PageRank

Social Networks Analysis and Graph Algorithms

Prof. Carlos Castillo — https://chato.cl/teach

Universitat Pompeu Fabra Barcelona

Sources

- D. Easley and J. Kleinberg (2010): Networks, Crowds, and Markets – Chapter 14
- Fei Li's lecture on PageRank (2011)
- Evimaria Terzi's lecture on link analysis (2013)
- URLs in the footer of specific slides

Back to the 1990s ...

The early days of the web

- March 1989: proposal by Tim Berners-Lee at CERN
- Early 1993: NCSA Mosaic graphical browser
- Jan 1994: Yahoo! Web directory (manual)
- 1994: WebCrawler, Lycos (automated, crawlers)
- End of 1994: the web has about 10,000 sites
- 1995-1996: Altavista, Inktomi, and many others ...

Backrub!

Part of a research project that started in 1995 ...

Search The Web (type only necessary words):

10 results | clustering on | Search

Current Repository Size: ~25 million pages (searchable index slightly smaller)

Research Papers about Google and the WebBase

Credits

Current Development: Sergey Brin and Larry Page
Design and Implementation Assistance: Scott Hassan and Alan Steremberg
Faculty Guidance: Hector Garcia-Molina, Rajeev Motwani, Jeffrey D. Ullman, and Terry Winograd
Equipment Donations: IBM, Intel, and Sun
Software: GNU, Linux, and Python
Collaborating Groups in the Computer Science Department at Stanford University: The Digital Libraries Project, The Project on People Computers and Design, The Database Group, The MIDAS Data Mining Group, and The Theory Division
Outside Collaborators: Interval Research Corporation and the IBM Almaden Research Center
Technical Assistance: The Computer Science Department's Computer Facilities Group, Stanford's Distributed Computing and Intra-Networking Systems Group

Note: Google is research in progress and there are only a few of us so expect some downtimes and malfunctions. This system used to be called Backrub.

New! Wonder what your search runs on? Here are some pictures and stats for the Google Hardware.

1. This new index contains only a very limited number of international pages because we do not want to congest busy international links.
2. When no documents match your query, the system will return 20000 random web pages.
3. For improved speed, try to avoid common words unless they are necessary, and use as few search terms as possible.

Before emailing a question please read the FAQ. Thanks! We can be reached at google@google.stanford.edu and we appreciate your comments.

Subscribe to google-friends

This is a moderated list with about one message per month

your e-mail

Subscribe

FindMail List Archive

PageRank

The **PageRank** citation ranking: Bringing order to the web. *[link]*

L Page, S Brin, R **Motwani**, T Winograd - 1999 - ilpubs.stanford.edu

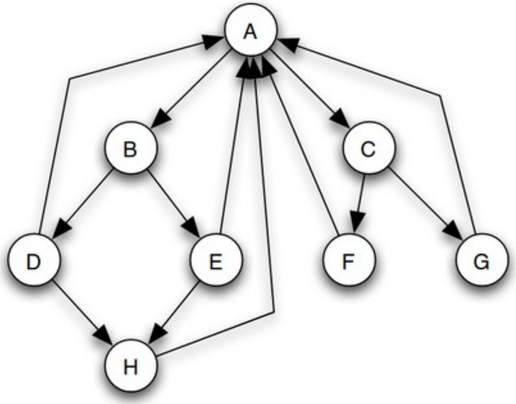
… We compare **PageRank** to an idealized random Web surfer. We show how to efficiently compute **PageRank** for large numbers of pages. And, we show how to apply **PageRank** to search …

- Today, PageRank and its variants are probably part of most ranking systems in linked collections of data
- Relevance = links + content + interactions + …

Simplified PageRank

(Simplified) PageRank

- All nodes start with score $1/N$
- Repeat t times:
  - Divide equally and "send" its score to out-links
  - Add received scores



