Link Analysis

Web Ranking

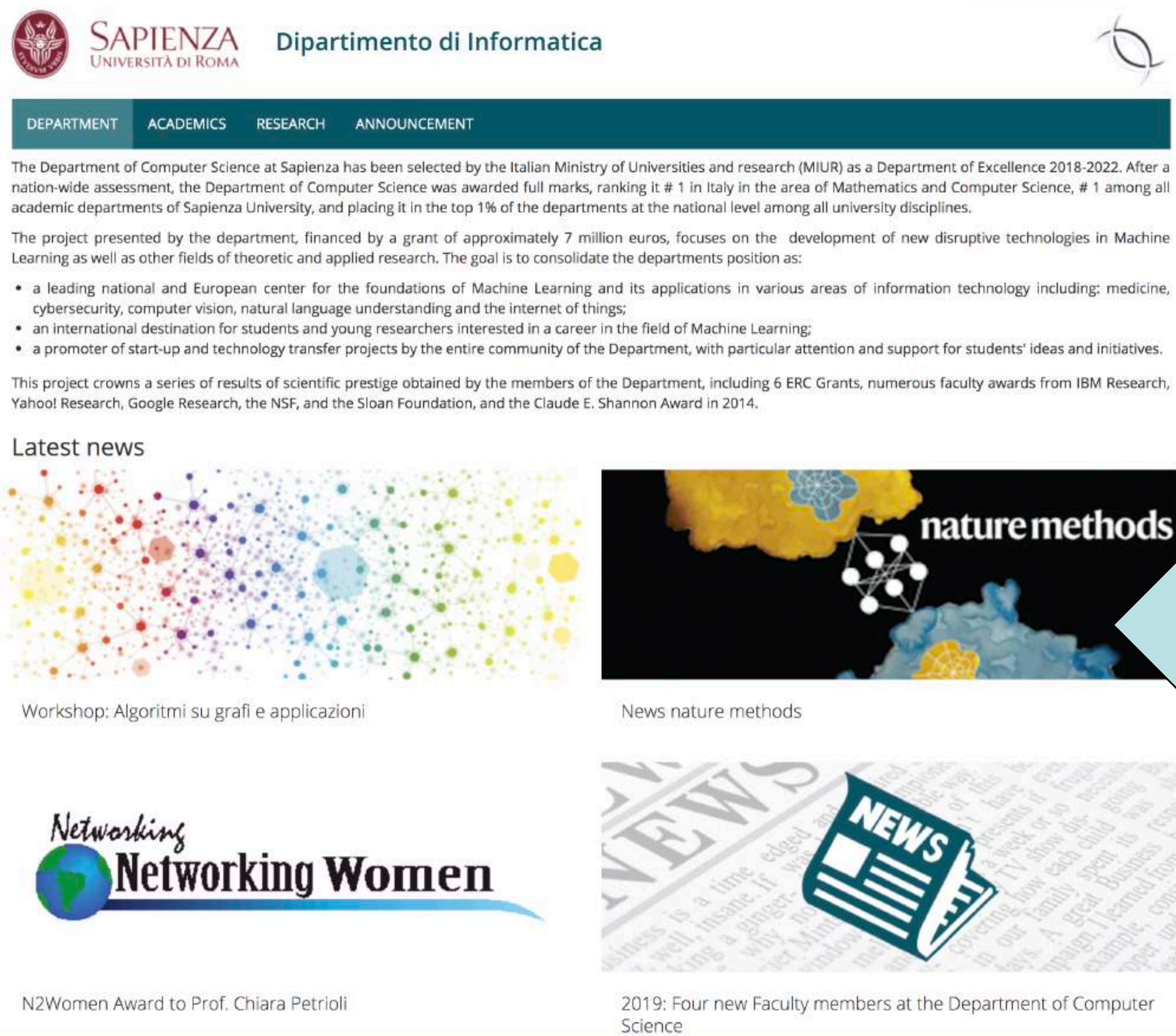
- Documents on the web are first ranked according to their relevance vrs the query
- Additional ranking methods are needed to cope with huge amount of information
- Additional ranking methods:
  - Classification (manual, automatic)
  - **Link Analysis** (today's lesson)

Why link analysis?

- The web is **not** just a collection of documents – its hyperlinks are important!
- A link from page *A* to page *B* may indicate:
  - *A* is related to *B*, or
  - *A* is recommending, citing, voting for or endorsing *B*
- Links are either
  - referential – *click here and get back **home***, or
  - Informational – *click here to get more **detail***
- Links affect the ranking of web pages and thus have commercial value.
- The idea of using links is somehow “borrowed” by citation analysis

A is related to *B, B is referential*

A is citing B, B is informational



SAPIENZA UNIVERSITÀ DI ROMA

Dipartimento di Informatica

DEPARTMENT ACADEMICS RESEARCH ANNOUNCEMENT

The Department of Computer Science at Sapienza has been selected by the Italian Ministry of Universities and research (MIUR) as a Department of Excellence 2018-2022. After a nation-wide assessment, the Department of Computer Science was awarded full marks, ranking it # 1 in Italy in the area of Mathematics and Computer Science, # 1 among all academic departments of Sapienza University, and placing it in the top 1% of the departments at the national level among all university disciplines.

The project presented by the department, financed by a grant of approximately 7 million euros, focuses on the development of new disruptive technologies in Machine Learning as well as other fields of theoretic and applied research. The goal is to consolidate the departments position as:

- a leading national and European center for the foundations of Machine Learning and its applications in various areas of information technology including: medicine, cybersecurity, computer vision, natural language understanding and the internet of things;
- an international destination for students and young researchers interested in a career in the field of Machine Learning;
- a promoter of start-up and technology transfer projects by the entire community of the Department, with particular attention and support for students' ideas and initiatives.

This project crowns a series of results of scientific prestige obtained by the members of the Department, including 6 ERC Grants, numerous faculty awards from IBM Research, Yahoo! Research, Google Research, the NSF, and the Sloan Foundation, and the Claude E. Shannon Award in 2014.

Latest news

Workshop: Algoritmi su grafi e applicazioni

nature methods

News nature methods

Networking Networking Women

N2Women Award to Prof. Chiara Petrioli

NEWS

2019: Four new Faculty members at the Department of Computer Science

A is recommending B, B is informational

SAPIENZA UNIVERSITÀ DI ROMA

Dipartimento di Informatica

DEPARTMENT ACADEMICS RESEARCH ANNOUNCEMENT

The Department of Computer Science at Sapienza has been selected by the Italian Ministry of Universities and research (MIUR) as a Department of Excellence 2018-2022. After a nation-wide assessment, the Department of Computer Science was awarded full marks, ranking it # 1 in Italy in the area of Mathematics and Computer Science, # 1 among all academic departments of Sapienza University, and placing it in the top 1% of the departments at the national level among all university disciplines.

The project presented by the department, financed by a grant of approximately 7 million euros, focuses on the development of new disruptive technologies in Machine Learning as well as other fields of theoretic and applied research. The goal is to consolidate the departments position as:

- a leading national and European center for the foundations of Machine Learning and its applications in various areas of information technology including: medicine, cybersecurity, computer vision, natural language understanding and the internet of things;
- an international destination for students and young researchers interested in a career in the field of Machine Learning;
- a promoter of start-up and technology transfer projects by the entire community of the Department, with particular attention and support for students' ideas and initiatives.

This project crowns a series of results of scientific prestige obtained by the members of the Department, including 6 ERC Grants, numerous faculty awards from IBM Research, Yahoo! Research, Google Research, the NSF, and the Sloan Foundation, and the Claude E. Shannon Award in 2014.

Latest news

Workshop: Algoritmi su grafi e applicazioni

News nature methods

N2Women Award to Prof. Chiara Petrioli

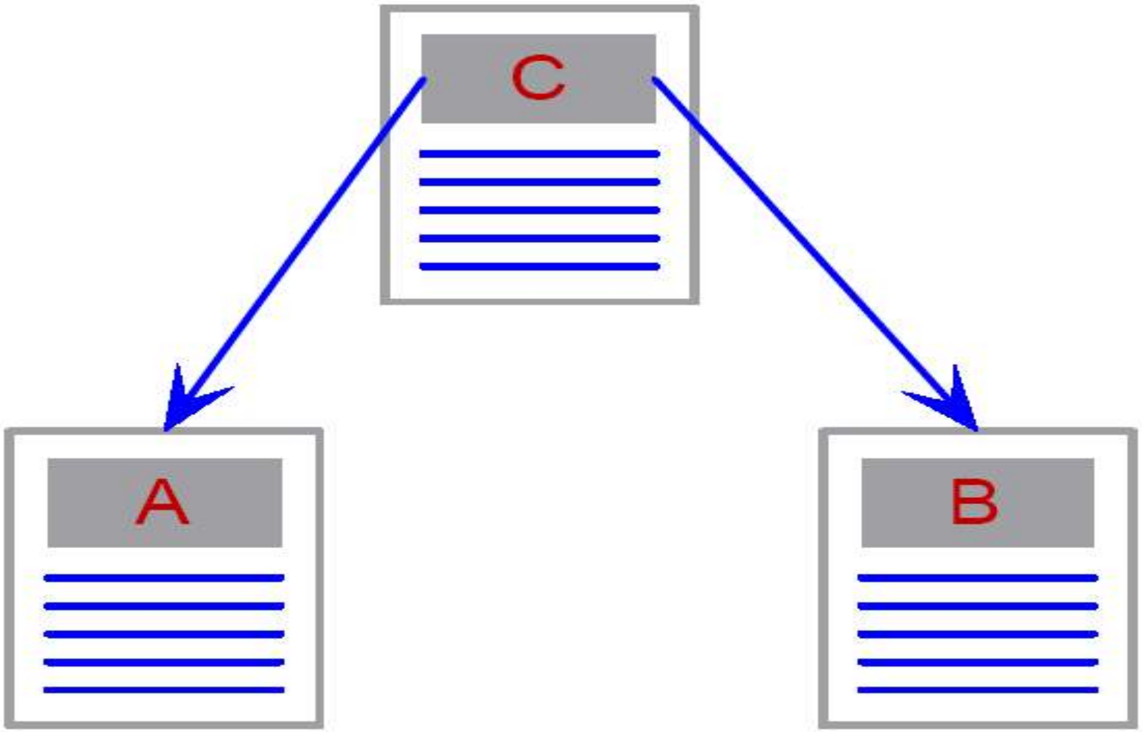
2019: Four new Faculty members at the Department of Computer Science

Citation Analysis

- The **impact factor** of a journal = *A/B*
  - *A* is the number of current year citations to articles appearing in the journal during previous two years.
  - *B* is the number of articles published in the journal during previous two years.

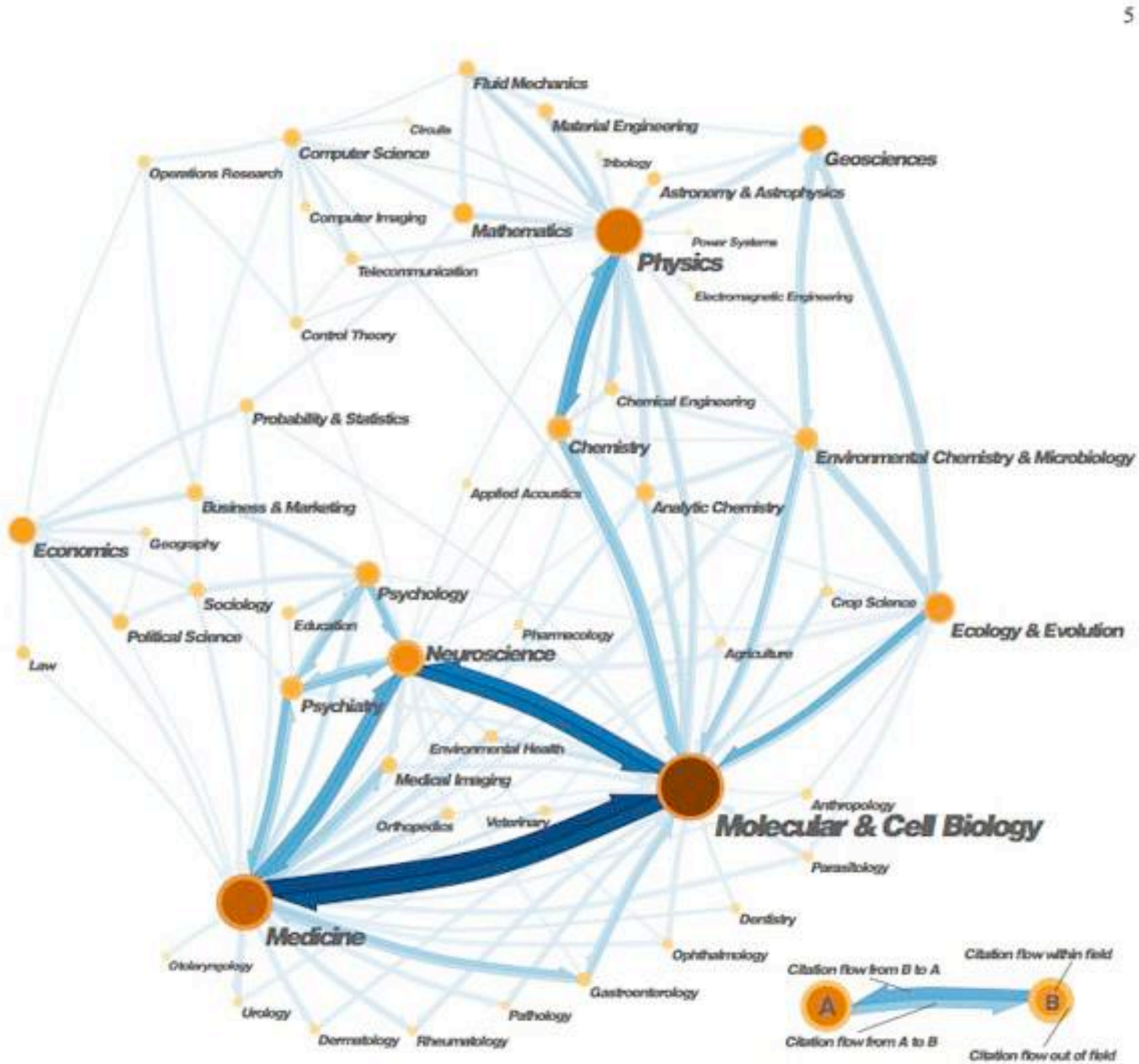
Journal Title (AI)	**Impact Factor (2004)**
J. Mach. Learn. Res.	5.952
IEEE T. Pattern Anal.	4.352
IEEE T. Evolut. Comp.	3.688
Artif. Intell.	3.570
Mach. Learn.	3.258

Co-Citation



- *A* and *B* are co-cited by *C*, implying that they are related or associated.
- The strength of co-citation between *A* and *B* is the number of times they are co-cited.

Clusters from Co-Citation Graph



Citations vs. Links

- Web links are a bit different than citations:
  - Many links are navigational.
  - Many pages with high out-degree are portals, rather than content providers.
  - Not all links are endorsements (e.g. pointers to "fake" conferences).
  - Company websites don't point to their competitors.

However, the general idea that

"many citations = authority"

has been borrowed in link analysis

HITS and Page Rank: algebra that you need

- Eigenvector, eigenvalue and eigen-decomposition of normal matrixes ✔
- **Iterative methods**

Iterative methods

- A mathematical procedure that generates a sequence of **improving approximate solutions** for a class of problems
- General formulation: $\boldsymbol{x}^{t+1}=A\boldsymbol{x}^{t}$ *where* $\boldsymbol{x}$ *is a vector and* t *an iteration*
- Iterative methods converge under specific hypotheses for matrix A.
- Condition a: A is square, real and symmetric
  - A **real symmetric** matrix is also **normal** and it exists a decomposition $U\Delta U^{-1}$ such that $\Delta=\text{diag}(\lambda_1,..\lambda_n)$ and $\lambda_1>..>\lambda_n$
  - Under these conditions, the method converges (as shown later)

Iterative methods (2)

- Condition b: A is <u>square, stochastic and irreducible</u>
  - In a **stochastic** matrix, either $\Sigma_i(a_{ij})=1$ (right stochastic) or $\Sigma_j(a_{ij})=1$ (left stochastics). It means that columns or rows are probabilities.
  - A *nxn* matrix is **reducible** if indices *i,j* = 1,2..n can be divided into two **disjoint nonempty** sets $i_1,i_2..i_\mu$ and $j_1,j_2..j_v$ such that $a_{i_\alpha j_\beta}=0$ for $\alpha=1,2..\mu$ and $\beta=1,2..v$ (equivalent to say that the subsumed graph **has disconnected components**)
- NOTE: Conditions a and b are not the ONLY conditions for convergence of iterative methods, **but those we need here**

Stochastic matrix (left stochastic)

$$P = \begin{bmatrix} 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 1 & 0 & 0 \\ 1/4 & 1/4 & 0 & 1/4 & 1/4 \\ 0 & 0 & 1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Reducible matrix

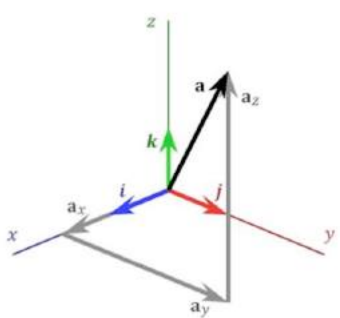
$$A' = x^{-1}Ax = x^{-1}\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n1} & \dots & a_{nn} \end{pmatrix} x = \begin{pmatrix} \begin{matrix} a_{11}' & a_{12}' & \cdots & a_{1m}' \\ a_{21}' & a_{22}' & \cdots & a_{2m}' \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}' & a_{m1}' & \dots & a_{mm}' \end{matrix} & 0 & \dots & 0 \\ \begin{matrix} 0 \\ \\ \vdots \end{matrix} & \begin{matrix} a_{11}'' & a_{12}'' & \cdots & a_{1l}'' \\ a_{21}'' & a_{22}'' & \cdots & a_{2l}'' \\ \vdots & \vdots & \vdots & \vdots \\ a_{l1}'' & a_{l1}'' & \dots & a_{ll}'' \end{matrix} & & \begin{matrix} \\ \vdots \\ 0 \end{matrix} \\ 0 & 0 & \dots & \begin{matrix} a_{11}''' & a_{12}''' & \cdots & a_{1k}''' \\ a_{21}'' & a_{22}'' & \cdots & a_{2k}''' \\ \vdots & \vdots & \vdots & \vdots \\ a_{k1}''' & a_{k1}''' & \dots & a_{lk}''' \end{matrix} \end{pmatrix} = \begin{pmatrix} A' & 0 & 0 \\ 0 & A'' & 0 \\ 0 & 0 & A''' \end{pmatrix}$$

A reducible matrix can be converted into a ***block diagonal form***

Geometric or graph interpretation of matrixes

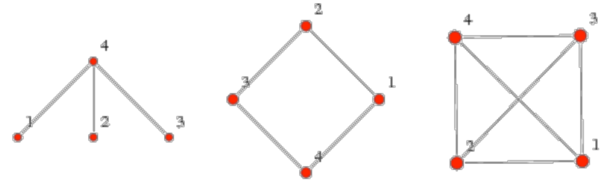
Geometric interpretation

- a_{ij} are coordinates of column vectors of the matrix on the carthesian axes i=1..n
- A**x** is a linear transformation: if A is **normal**, $\lambda_1 > .. > \lambda_n$ and there exist an othonormal space defined by A's eigenvectors on which **x** is projected.



