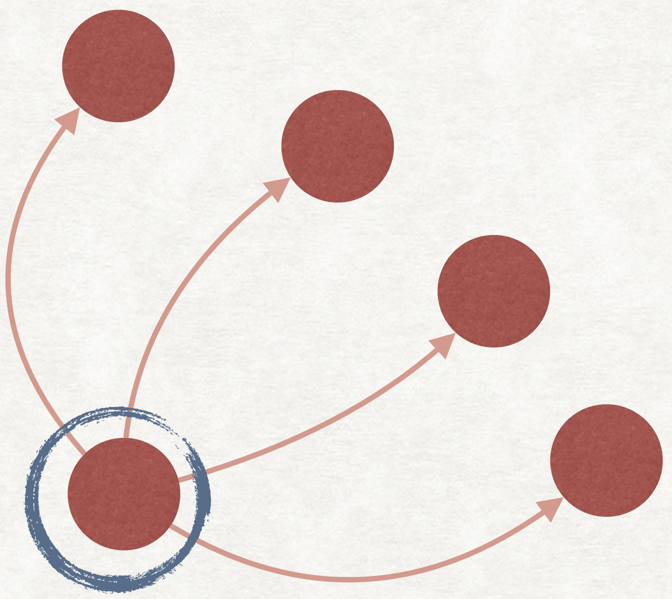
  ```
  <a href="http://www.example.com/" rel="nofollow">Link</a>
  ```

  would not be considered for the purpose of computing the PageRank score.

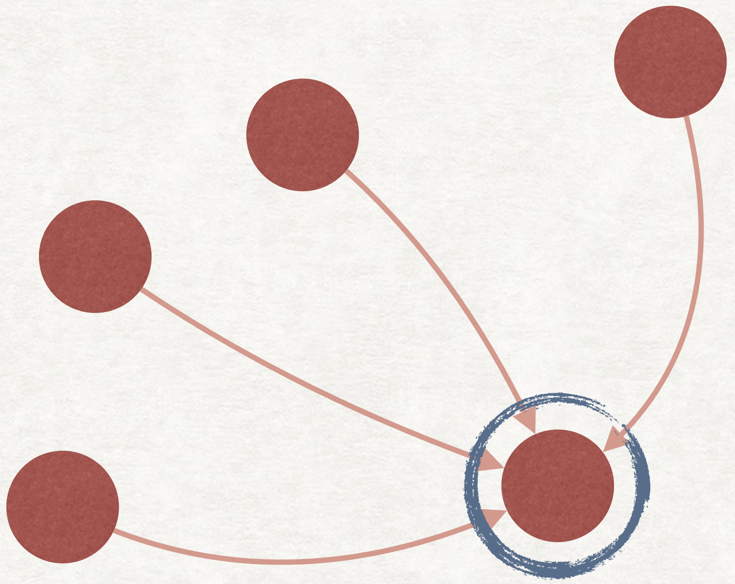
HITS

HUBS AND AUTHORITIES

TWO TYPES OF SCORES



This page links a lot of other pages. It can be considered an **hub**.



This page is linked by a lot of other pages. It can be considered an **authority**.

HUBS AND AUTHORITIES

TWO TYPES OF SCORES

- Hubs are pages that are important not for their content, but for the links that they provide toward pages with interesting content.

- Authorities are pages that are important for their content; therefore, they are linked by many pages.

- The **hyperlink-induced topic search** (HITS) algorithm assigns two different scores to each page, an authority and a hub score.

- The main idea behind the algorithm is:

    - A good hub points to pages with high authority score.

    - A good authority is pointed by pages with high hub scores.

HUBS AND AUTHORITIES

TWO TYPES OF SCORES

- Differently from PageRank, HITS is usually computed when the query is executed:

- A set of pages is obtained by some other methods (e.g., by looking at the text content of the page).

- We consider the subset of pages that we have retrieved (which will probably have very few links to each other) as a **root set**.

- We add to the root set all pages pointed and pointing to it.

- In this extended set we compute the two scores, that can now be used for ranking.

HOW TO COMPUTE SCORES

HUBS AND AUTHORITIES

The hub score $h(x)$ of a page x is defined as:

$$h(x) = \sum_{x \to y} a(y)$$

The sum of the authority scores for all pages *linked* by x.

The authority score $a(x)$ of a page x is defined as:

$$a(x) = \sum_{y \to x} h(y)$$

The sum of the hub scores for all pages *that links to* x.

HOW TO COMPUTE SCORES

HUBS AND AUTHORITIES

As with PageRank, we can compute the scores analytically.
But here we illustrate an iterative method

Start with all hub and authority scores set to 1.
At each time step $t > 0$ update them as:

$$\bar{h}_t(x) = \sum_{x \to y} a_{t-1}(y) \qquad \bar{a}_t(x) = \sum_{y \to x} h_{t-1}(y)$$

But if we only perform this update we might not converge!
We need to normalise the scores:

$$h_t(x) = \frac{\bar{h}_t(x)}{\sqrt{\sum_y (\bar{h}_{t-1}(y))^2}} \qquad a_t(x) = \frac{\bar{a}_t(x)}{\sqrt{\sum_y (\bar{a}_{t-1}(y))^2}}$$