Graph representation

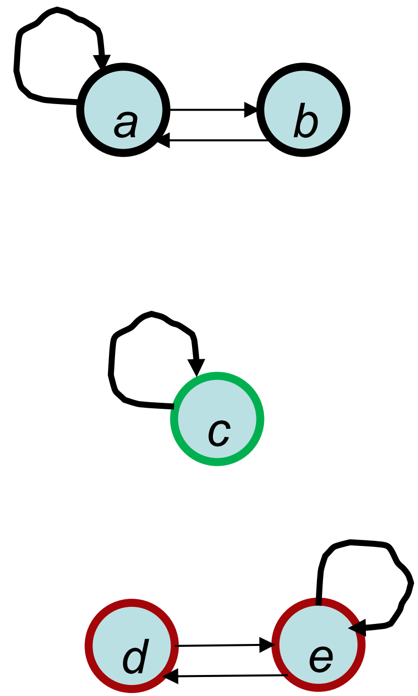
- A matrix is a **weighted graph**, a_{ij} represent the weight of an edge between nodes i and j
- Irreducible matrix= the subsumed graph is **connected**



$$\begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

Example stochastic reducible

$$
\begin{array}{c}
\begin{array}{cccccc} a & b & c & d & e & f \end{array} \\
\begin{bmatrix}
0.5 & 0.5 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.5 & 0.5 & \\
0 & 0 & 0 & 0 & 1 &
\end{bmatrix}
\end{array}
\quad
\begin{array}{c} a \\ b \\ c \\ d \\ e \\ f \end{array}
$$



Link Analysis

- **HITS (Hyperlink Induced Topic Serach) Jon *Kleinberg***
- Page Rank *Larry Page, Sergei Brin*

Hyperlink Induced Topic Search (HITS)

- Or Hypertext-Induced Topic Search(HITS) developed by Jon Kleinberg, while visiting IBM Almaden
- IBM expanded HITS into Clever, a web search engine for advertising (no longer an active project)
- However, HITS still used in many graph-based applications (e.g. social networks)

Main concept of the algorithm

- HITS stands for **Hypertext Induced Topic Search**.
- HITS is search-query dependent.
- When the user issues a search query,
  - HITS first expands the list of “relevant” (according to, e.g. vector space model) pages returned by a search engine
  - Next, it produces two rankings of the expanded set of pages, **authority ranking** and **hub ranking**.

Main concept of the algorithm-cont.

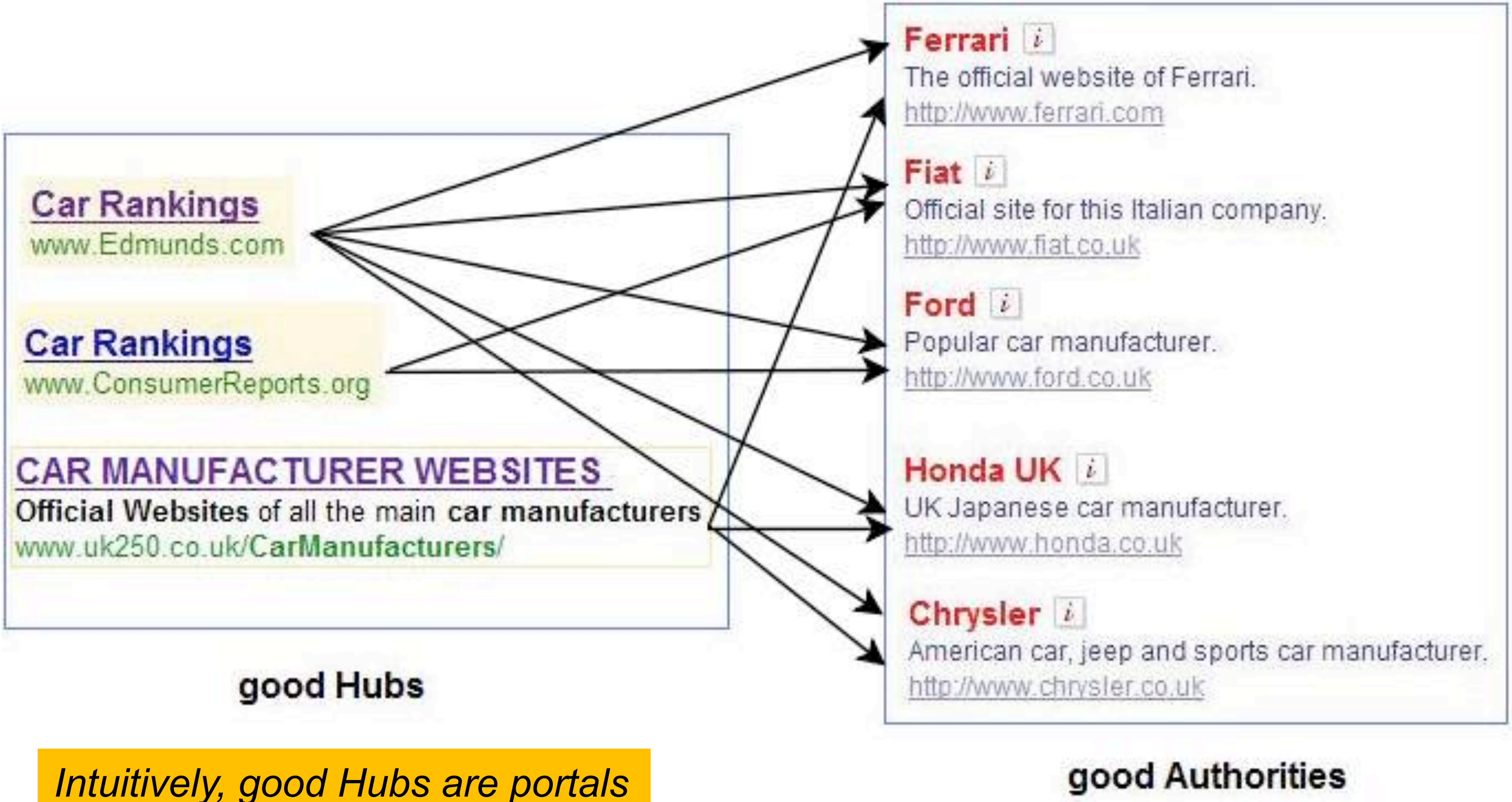
Authority: A authority is a page with many incoming links (in-links, back-links).

- The idea is that the page may have good or authoritative content on some topic and thus many people trust it and link to it.

Hub: A hub is a page with many out-links.

- The page serves as an organizer of the information on a particular topic and points to many good authority pages on the topic (e.g. a portal).

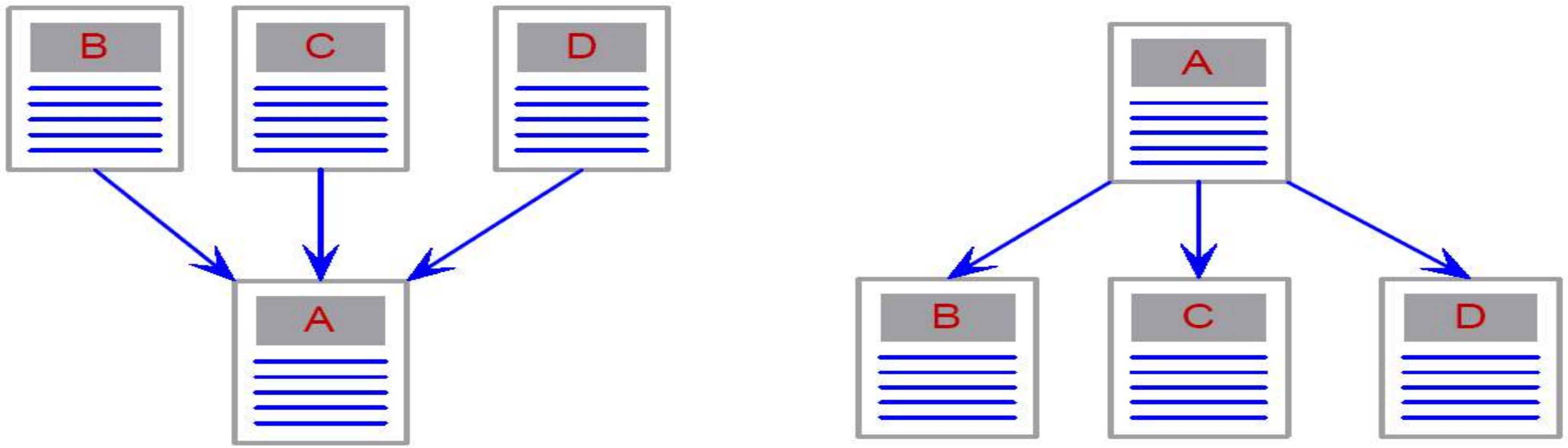
Example



Intuitively, good Hubs are portals

Query: Top automobile makers

HITS – Hubs and Authorities –



- **A** on the left is an **authority**
- **A** on the right is a **hub**

Description of HITS

- A good hub points to many good authorities, and
- A good authority is pointed to by many good hubs.
- Authorities and hubs have a **mutual reinforcement relationship**. The figure shows some densely linked authorities and hubs (a **bipartite sub-graph**).

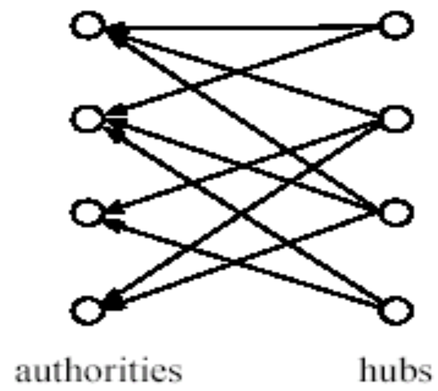


Fig. 8. A densely linked set of authorities and hubs

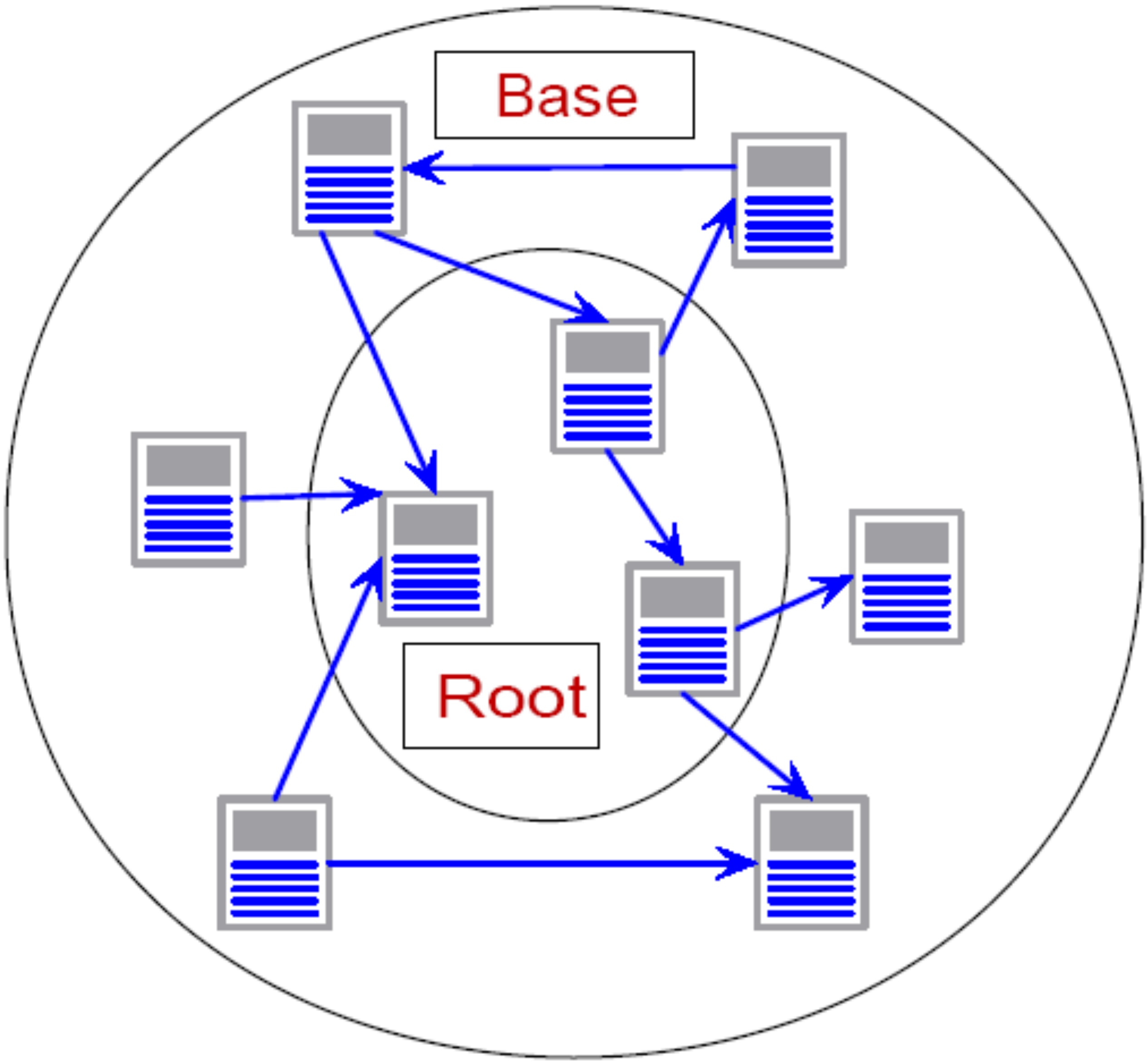
Hubs and Authorities: two steps

- First Step:
  - Constructing a **focused subgraph** of the WWW, <u>based on a user's query</u>
- Second step:
  - **Iteratively** calculate authority weight and hub weight for each page in the subgraph

The HITS algorithm: focused graph

- Given a broad search query, *q*, HITS first collects a set of pages as follows:
  - It sends the query *q* to a search engine.
  - It then collects *t* (*t* = 200 is used in the original HITS paper) highest ranked pages. This set is called the **root** set *W*.
  - It then grows *W* by including any page **pointed to** by a page in *W* and any page **that points** to a page in *W*. This gives a larger set *S*, **base set**.

Expanding the Root Set

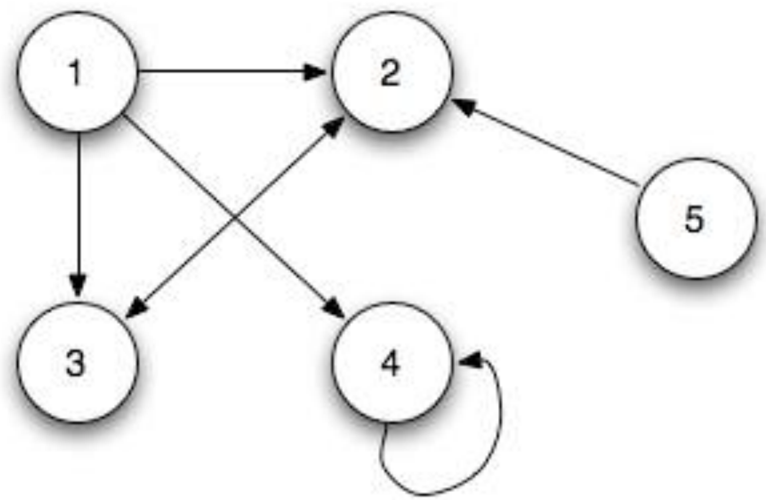


The link graph G

- HITS works on the pages in *S*= B *U* R, and assigns every page in *S* an **authority score** and a **hub score**.
- Let the number of web pages in *S* be *n*.
- We use *G* = (*V*, *E*) to denote the hyperlink graph of *S*. (V nodes, E edges). (NOTE: by construction, this is a CONNECTED graph)
- We use ***L*** to denote the adjacency matrix of the graph.

$$L_{ij} = \begin{cases} 1 & if\,(i,j) \in E \\ 0 & otherwise \end{cases} \qquad i \rightarrow j$$

Adjacency matrix (directed graph)



	1	2	3	4	5
1	0	1	1	1	0
2	0	0	1	0	0
3	0	1	0	0	0
4	0	0	0	1	0
5	0	1	0	0	0

What are the properties of this matrix (in general)? Irreducible? Symmetric? Square? Stocastic?

The HITS algorithm (cont'd)

- Let the authority score of the page *i* be *a*(*i*), and the hub score of page *i* be *h*(*i*).
- The **mutual reinforcing relationship** of the two scores is represented as follows:

$$a(i) = \sum_{(j,i)\in E} h(j)$$

$$h(i) = \sum_{(i,j)\in E} a(j)$$

Remember: E is the set of edges of the derived hyperlink graph

HITS formulation in matrix form

- We use ***a*** to denote the column **vector** with all the authority scores,

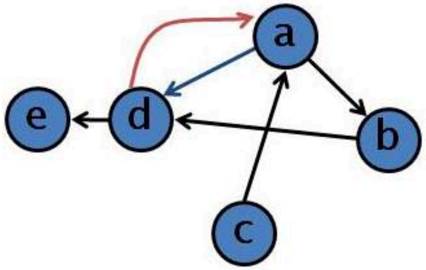
  a = (*a*(1), *a*(2), …, *a*(*n*))^{T}, and
- use ***h*** to denote the column **vector** with all the"hub scores",

  h = (*h*(1), *h*(2), …, *h*(*n*))^{T},
- L is the adjacency matrix and L^T its transpose
- Then, we can express previous formulas in matrix form as:

  $\mathbf{a} = L^T\mathbf{h}$ (I step)

  $\mathbf{h} = L\mathbf{a}$ (O step)

  normalize: **a**=**a**/||a|| **h**=**h**/||h||
- It is an **equivalent formulation** wrt $a(i) = \sum_{(j,i)\in E} h(j)$ since the sum in previous formula is **for all *j* linked to *i*** in E, and **L** has 1 where there is a link between i and j



	a	b	c	d	e
a	0	1	0	1	0
b	0	0	0	1	0
c	1	0	0	0	0
d	1	0	0	0	1
e	0	0	0	0	0

$$\begin{pmatrix} h_a \\ h_b \\ h_c \\ h_d \\ h_e \end{pmatrix} = \begin{pmatrix} 01010 \\ 00010 \\ 10000 \\ 10001 \\ 00000 \end{pmatrix} \times \begin{pmatrix} a_a \\ a_b \\ a_c \\ a_d \\ a_e \end{pmatrix} \quad \Longleftrightarrow \quad \begin{gathered} h_a = a_b + a_d \\ h_b = a_d \\ h_c = a_a \\ h_d = a_b + a_e \\ h_e = 0 \end{gathered}$$

***h*=*L*a**

Computation of HITS

- The computation of authority scores and hub scores uses power iteration iterative method.
- If we use $\boldsymbol{a}^t$ and $\boldsymbol{h}^t$ to denote authority and hub vectors at the t_{th} iteration, the iterations for generating the final (stationary) solutions are:

$$\boldsymbol{a}^t = \boldsymbol{L}^T\boldsymbol{h}^{t-1}$$
$$\boldsymbol{h}^{t-1} = \boldsymbol{L}\boldsymbol{a}^{t-1}$$

$$\longrightarrow$$

$$\boldsymbol{a}^t = \boldsymbol{L}^T\boldsymbol{L}\boldsymbol{a}^{t-1}$$
$$\boldsymbol{h}^t = \boldsymbol{L}\boldsymbol{L}^T\boldsymbol{h}^{t-1}$$

Example (simple algorithm)

- 2^{nd} Iteration
- I Step
- O Step

The HITS algorithm (with normalization)

Initialize

Iterate

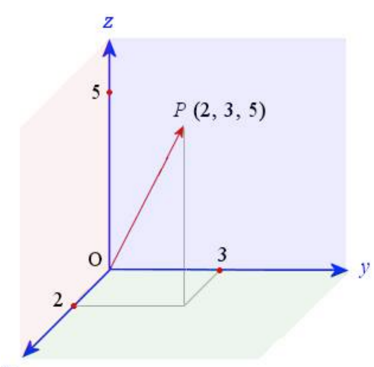
- $\underline{h}^{(0)} := (1, 1, \ldots$
- $t := 1$
- *Until convergence, do:*
  - $\underline{a}^{(t)} := L^T \underline{h}^{(t-1)} = L^T L \, \underline{a}^{(t-1)}$ (**update** **$\underline{a}$**)
  - $\underline{h}^{(t)} := L \, \underline{a}^{(t)} = L L^T \underline{h}^{(t-1)}$ (**update** **$\underline{h}$**)
  - $\underline{a}^{(t)} := \underline{a}^{(t)} / ||\underline{a}^{(t)}||$ *and* $\underline{h}^{(t)} := \underline{h}^{(t)} / ||\underline{h}^{(t)}||$ (***normalize***)

Does it converge to a stationary solution?

Digression: many vector notations, don't be confused!

- $\underline{X}$, $\vec{X}$ and $\mathbf{X}$ (underline, arrow and bold) are all valid *intensional* notations for vectors!!
- X: $(x_1, x_2 \ldots x_n)$ is an *extensional* notation (shows all coordinates of the vector) and is either column or row:

$$\begin{pmatrix} x_1 \\ x_2 \\ \square \\ \square \\ x_{n-1} \\ x_n \end{pmatrix}$$

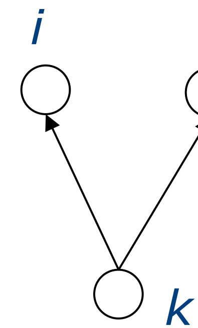


- *Finally, we have the graphic notation*

Back to HITS: meaning of the $L\,L^T$ and $L^T\,L$ matrixes

- *L is the* ***adjacency matrix*** *of the graph*
- *$L^T L$ is the* ***authority matrix****:*

$$A = L^T L \;=\; \begin{pmatrix} & & \\ & A_{ij} & \\ & & \end{pmatrix} = \underbrace{\begin{pmatrix} & & \\ & L^T_{ik} & \\ & & \end{pmatrix}}_{L^T} \underbrace{\begin{pmatrix} & & \\ & L_{kj} & \\ & & \end{pmatrix}}_{L}$$

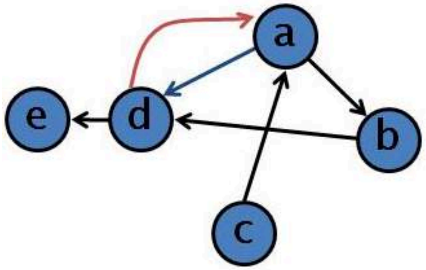


L_{kj} column "means" that node j is pointed by all non-zero nodes k;
L^T_{ik} "means" that node i is pointed by all non-zero nodes k

$$A_{ij} = \sum_{k=1}^{n} L^T_{ik} L_{kj} = \sum_{k=1}^{n} L_{ki} L_{kj}$$

A_{ij} is the number of **co-citations**, the number of nodes pointing to both i and j

..is this something you have already seen???????



	a	b	c	d	e
a	0	1	0	1	0
b	0	0	0	1	0
c	1	0	0	0	0
d	1	0	0	0	1
e	0	0	0	0	0

$$L^T \times L = \begin{pmatrix} 0&0&1&1&0 \\ 1&0&0&0&0 \\ 0&0&0&0&0 \\ 1&1&0&0&0 \\ 0&0&0&1&0 \end{pmatrix} \times \begin{pmatrix} 0&1&0&1&0 \\ 0&0&0&1&0 \\ 1&0&0&0&0 \\ 1&0&0&0&1 \\ 0&0&0&0&0 \end{pmatrix} = \begin{pmatrix} 2&0&0&0&\boxed{1} \\ 0&1&0&1&0 \\ 0&0&0&0&0 \\ 0&1&0&2&0 \\ 1&0&0&0&1 \end{pmatrix}$$

Aij is the number of nodes pointing both i and j.

*For example, A_{ae}=1 since only node **d** points to both **a** and **e***

Proof of convergence of HITS (power method)

- *Since matrix A (=LL^T) is* ***square and symmetric****, it has an eigen-decomposition $U\Lambda U^{-1}$ where $\lambda_1, \lambda_2, \ldots, \lambda_k$ are then* ***eigenvalues*** *and $|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_k|$ (note: they are eigenvalues of A and* ***singular values*** *of L!!)*
- *$\underline{x}_1, \ldots, \underline{x}_k$ are the* ***eigenvectors*** *of A and they form an orthonormal basis (**e.g.** :*

$\alpha_1\underline{x}_1 + \alpha_2\underline{x}_2 + \ldots + \alpha_k\underline{x}_k = 0$ ***iff*** *$\alpha_1 = \ldots = \alpha_k = 0$, sinc*

$$\underline{a}^{(k)} := L^T L\, \underline{a}^{(k-1)} = A\, \underline{a}^{(k-1)}$$

- *A generic vector $\underline{v}_0$ (in our case, either $\underline{\mathbf{h}}^{(0)}$ o $\underline{\mathbf{a}}^{(0)}$) can be re-written as:*
  - *$\underline{v}^0 = \alpha_1\underline{x}_1 + \alpha_2\underline{x}_2 + \ldots + \alpha_k\underline{x}_k$ (its projection on the orthonormal space of A)*

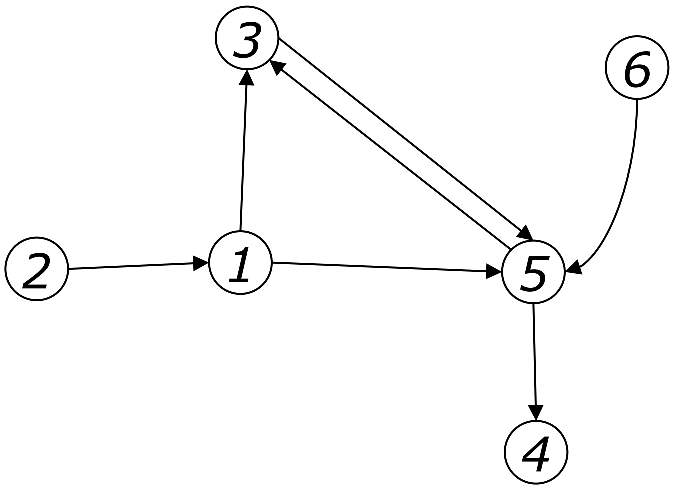
$$\forall i: \; \vec{x}_i, \lambda_i \; eigenvalue, eigenvector \; of \; A, \; A = \lambda_i \vec{x}_i$$

- *Hence (let*
  - *$\underline{\mathbf{h}}^1 = \mathbf{A}\underline{\mathbf{h}}^0 = \alpha_1 A\underline{x}_1 + \alpha_2 A\underline{x}_2 + \ldots + \alpha_k A\underline{x}_k = \boldsymbol{\alpha_1\lambda_1\underline{x}_1 + \alpha_2\lambda_2\underline{x}_2 + \ldots + \alpha_k\lambda_k\underline{x}_k} =$*
  - *$\lambda_1[\alpha_1\underline{x}_1 + \alpha_2(\lambda_2/\lambda_1)\underline{x}_2 + \ldots + \alpha_k(\lambda_k/\lambda_1)\underline{x}_k]$*
- *And in general:*
  - *$\underline{h}^m = A\underline{h}^{m-1} = A^{\mathbf{m}}\underline{h}^0 = \alpha_1 A^m\underline{x}_1 + \alpha_2 \mathbf{A^m}\underline{x}_2 + \ldots + \alpha_k A^m\underline{x}_k = \alpha_1\lambda_1^m\underline{x}_1 + \alpha_2\boldsymbol{\lambda_2^m}\underline{x}_2 + \ldots + \alpha_k\lambda_k^m\underline{x}_k = \; \boldsymbol{\lambda_1^m[\alpha_1\underline{x}_1 + \alpha_2(\lambda_2/\lambda_1)^m\underline{x}_2 + \ldots + \alpha_k(\lambda_k/\lambda_1)^m\underline{x}_k]}$*
- *Since $|\lambda_i/\lambda_1| < 1$, $i = 2, 3, \ldots, n$, we get:*

$$\lim \frac{1}{\ldots} \underline{h}_m = \lim \frac{1}{\ldots} A^m \underline{h}^0 = \alpha_1 \underline{x}_1$$

Speed of convergence depends on λ_2/λ_1 **and on initial choice of h^0**

HITS: Example (1)



$$L = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

$$L^T L = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

$$LL^T = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 2 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 2 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

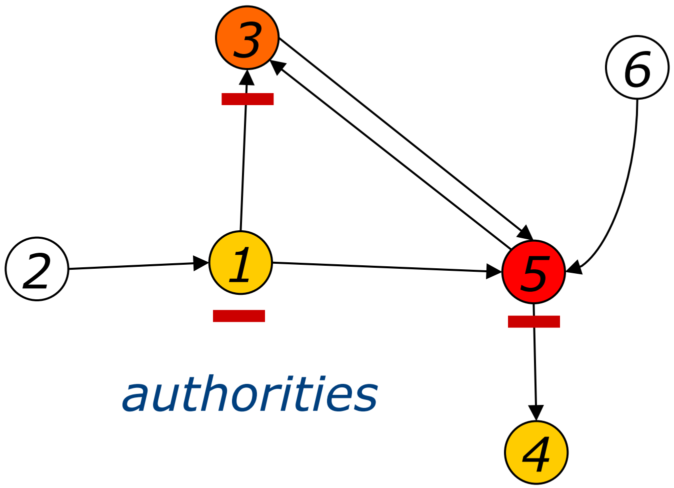
Ex: 3 and 4 are "co-cited" by 5

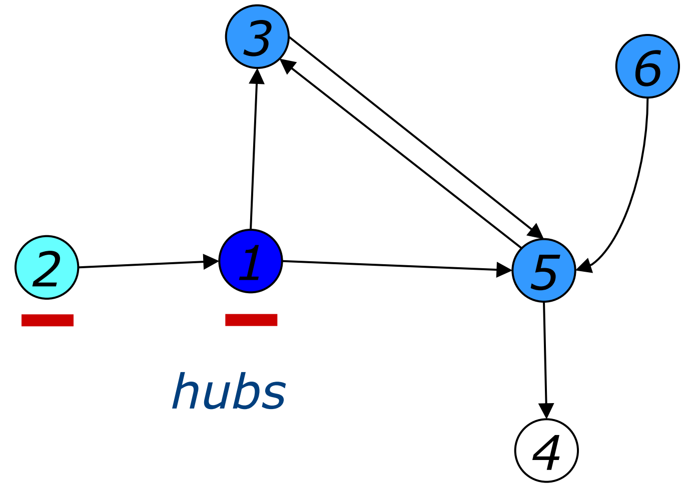
3 and 1 co-cite 5"

$-\underline{a}^{(1)} := L^T \underline{h}^{(0)}$

$-\underline{h}^{(1)} := L\, \underline{a}^{(1)}$

HITS: Example (2)





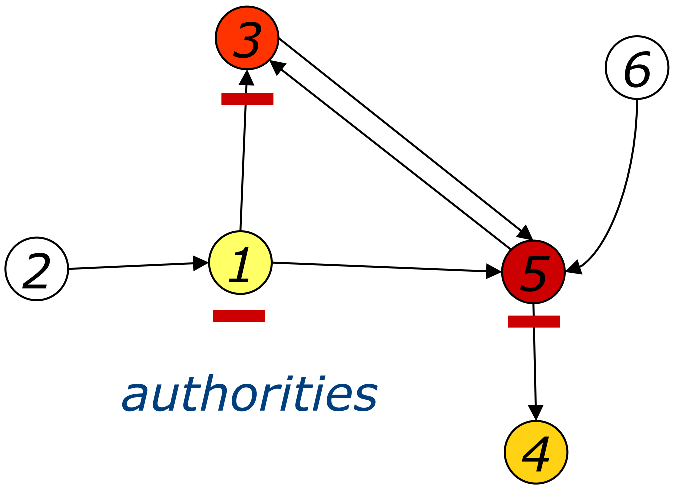
$$\underset{\underline{a}_1}{\begin{pmatrix} 0.258 \\ 0 \\ 0.516 \\ 0.258 \\ 0.775 \\ 0 \end{pmatrix}} = \begin{array}{c c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T \end{array} \underset{\underline{h}_0}{\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}}$$

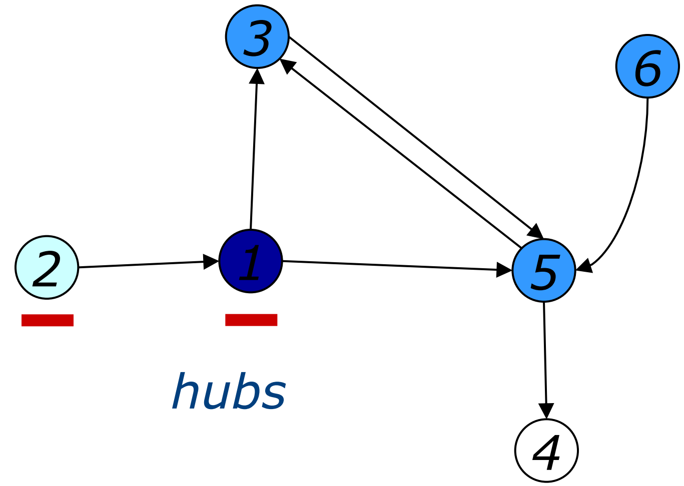
$$\begin{array}{c c} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} & \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \underset{\underline{a}_1}{\begin{pmatrix} 0.258 \\ 0 \\ 0.516 \\ 0.258 \\ 0.775 \\ 0 \end{pmatrix}} = \underset{\underline{h}_1}{\begin{pmatrix} 0.687 \\ 0.137 \\ 0.412 \\ 0 \\ 0.412 \\ 0.412 \end{pmatrix}}$$

(NOTE: normalization step is not shown, however results are normalized)

- $\underline{a}^{(2)} := L^T \underline{h}^{(1)}$
- $\underline{h}^{(2)} := L\, \underline{a}^{(2)}$

HITS: Example (3)





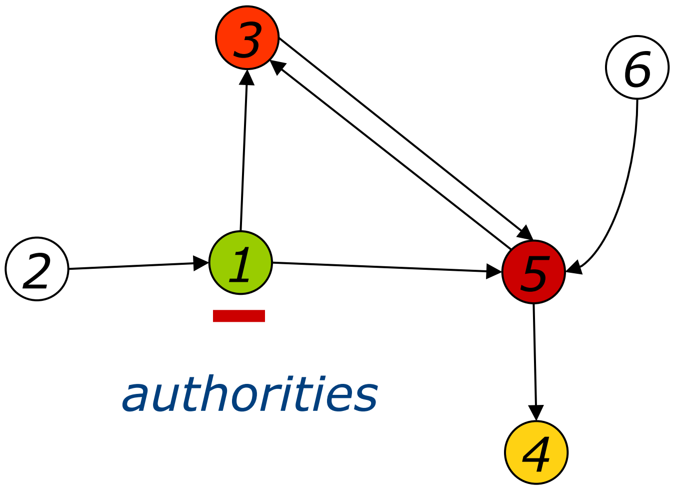
$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}
\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T \end{array}
\begin{array}{c} \underline{h}_1 \\ \begin{pmatrix} 0.687 \\ 0.137 \\ 0.412 \\ 0 \\ 0.412 \\ 0.412 \end{pmatrix} \end{array}
=
\begin{array}{c} \underline{a}_2 \\ \begin{pmatrix} \underline{0.072} \\ 0 \\ \underline{0.573} \\ 0.215 \\ \underline{0.788} \\ 0 \end{pmatrix} \end{array}
$$

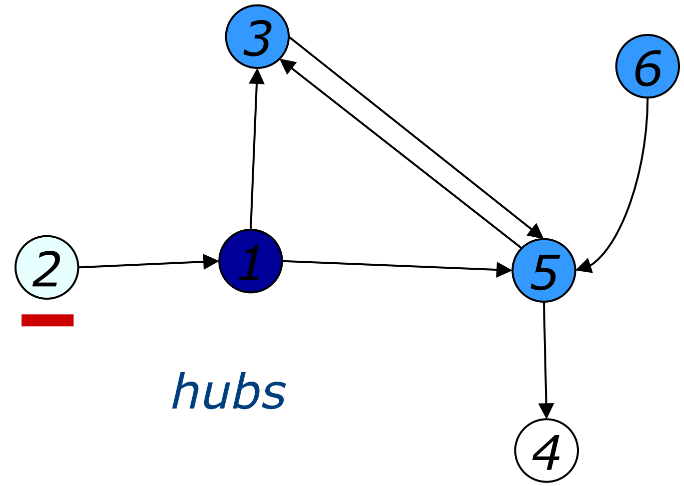
$$
\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array}
\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}
\begin{array}{c} \underline{a}_2 \\ \begin{pmatrix} 0.072 \\ 0 \\ 0.573 \\ 0.215 \\ 0.788 \\ 0 \end{pmatrix} \end{array}
=
\begin{array}{c} \underline{h}_2 \\ \begin{pmatrix} \underline{0.706} \\ \underline{0.037} \\ 0.409 \\ 0 \\ 0.409 \\ 0.409 \end{pmatrix} \end{array}
$$

HITS: Example (4)

– $\underline{a}^{(3)} := L^T\, \underline{h}^{(2)}$

– $\underline{h}^{(3)} := L\, \underline{a}^{(3)}$



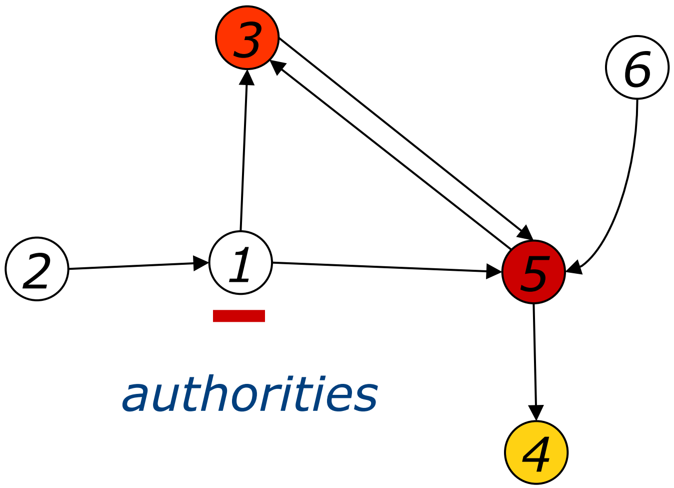


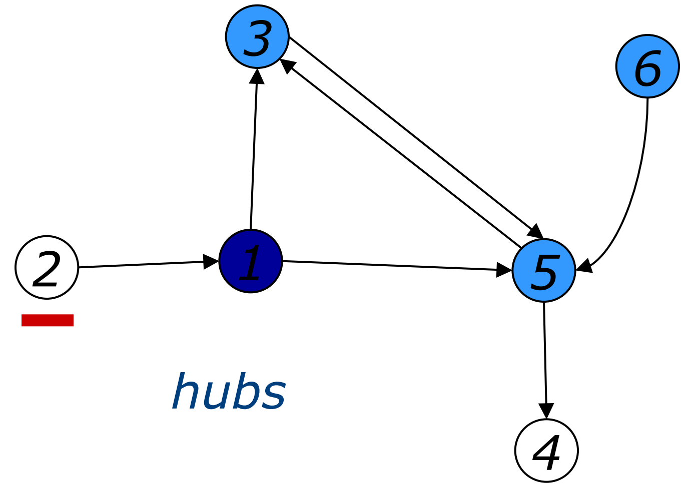
$$\begin{array}{c} \\ 1\\2\\3\\4\\5\\6 \end{array}\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T \end{array} \begin{array}{c} \underline{h}_2 \\ \begin{pmatrix} 0.706 \\ 0.037 \\ 0.409 \\ 0 \\ 0.409 \\ 0.409 \end{pmatrix} \end{array} = \begin{array}{c} \underline{a}_3 \\ \begin{pmatrix} \underline{0.019} \\ 0 \\ 0.577 \\ 0.212 \\ 0.789 \\ 0 \end{pmatrix} \end{array}$$

$$\begin{array}{c} \\ 1\\2\\3\\4\\5\\6 \end{array}\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array} \begin{array}{c} \underline{a}_3 \\ \begin{pmatrix} 0.019 \\ 0 \\ 0.577 \\ 0.212 \\ 0.789 \\ 0 \end{pmatrix} \end{array} = \begin{array}{c} \underline{h}_3 \\ \begin{pmatrix} 0.707 \\ \underline{0.001} \\ 0.408 \\ 0 \\ 0.408 \\ 0.408 \end{pmatrix} \end{array}$$

HITS: Esempio (5)

- $\underline{a}^{(4)} := L^T \underline{h}^{(3)}$
- $\underline{h}^{(4)} := L\, \underline{a}^{(4)}$





$$
\begin{array}{c} \\ 1\\2\\3\\4\\5\\6 \end{array}
\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}^T \end{array}
\begin{array}{c} \underline{h}_3 \\ \begin{pmatrix} 0.707 \\ 0.001 \\ 0.408 \\ 0 \\ 0.408 \\ 0.408 \end{pmatrix} \end{array}
=
\begin{array}{c} \underline{a}_4 \\ \begin{pmatrix} 0 \\ 0 \\ 0.577 \\ 0.211 \\ 0.789 \\ 0 \end{pmatrix} \end{array}
$$

$$
\begin{array}{c} \\ 1\\2\\3\\4\\5\\6 \end{array}
\begin{array}{c} \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\
\begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \end{array}
\begin{array}{c} \underline{a}_4 \\ \begin{pmatrix} 0 \\ 0 \\ 0.577 \\ 0.211 \\ 0.789 \\ 0 \end{pmatrix} \end{array}
=
\begin{array}{c} \underline{h}_4 \\ \begin{pmatrix} 0.707 \\ 0 \\ 0.408 \\ 0 \\ 0.408 \\ 0.408 \end{pmatrix} \end{array}
$$

Strengths and weaknesses of HITS

- Strength: its ability to rank pages according to the query topic, which may be able to provide more relevant authority and hub pages.
- Weaknesses:
  - It is easily spammed. It is in fact quite easy to influence HITS since adding out-links in one's own page is so easy.
  - Topic drift. Many pages in the expanded set may not be on topic.
  - Inefficiency at query time: The query time evaluation is slow. Collecting the root set, expanding it and performing eigenvector computation are all expensive operations

Applications of HITS

- Search engine querying (speed is an issue)
- Finding web communities.
- Finding related pages.
- Populating categories in web directories.
- Citation analysis
- Social network analysis

Link Analysis

- HITS (Hyperlink Induced Topic Serach) Jon *Kleinberg*
- **Page Rank** ***Larry Page, Sergei Brin***

Page Rank

- Ranks pages by authority.
- Applied to the **entire web** rather than a local neighborhood of pages surrounding the results of a query.
- **Not query-dependent**
- It is the Google algorithm invented by Brin&Page for ranking pages

PageRank----Idea

Every page has some number of out-links and in-links

PageRank----intuition

Two key ideas:

1. *Web pages vary greatly in terms of the number of backlinks (in-links) they have. For example, the Netscape home page has thousand in-links compared to most pages which have just a few.* ***Generally, highly linked pages are more "important" than pages with few links.***

EUGENE GARFIELD, FRANCIS NARIN, PAGERANK: THE THEORETICAL BASES OF THE GOOGLE SEARCH ENGINE

PageRank----intuition

2. ***In-links coming from important pages convey more importance to a page.*** *For example, if a web page has a link off the Yahoo home page, it may be just one link* ***but it is a very important one.***

A page has high rank if the sum of the ranks of its incoming links is high. This covers both the case when a page has many in-links and when a page has a few highly ranked in-links.

PageRank----Definition

u,v: a web page

F_u: set of pages that u points to (forwardlinks or out links)

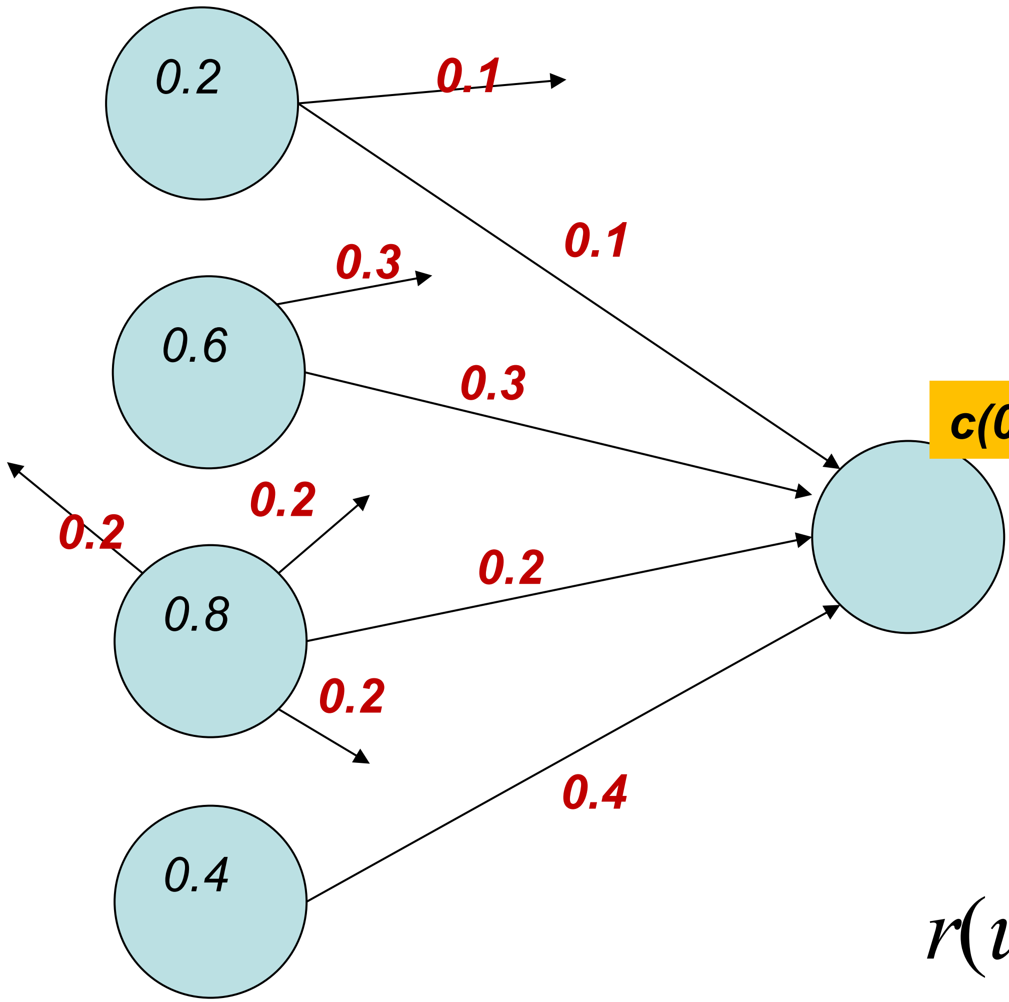
B_u: set of pages that point to u (backlinks or inlinks)

$N_u=|F_u|$: the number of links outgoing from u (outlinks)

c: a factor used for normalization

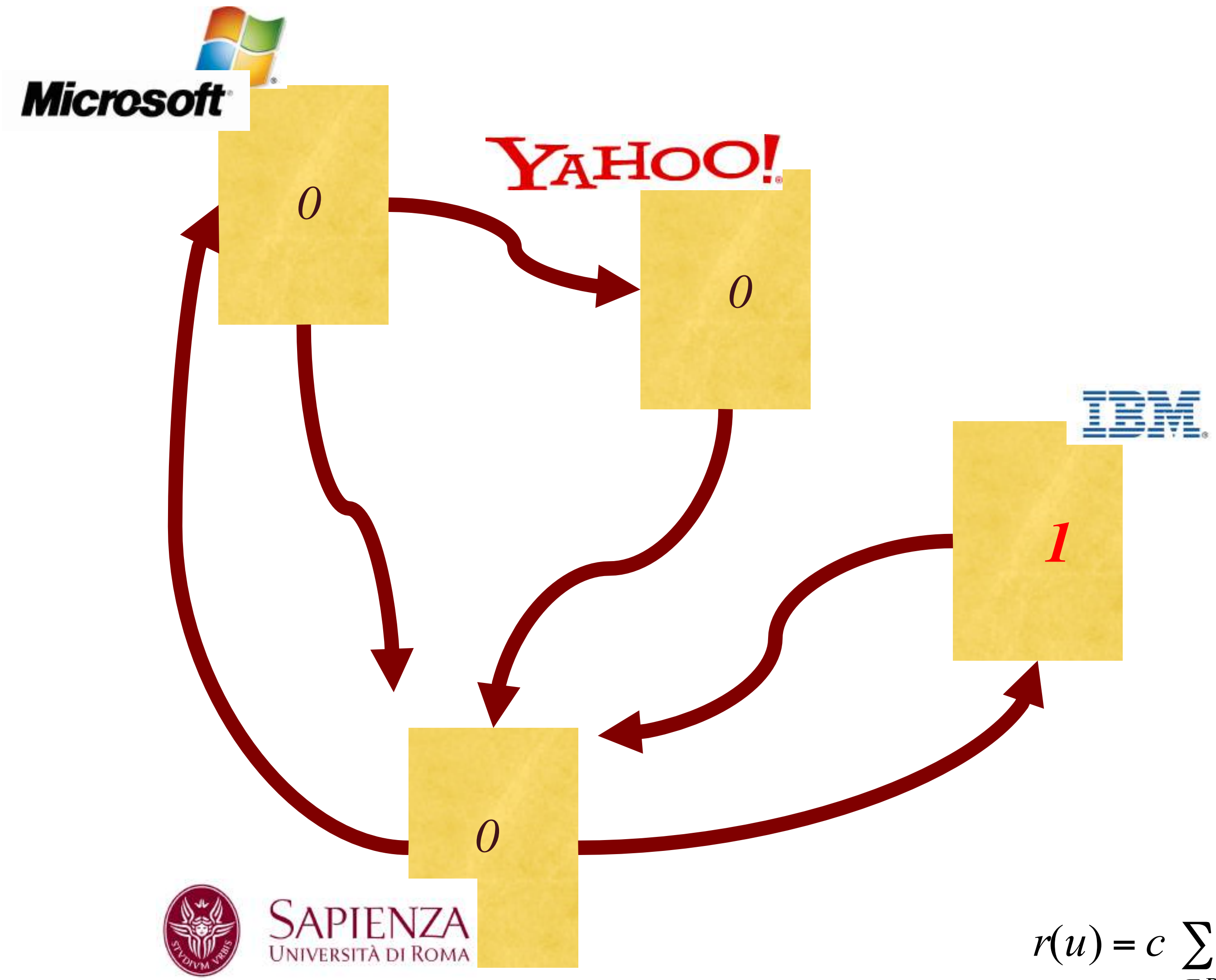
$$r(u) = c \sum_{v \in B_u} \frac{r(v)}{N_v}$$

*The equation is **recursive**, but it may be computed by starting with any set of ranks and iterating the computation until it converges.*

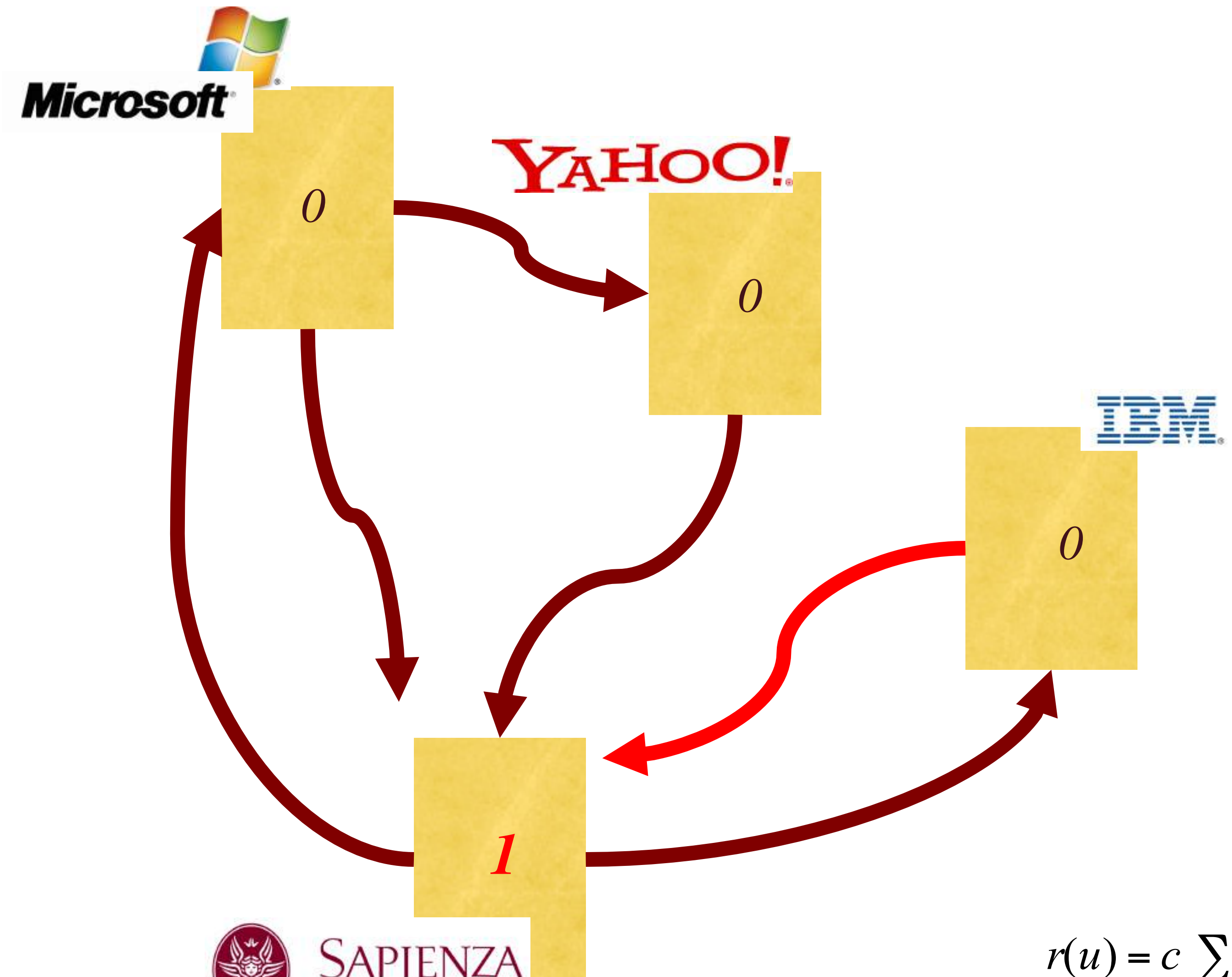


c(0,.1 +0.3+0.2+0.4)=c(1.0)

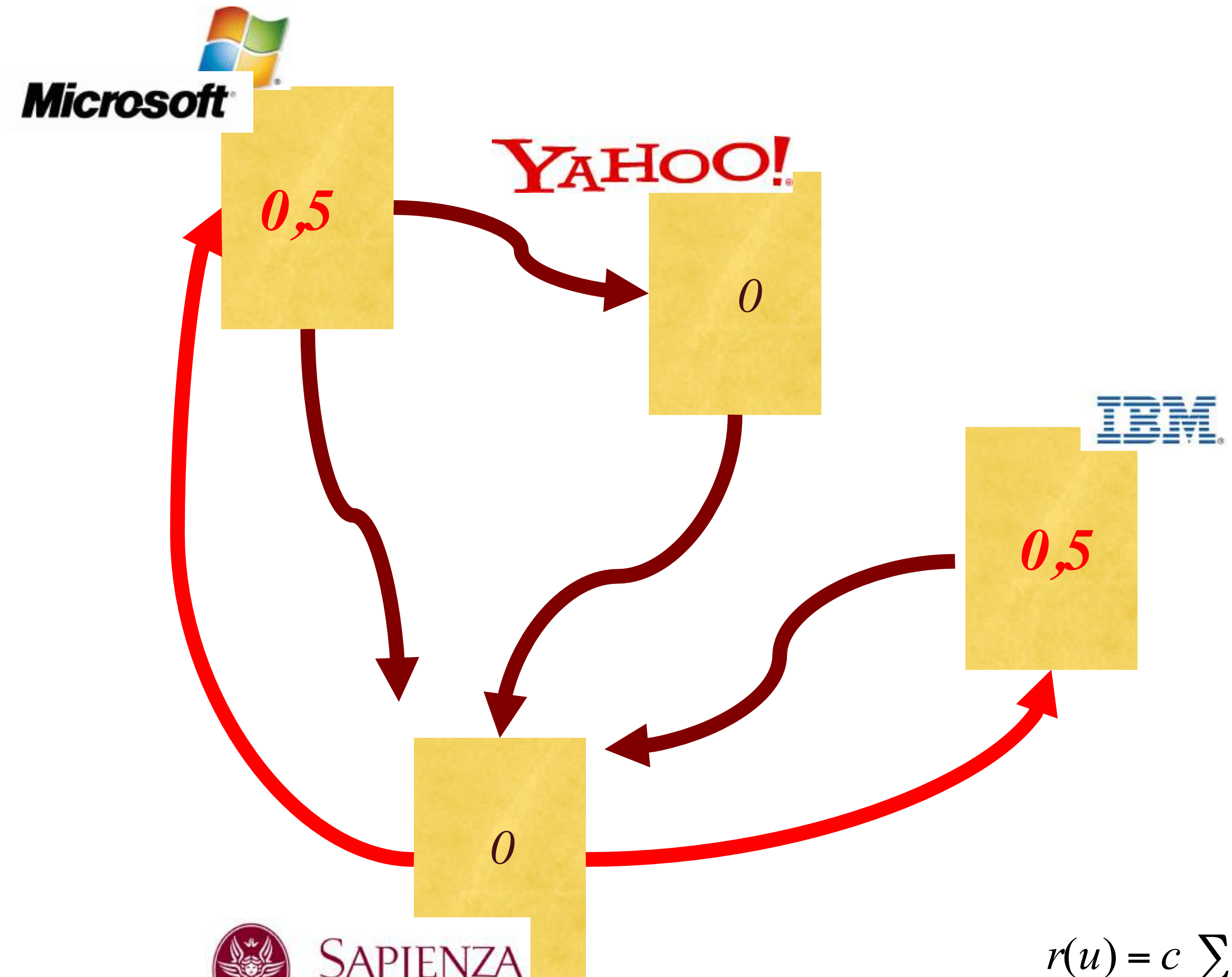
$$r(u) = c \sum_{v \in B_u} \boxed{\frac{r(v)}{N_v}}$$



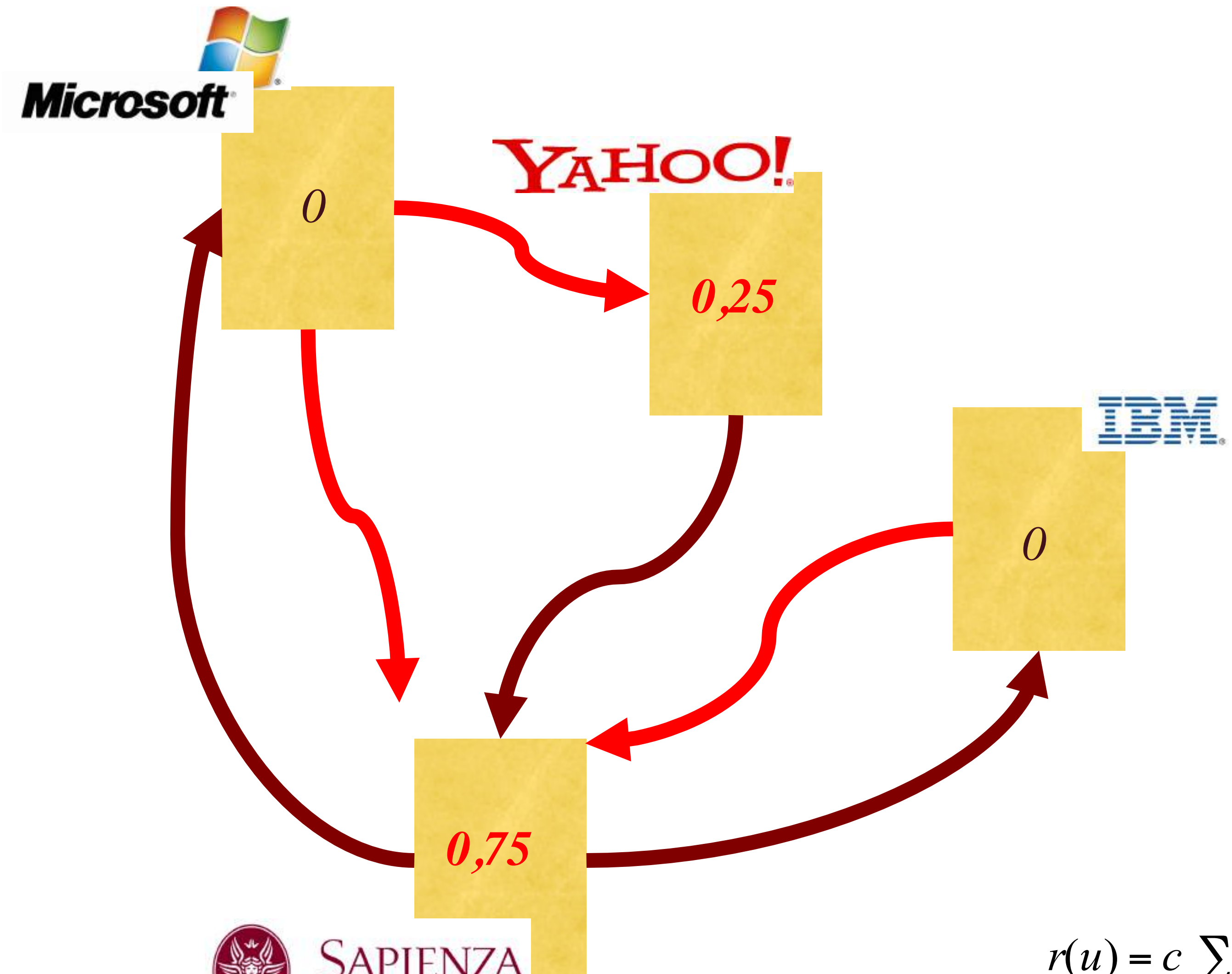
$$r(u) = c \sum_{v \in B_u} \frac{r(v)}{N_v}$$



$$r(u) = c \sum_{v \in B_u} \frac{r(v)}{N_v}$$

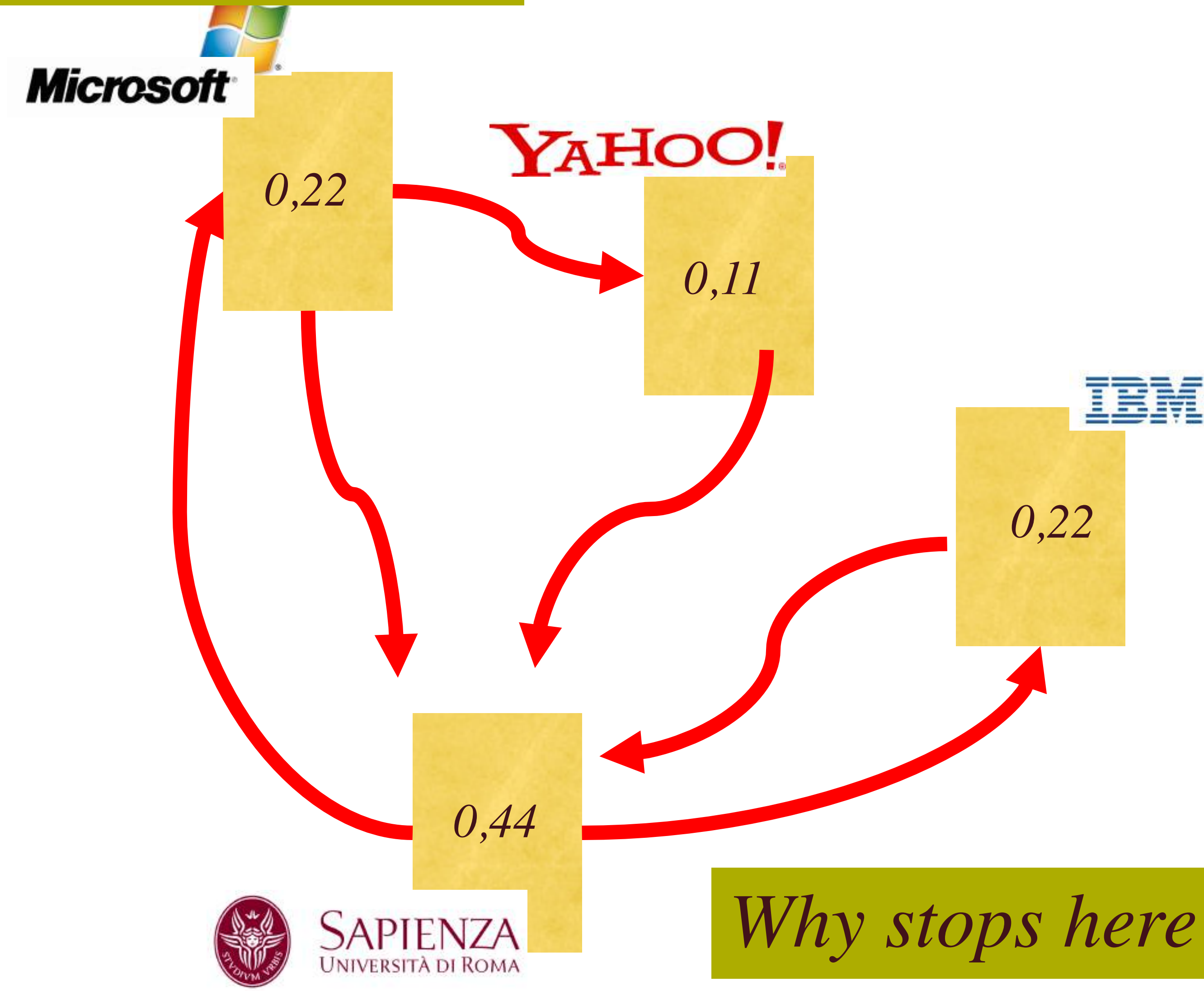


$$r(u) = c \sum_{v \in B_u} \frac{r(v)}{N_v}$$



$$r(u) = c \sum_{v \in B_u} \frac{r(v)}{N_v}$$

After several iterations..



Why stops here?

A probabilistic interpretation of PageRank

- *The definition corresponds to the probability distribution of a* ***random walk*** *on the web graphs.*
- *First note: we can write* $r(u) = c \sum_{v \in B_u} \frac{r(v)}{N_v}$ *in matrix iterative form as* $\mathbf{r}^t = P\mathbf{r}^{t-1}$
- *P* ***(transition matrix)*** *is* left stochastic *since* $\Sigma_j p_{ij} = 1$ *(column j in P corresponds to the outlink weights of node j, and for k outlinks, each weights 1/k and their sum is 1)*

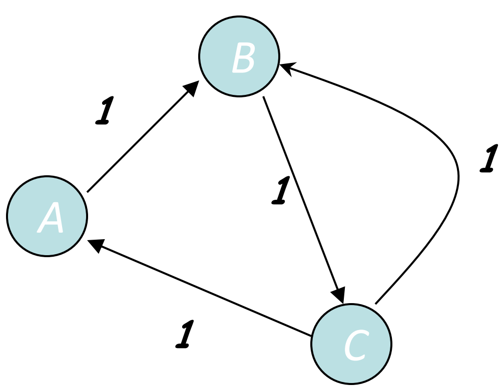
An example

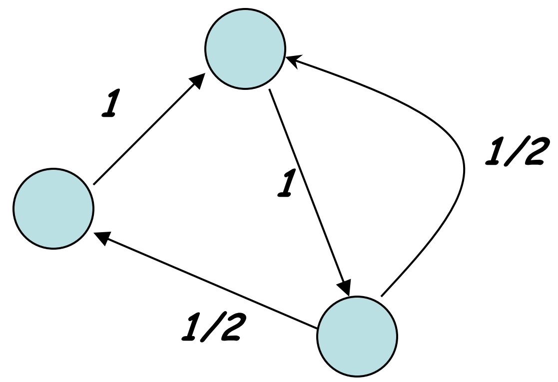
0	1	0
0	0	1
1	1	0

0	1	0
0	0	1
1/2	1/2	0

Adjacency matrix A

Transition matrix P



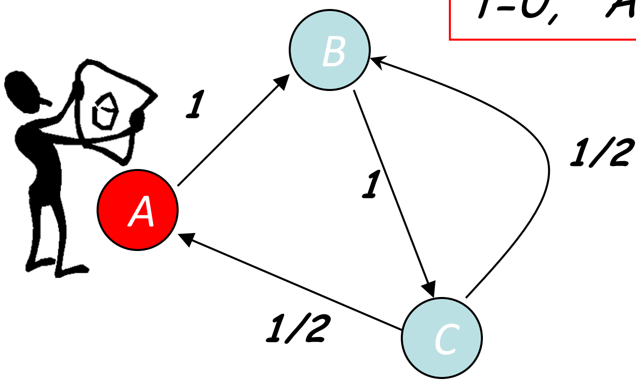


What is a Random Walk?

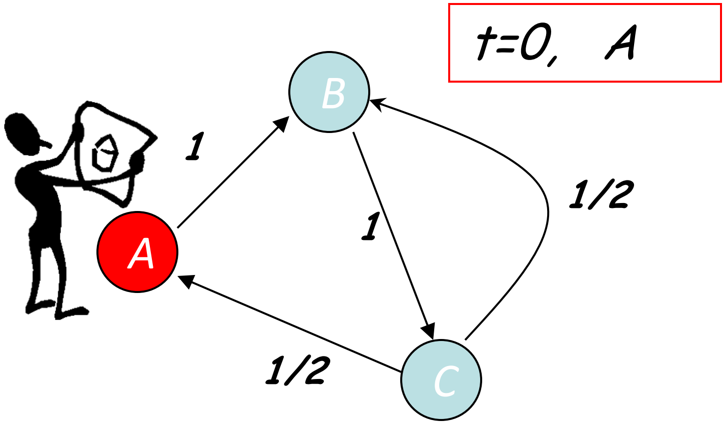
- Given a graph and a starting point (node), we select a neighbor (= a pointed node) of it at random, and move to this neighbor;
- Then we select (at random) a neighbor of this node and move to it, and so on;
- The (random) sequence of nodes selected this way is a random walk on the graph
- In our case, if the walker is on node j at time t, it has $1/k$ probability of jumping on any of its hyperlinked nodes at time $t+1$

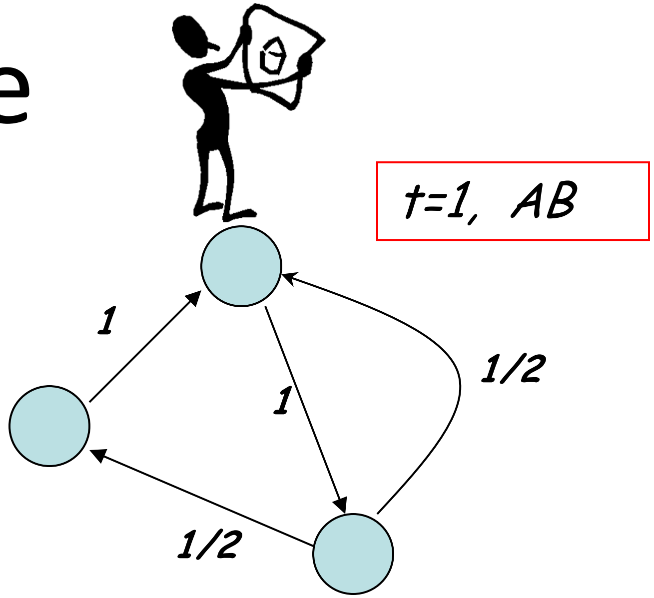
An example

t=0, A



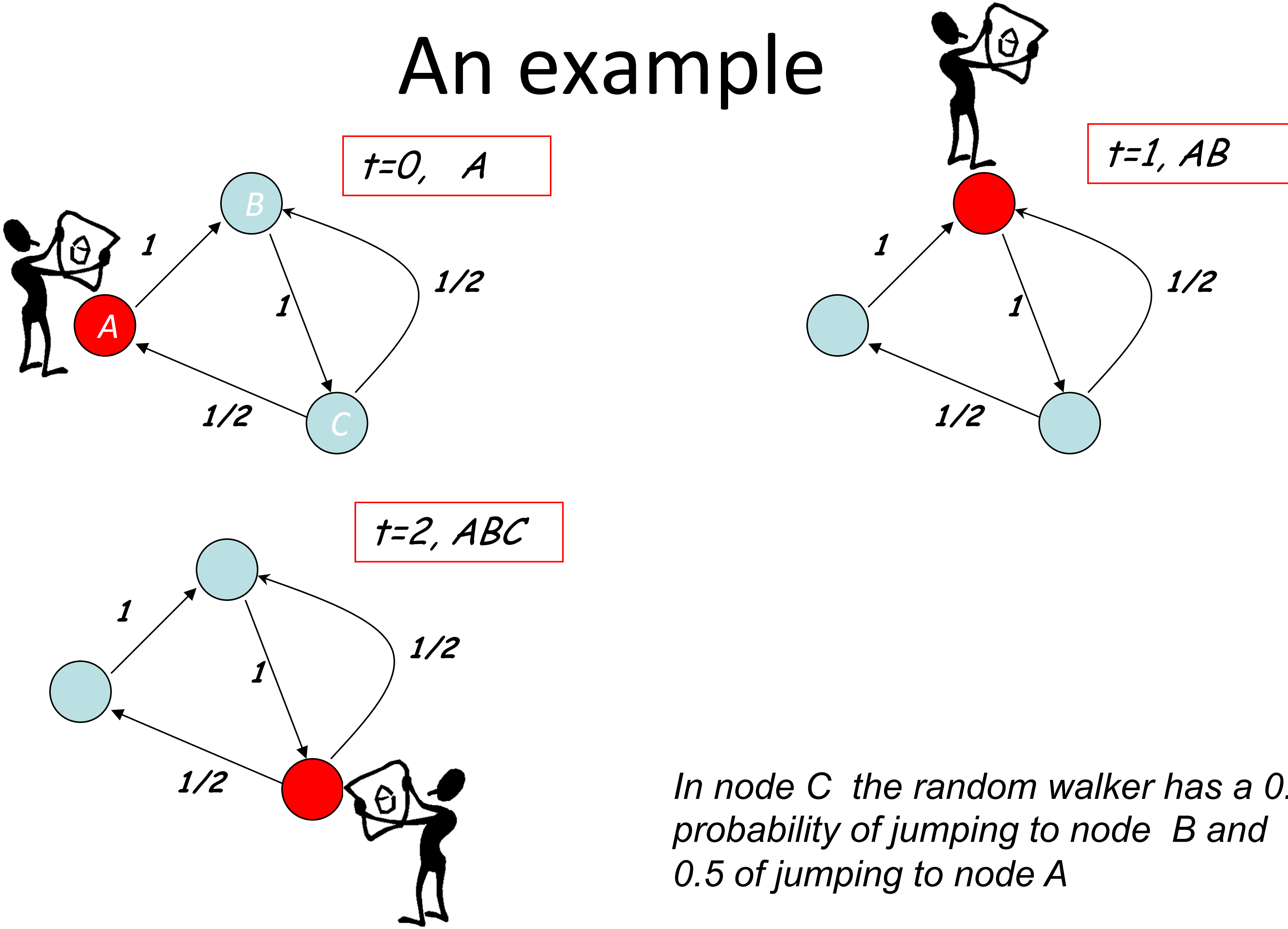
An example





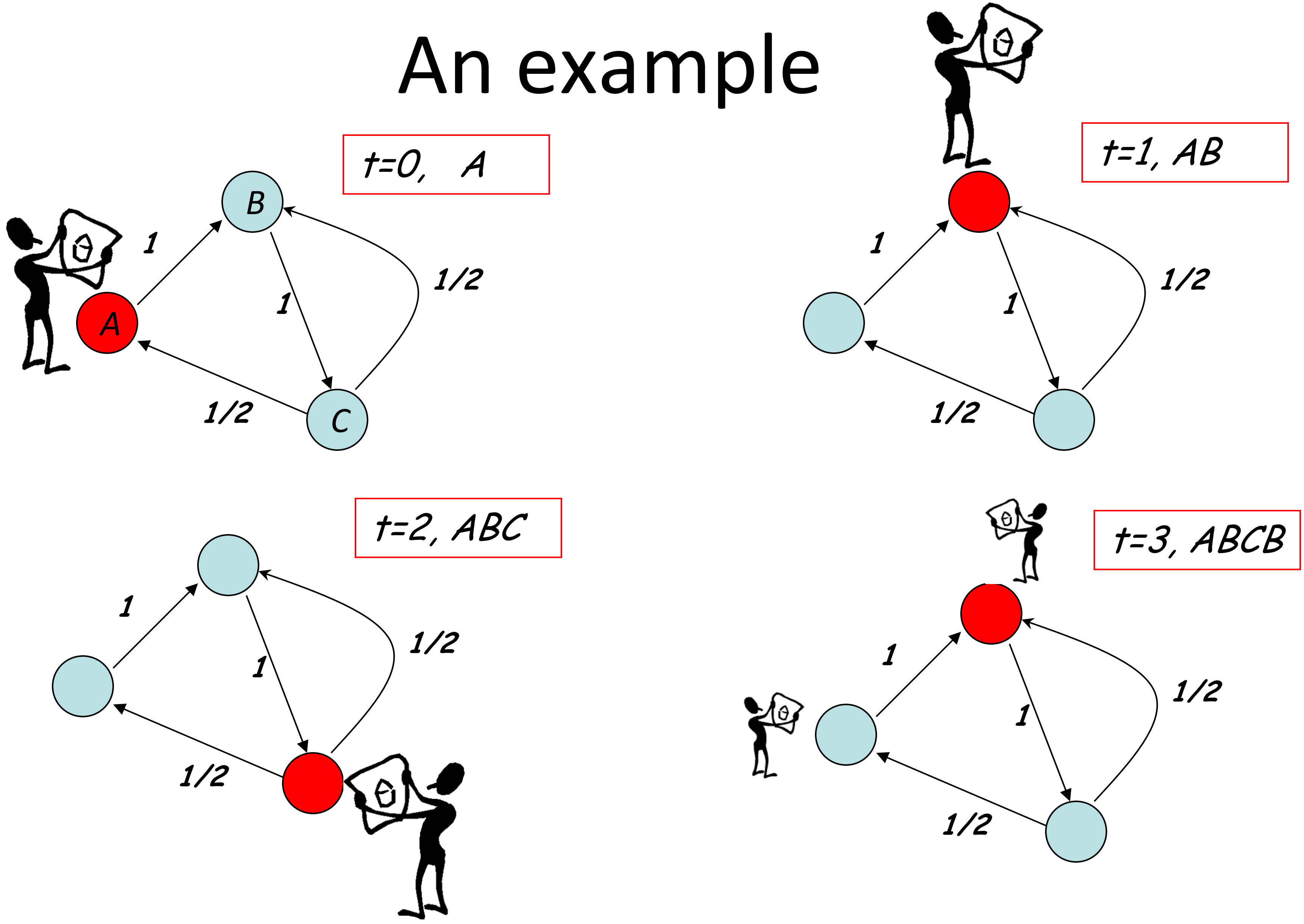
Slide from Purnamitra Sarkar, Random Walks on Graphs: An Overview

An example



In node C the random walker has a 0.5 probability of jumping to node B and 0.5 of jumping to node A

An example



Probabilistic interpretation

- N total number of web pages
- k outgoing links from page j
- P Transition matrix with elements:

$$\boldsymbol{p_{ij}} = \begin{cases} \boldsymbol{\frac{1}{k}\ if\ i \rightarrow j} \\ \boldsymbol{0} \\ \boldsymbol{p_{ii} > 0} \end{cases}$$

k is the number of outlinks of node i

$$\Sigma_i p_{ij} = 1$$

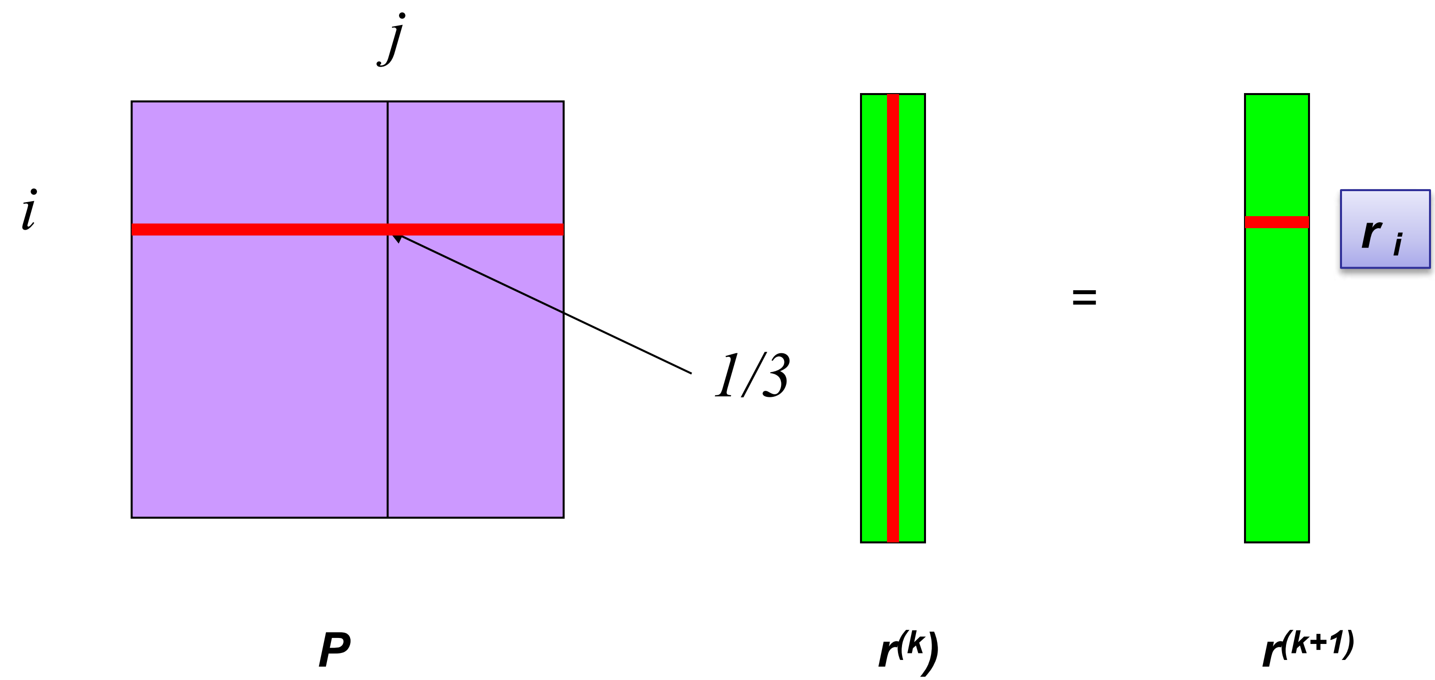
- The PageRank formulation can be written in matrix vector form as:

$$\boldsymbol{r}^t = P\,\boldsymbol{r}^{t-1}$$

P is a ***left stochastic*** *matrix, as we anticipated*

Example

Suppose page j links to 3 pages, including i



The new value of r_i (the page rank of node i) at iteration k is obtained by summing the r_j of all pages pointing to i, multiplied by their p_{ij} value: $r_i^{(k+1)}=\Sigma_j p_{ij} r_j^{(k)}$ where p_{ij} is the (uniform)probability of jumping from j to i (1/3 in this example).

How to compute the vector **r** of page ranks?

- We have the iterative formulation $\boldsymbol{r}^t = P\,\boldsymbol{r}^{t-1}$
- P is left stochastic but, contrary to HITs, it is not irreducible in general (we have now the entire web graph)
- So, how to reach convergence?
- The random surfer (or random walks) model can be represented using **Markov Chains**

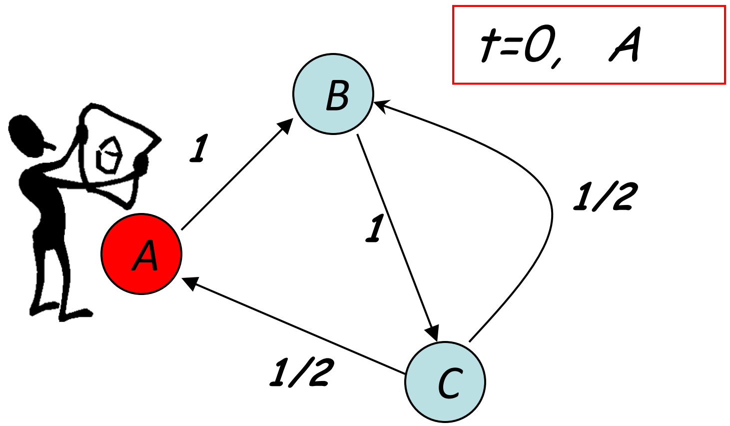
Markov Chains (1)

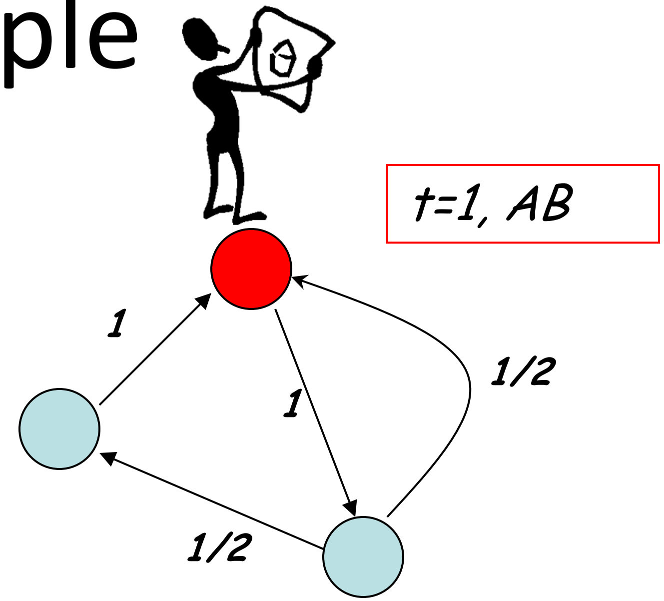
- A Markov Chain consists in N states (let S the set of possible states), and a matrix of transition probabilities $N \times N$, $\mathbf{P}$.
- At each step, the system is precisely in one state (states are web pages, in our case).
- For $1 \leq i,j \leq N$, $p(s_i \to s_j) = p_{ij}$ is the probability of jumping to s_j, given we are in s_i.
- Furthermore, if X_k is the random variable indicating the state **s** reached at time t_k (X gets values in S) , then:

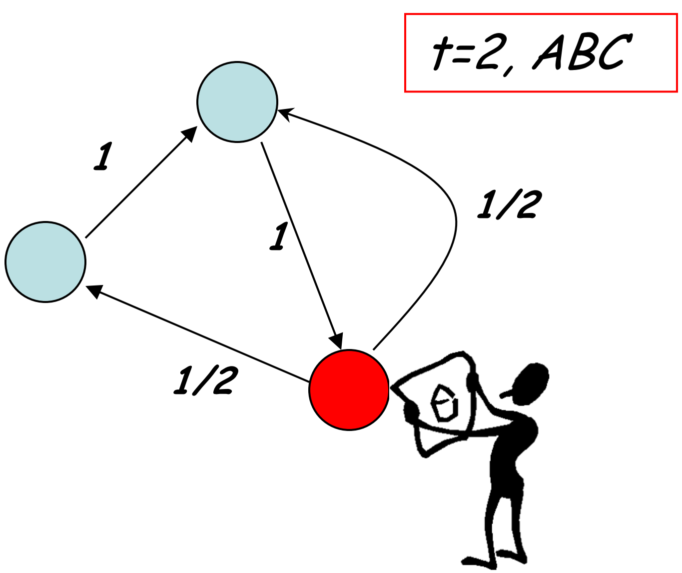
$$p(X_k = s_j / X_1 = s_{1j}, X_2 = s_{2j}, \ldots X_{k-1} = s_{(k-1)j}) = p(X_k / X_{k-1})$$

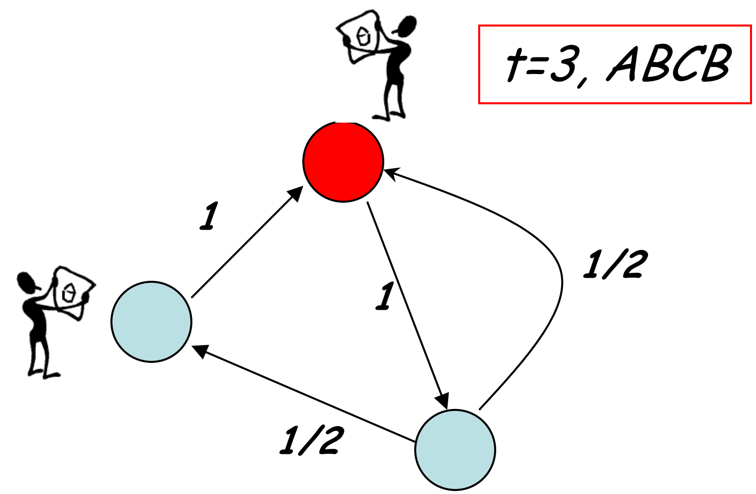
- The value of X_k at time t_k **depends only on the value of the random variable at time k-1!** (This is the basic property of Markov Chains: 1 state memory!)
- **Which means: the probability of being on page s_j at time k only depends on the page s_i on which the surfer was at time k-1**

Back to previous example









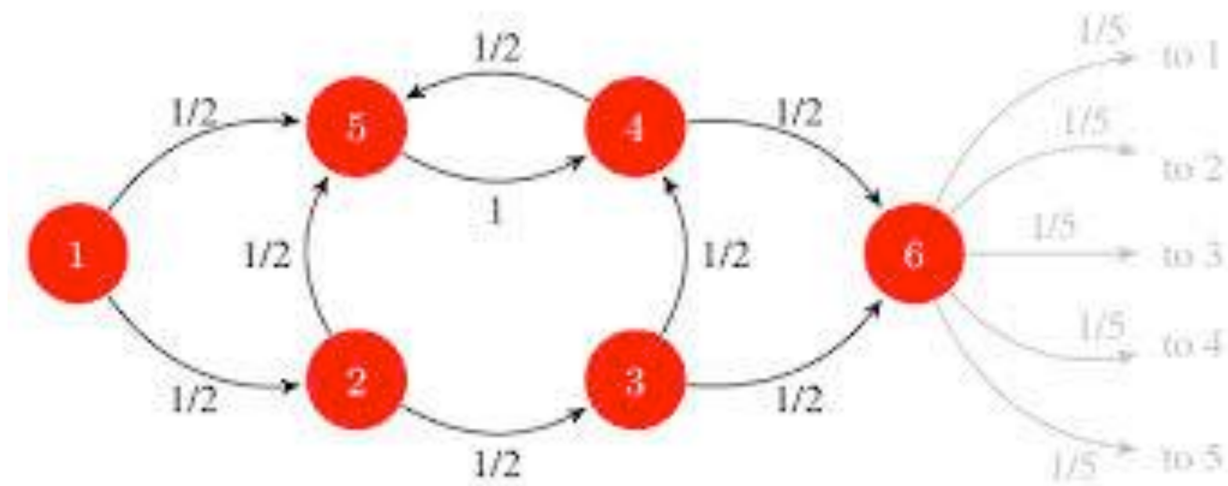
*P(**X4=B**/X1=A,X2=B,X3=C)=P(X4=B/X3=C) in a Markov Model!*

Markov chains (2)

- We also have that: $\Sigma_i p_{ij} = 1$

  $j=1$

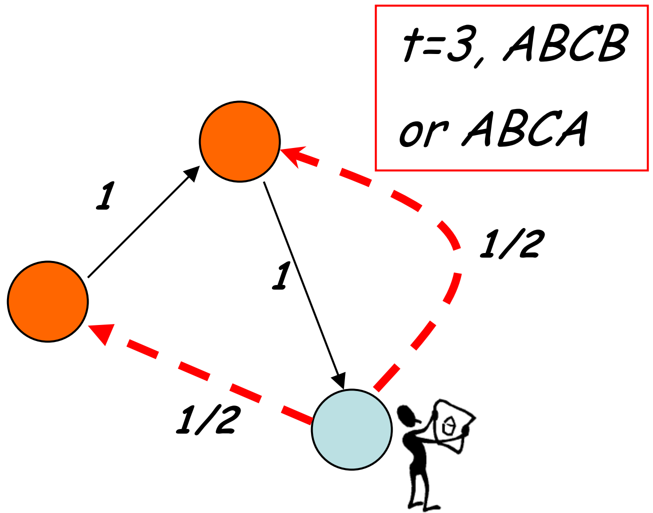
- Markov Chains are a model of random walks!



Probability Vectors

Since ***we are modeling a stochastic process****, we can define a* <u>*vector of probabilities*</u> $r^{(t)}=(p(t,s_1), \ldots p(t,s_N))= (r^t_1, \ldots r^t_N)$*, indicating that at step* $\mathbf{t}$ *the random walk will bring to state* $\mathbf{s_i}$ *with probability* $\mathbf{r^t_i}$*,*

Example:



$r^{(3)}=(0.5,0.5,0)$ since the walker in t=2 has a 0.5 probability of jumping To B and 0.5 of jumping to A in t=3

Difference between p_{ij} and $p(t,s_j)$

- p_{ij} is the probability **of jumping** in state (page) j when **we are** in state (page) i : those probabilities *are known and uniform* (all equal to 1/k for k outlinks) for any starting node i
- $p(t,s_j)=r^t_j$ is the probability **of being** in state s_j when starting in some initial state s_0 and **after t jumps** (=at time t).
- We wish to compute **stationary values r_i** for r^t_i !! These **are the Page Ranks**: *the asinthotic probabilities ($t \rightarrow \infty$) of being in a given page for a walker who start in a page at random and travels the web graph at random*.

Ergodic Markov Chains

- The Random Walk is modeled with Markov Chains. Stationary probabilities can be computed if the process is ERGODIC
- A Markov Chain is **ergodic** if:
  - There is a path between any pair of states (= adjacency matrix is irreducible,➔the graph is connected)
  - Starting from any state, after a finite transition time T_0, the probability to reach any other state in a finite time <u>$T>T_0$ is always different from zero.</u>
  - <u>Note</u>: **not true for the web graph**! Since not fully connected and cyclic. Will see how to cope with this

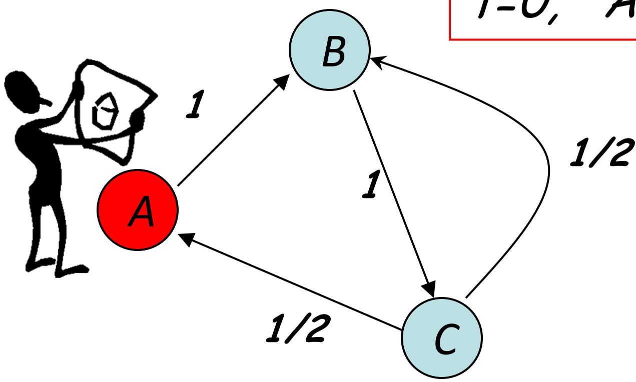
Ergodic Chains

- If a Markov Chain is ergodic, every state has a **stationary probability** of being visited, regardless of the initial state of the random walker .
  - The vector $\mathbf{r}^{(t)}$ of state probabilities converges to a **stationary vector** $\mathbf{r}$ as $t \rightarrow \infty$

Computing State Probability Vector

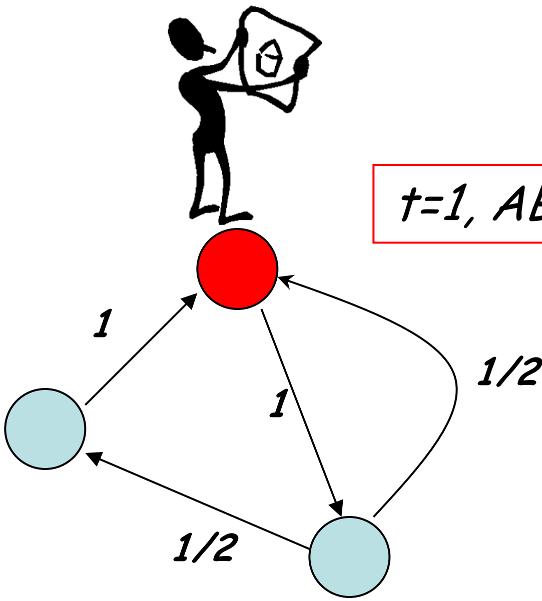
- If $\mathbf{r}^{(\mathbf{t})} = (r_1, \ldots r_N)$ is the probability vector in step t, how would it change after the next jump?
- The adjacency matrix **P** tells us where we are likely to jump from any state *(since it has all transition probabilities from s_i to the other linked states)*:
- Therefore, from $\mathbf{r}^{(\mathbf{t})}$, the probability of next state $\mathbf{r}^{(\mathbf{t+1})}$ is computed according to: ($\mathbf{r}^{(\mathbf{t+1})} = \mathbf{P}\mathbf{r}^{\mathbf{t}}$)
- Even under the random walk model, *we obtain again our iterative formulation*!

t=0, A



$$(100)\times\begin{pmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ \frac{1}{2} & \frac{1}{2} & 0\end{pmatrix}=\begin{pmatrix}0\\ 1\\ 0\end{pmatrix}$$

$r^{(0)}$ $\qquad$ $r^{(1)}$

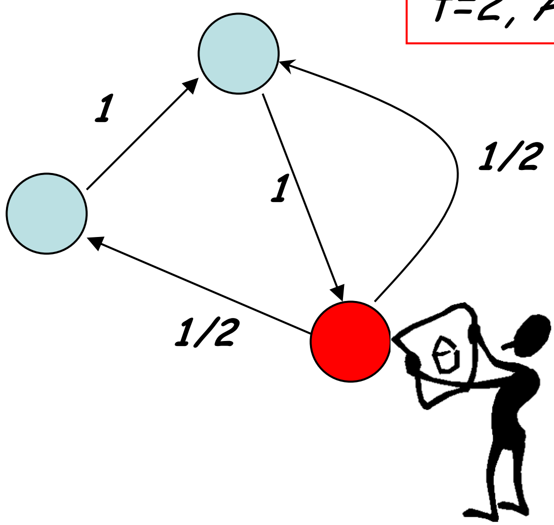


t=1, AB

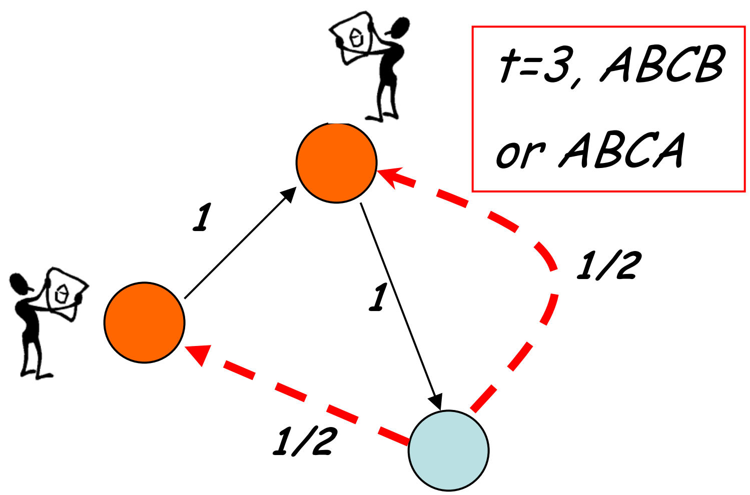
$$(010)\times\begin{pmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ \frac{1}{2} & \frac{1}{2} & 0\end{pmatrix}=\begin{pmatrix}0\\0\\1\end{pmatrix}$$

$r^{(1)}$ $\uparrow$ $r^{(2)}$

t=2, ABC



$$\underbrace{(001)}_{r^{(2)}} \times \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \\ 0 \end{pmatrix}}_{r^{(3)}}$$



$$\underset{r^{(3)}}{\left(\frac{1}{2}\,\frac{1}{2}\,0\right)} \times \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} = \underset{r^{(4)}}{\begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}}$$

Computing Stationary Probability Vector

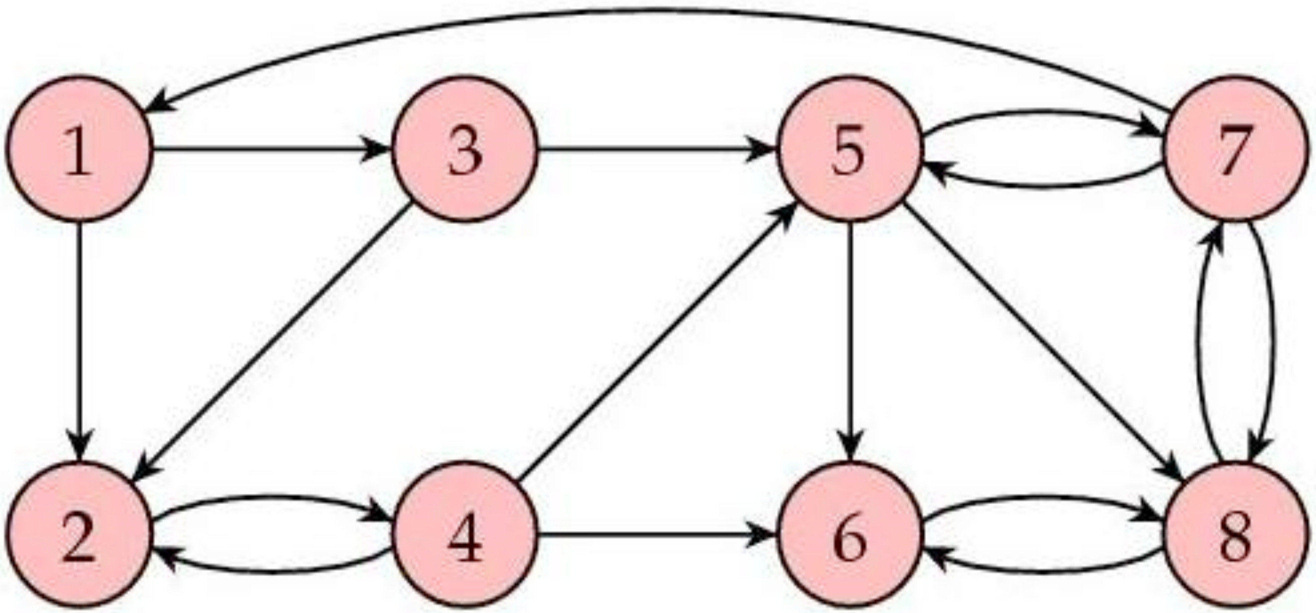
- If the process is ergodic, $\mathbf{r}^t$ will eventually converge to a stationary vector r such that $r = Pr$
- **Since P is a matrix and *r* is a vector, what kind of vector is *r*?? In other terms, if *r=Pr* holds, what vector is r??**

Eigenvector of P with eigenvalue 1!!

Again: the Power method!

- $\underline{r}^{(t+1)} = P\underline{r}^{(t)}$ where P is a square stochastic matrix and r^t r^{t+1} are stochastic vectors
- The **sequence of vectors r^t converge to the stationary vector r** **if P is stochastic and irreducible**
- To compute **r** we use the same method as for HITS
- $\underline{r}^{(t+1)} = P\underline{r}^{(t)} = P^t\underline{r}^{(0)} = \lambda_1[\alpha_1 r_1 + \alpha_2 (\frac{\lambda_2}{\lambda_1})^t r_2 + \cdots + \alpha_n (\frac{\lambda_n}{\lambda_1})^t r_n]$
- The method converges provided there is a dominant **(principal) eigenvector**
- Since the stationary condition is: **r=Pr**, **r** is the <u>principal eigenvector</u> r_1 of **P** **and** $\lambda_1 = 1$
- The principal eigenvalue is 1 (because P is stochastic), all other egeinvalues are <1 (demonstration is <u>here</u>)

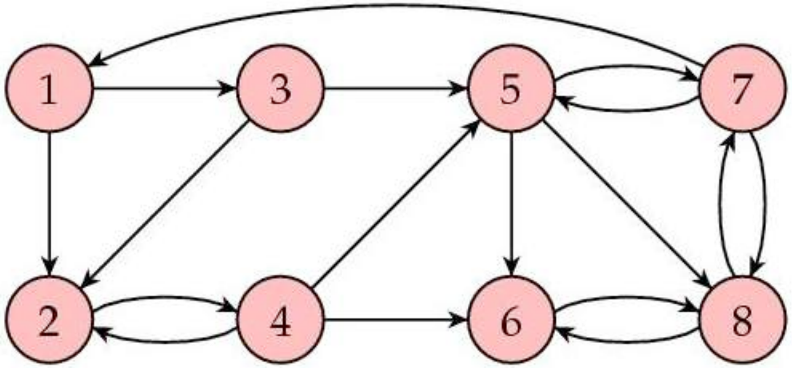
Example



The normalized adjacency matrix P

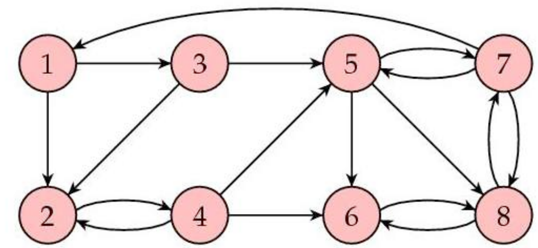
$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1/3 & 0 \\ 1/2 & 0 & 1/2 & 1/3 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/2 & 1/3 & 0 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1/3 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 1 & 1/3 & 0 \end{bmatrix}$$

$1/N(u_i)$



Iterations

$\underline{r}^{(t+1)}=P\underline{r}^{(t)}$



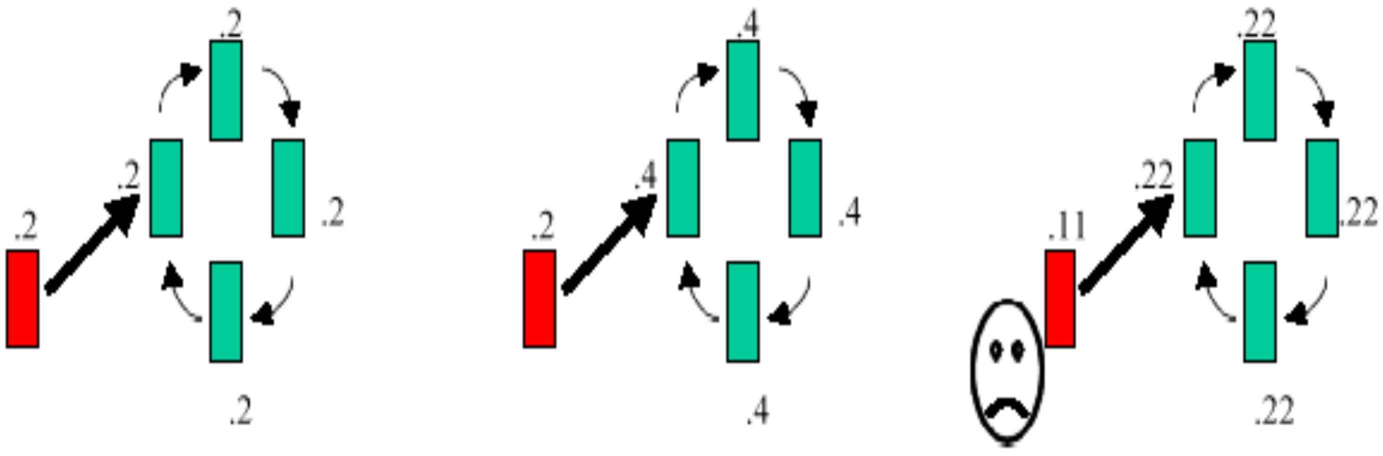
r^0	r^1	r^2	r^3	r^4		r^{60}	r^{61}
1	0	0	0	0.0278	...	0.06	0.06
0	0.5	0.25	0.1667	0.0833	...	0.0675	0.0675
0	0.5	0	0	0	...	0.03	0.03
0	0	0.5	0.25	0.1667	...	0.0675	0.0675
0	0	0.25	0.1667	0.1111	...	0.0975	0.0975
0	0	0	0.25	0.1806	...	0.2025	0.2025
0	0	0	0.0833	0.0972	...	0.18	0.18
0	0	0	0.0833	0.3333	...	0.295	$\boxed{0.295}$

Problem with our PageRank formulation

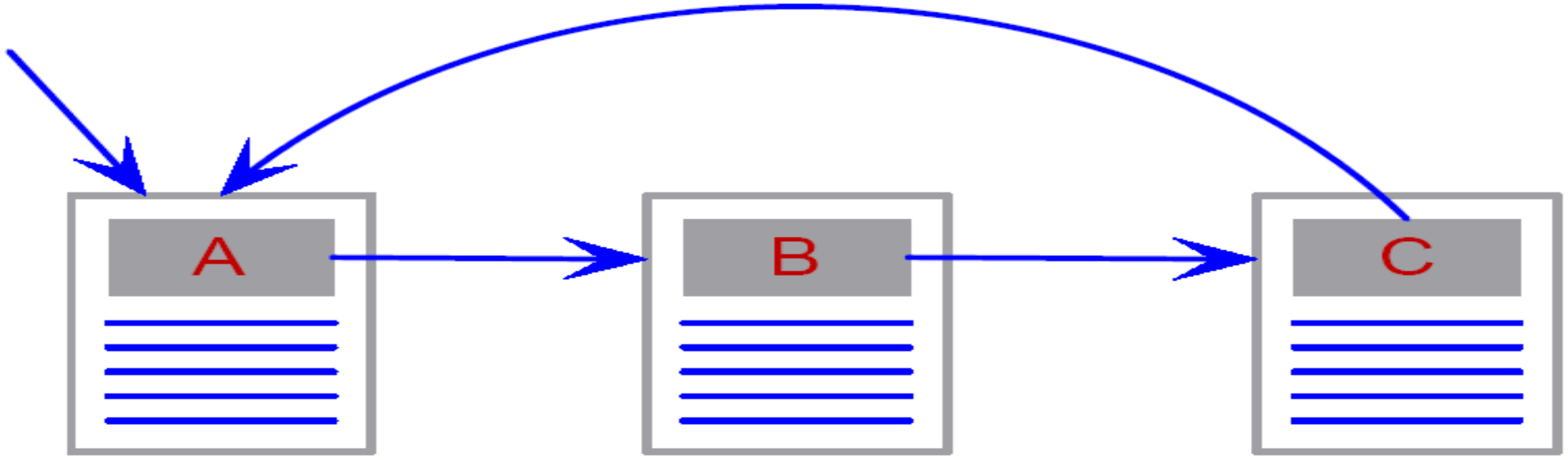
- Markov process converges under condition of ergodicity. Iterative computation (power method) converges if matrix P is irreducible. Two ways of saying the same thing!
  - As we said, these conditions are satisfied if the graph is fully connected and not deeply cyclic, **which is not the case** for the web graph
  - What causes the problem in practice?

Example: Rank synk

If a group of web pages point to each other but to no other page, during the iteration, this loop will accumulate rank but never distribute any rank.



Solution: Teleporting



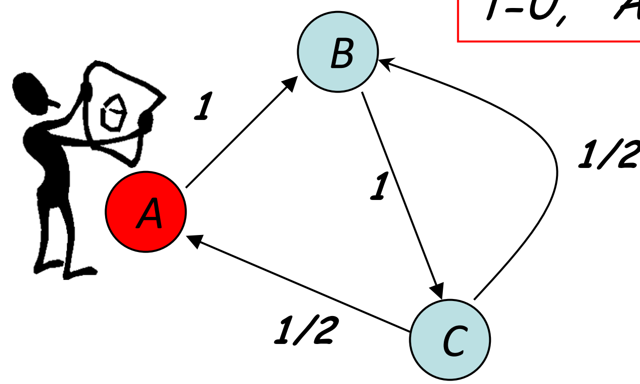
- Problem: Pages in a loop (or in a disconnected component) accumulate rank but do not distribute it to the rest of the graph.
- Solution: Teleportation, i.e. with a certain small probability the surfer can jump to any other web page (to which it is not connected) to get out of the loop.

Page rank Definition modified (with teleporting)

$$r(u) = c \sum_{v \in B_u} \frac{r(v)}{N_v} + (1-c)E(u)$$

- *E(u) is a* ***vector of probabilities*** *over the set of web pages (for example uniform prob., favorite page etc.) that corresponds to a source of rank.*
- *c is called the* ***dumping factor*** *(sometimes denoted with d)*
- *E(u) can be thought as if the random surfer "gets bored" periodically to travel from one page to another adjacent page, and "flies" to a different page (even though not connected), so he is not kept in a loop forever.*

$t=0,\quad A$



NOTE: ***needs Rank-normalization step*** *to get $\Sigma r_i=1$*

$$0.9 \times (100) \times \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} + 0.1 \times \begin{pmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0.033 \\ 0.033 \\ 0.033 \end{pmatrix} = \begin{pmatrix} 0.033 \\ 1.033 \\ 0.033 \end{pmatrix}$$

$$c \sum_{v \in B_u} \frac{r(v)}{N_v} + (1-c)E(u) = r(u)$$

Note that in case of teleporting we can still use the matrix formulation r^{t+1}=Pr^t

The transition matrix *is now* $P = cP' + (1-c)R'$

- P' *is a stochastic matrix where* P_{ij} *is the probability that the surfer jumps from i to j (0 if i is not connected to j)*
- R' *is (in case of uniform teleporting) a stochastic matrix where* R_{ij}*=1/N for all i,j*

  Example:

$$P = 0.9\times\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix} + 0.1\times\begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{pmatrix}$$

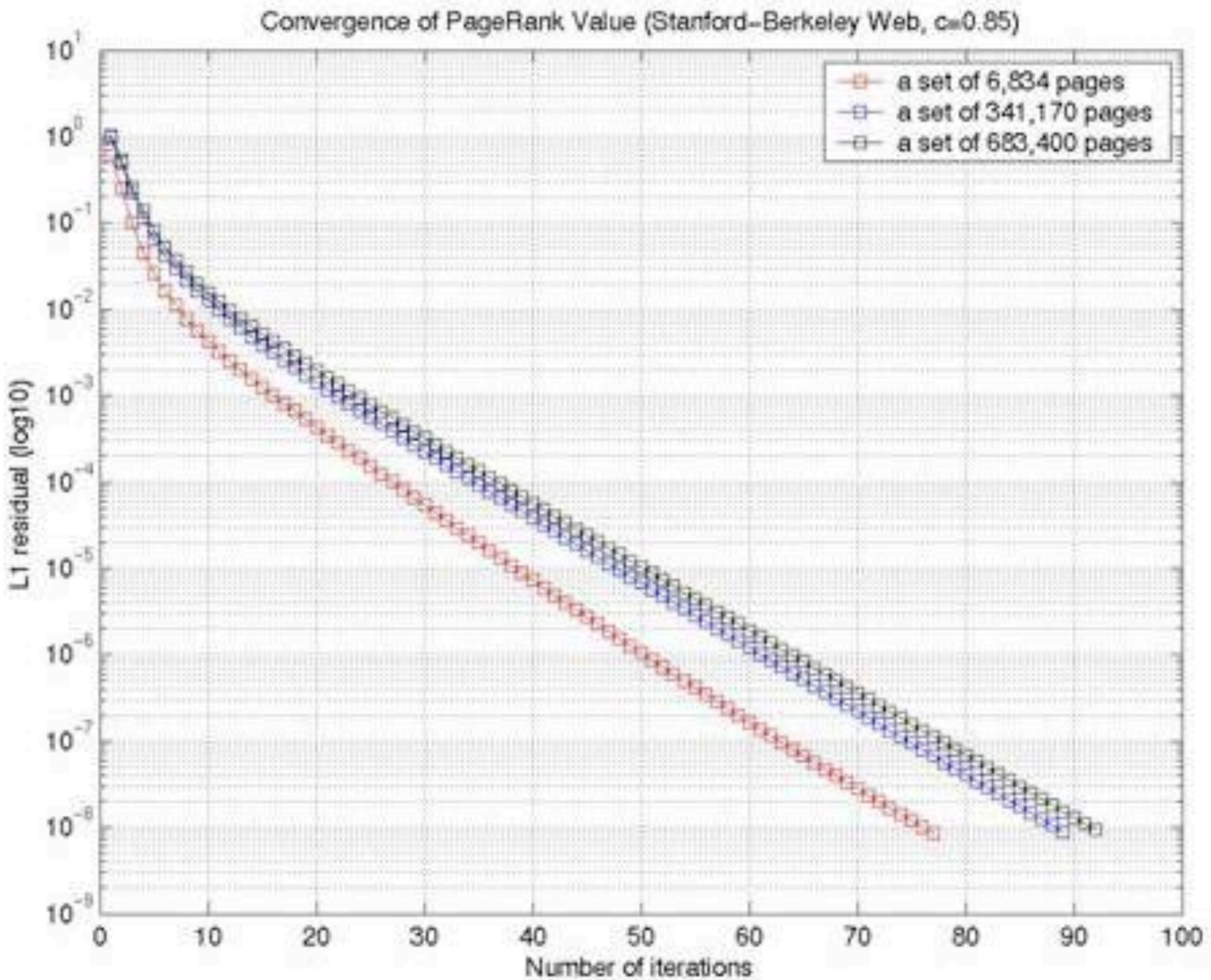


Figure 5: PageRank convergence as a function of the size of the web graph

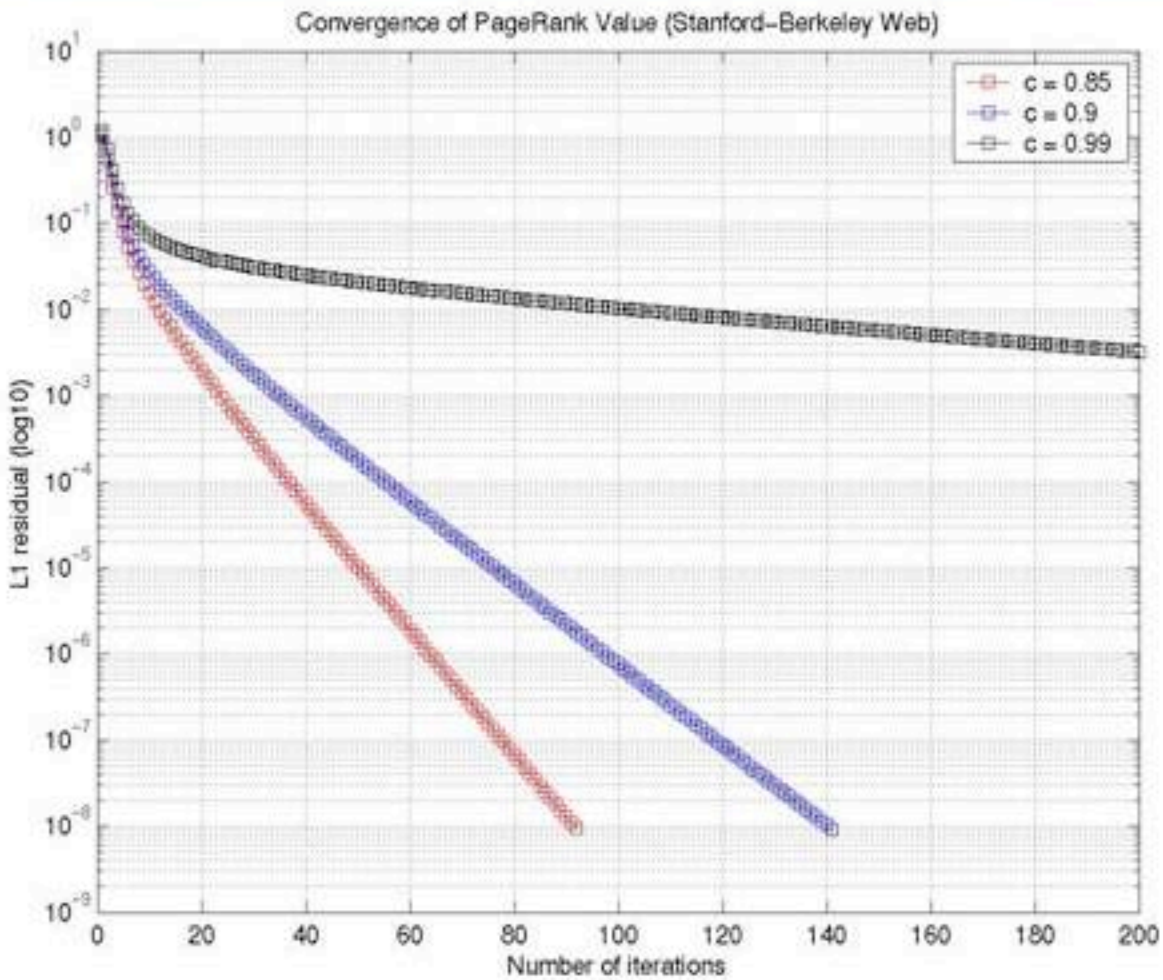


Figure 6: PageRank convergence as a function of C

Note: "c" is the dumping factor

Teleporting is a great “trick”

- This solves:
  - Sink problem
  - Disconnectedness of the web graph
  - Converges fast if *set of initial rank values* $\boldsymbol{r}^{(0)}$ is chosen appropriately. In algebraic terms, the initial vector must have <u>a non-zero component in the direction of the principal eigenvector</u> (else it will never move in that direction)
- But we still have problems:
  1. Computing Page Rank for all web pages is computationally very intensive, plus needs frequent updates (web is dynamic)
  2. Does not takes into account the specific query
  3. Easy to fool (less than HITS: less easy to be cited than cite!)

The Largest Matrix Computation in the World

- Computing PageRank can be done via matrix multiplication, where the matrix has billions rows and columns.
- The matrix is sparse as average number of outlinks is between 7 and 8.
- Setting c = 0.85 or above requires at most 100 iterations to convergence.
- Researchers still trying to speed-up the computation (“Big Data” problem, but you have a “Big Data” course).

Monte Carlo Methods in Computing PageRank

- Rather than following a single long random walk, the random surfer can follow many “sampled” random walks (*threads*, like for crawlers).
- Each walk starts at a random page and either teleports with probability c or continues choosing a connected link with uniform probability.
- The PageRank of a page is the proportion of times a “sample” random walk ended at that page.

Another variant: Personalised PageRank

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v} + (1-c)v$$

- Change *c*E*(v)* with *c**v***
- Instead of teleporting uniformly to any page we *bias* the jump to prefer some pages over others.
  - E.g. *v* is 1 for “your home page” and 0 otherwise.
  - E.g. *v* prefers the topics you are interested in.

Summary

- Link analysis one of the main mechanisms to rank web pages (others are content-based methods and personalization)
- PageRank is the most well know and used, HITS is used but more in social networks (due to query-time computation delay)