Exercise

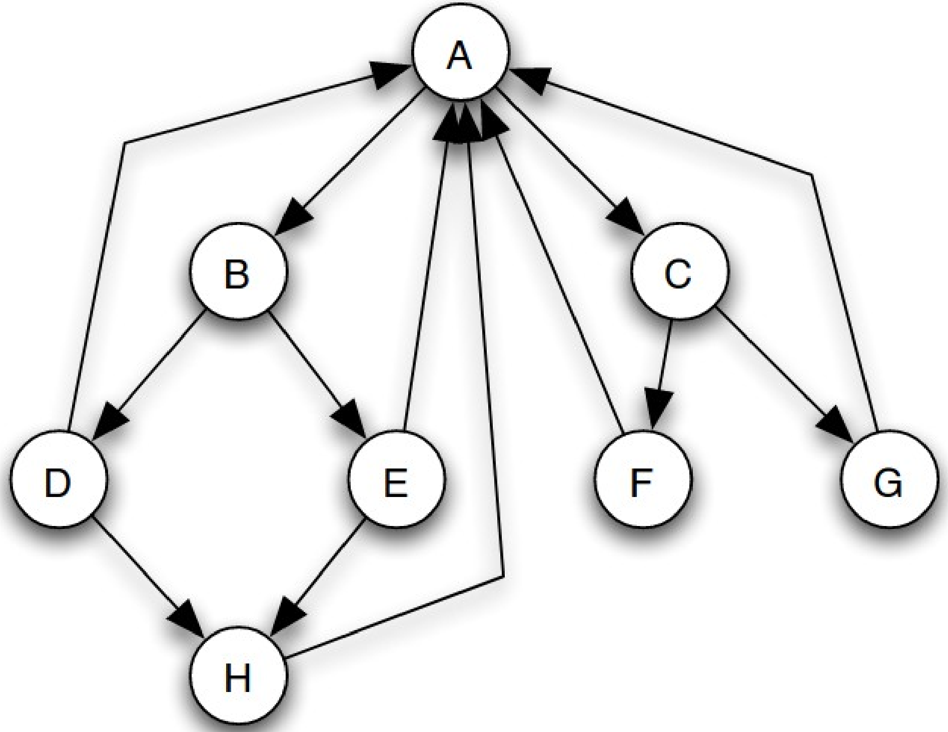
Execute simplified PageRank

All nodes start with score $1/N$

- Repeat t times:
  - Divide equally and "send" the score of each node to out-links
  - Add received scores

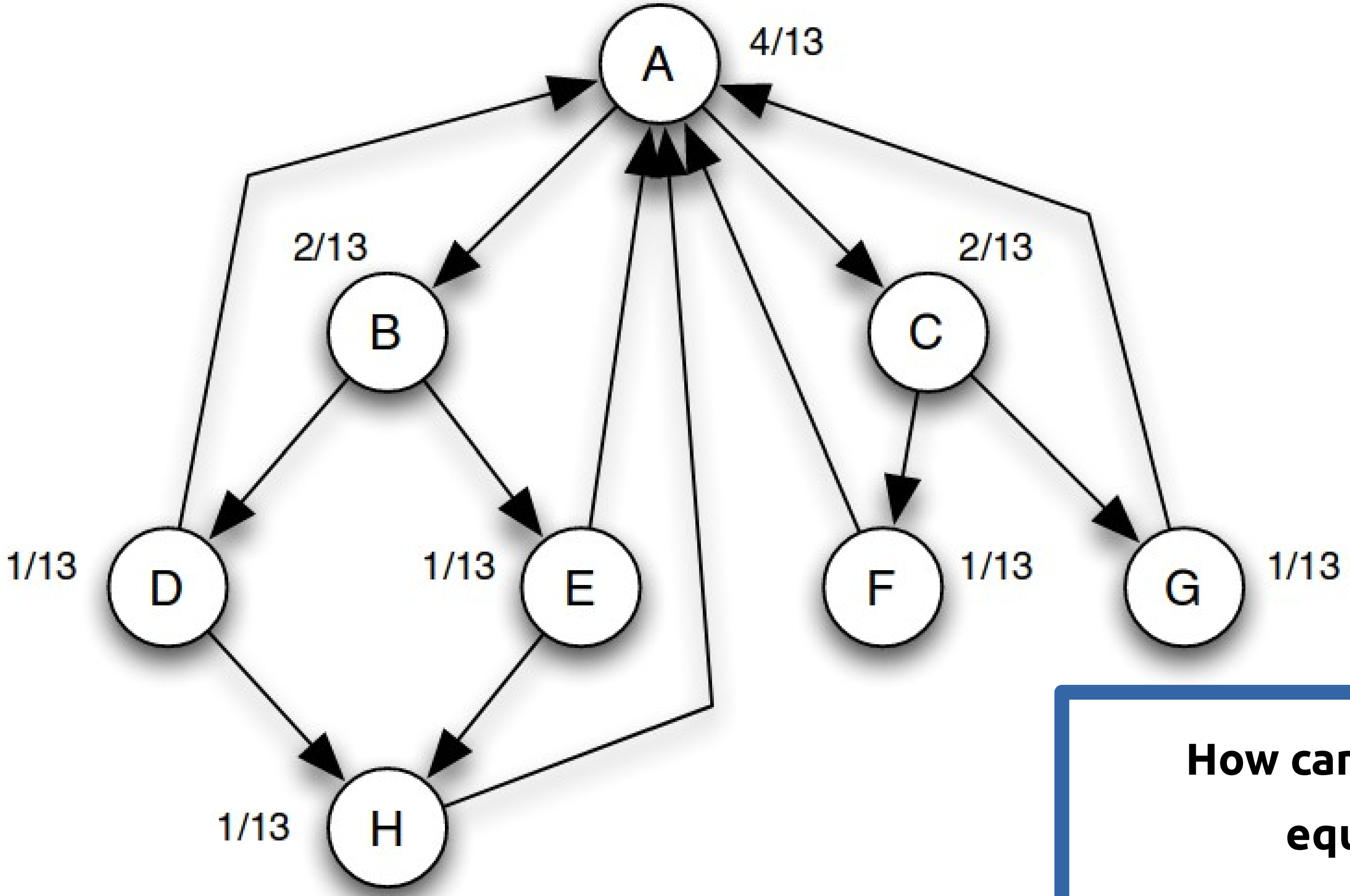
Keep intermediate values in a table

Try to arrive to equilibrium values



Spreadsheet links: https://upfbarcelona.padlet.org/chato/shyq9m6f2g2dh1bw

Equilibrium values



How can you prove these are equilibrium values?

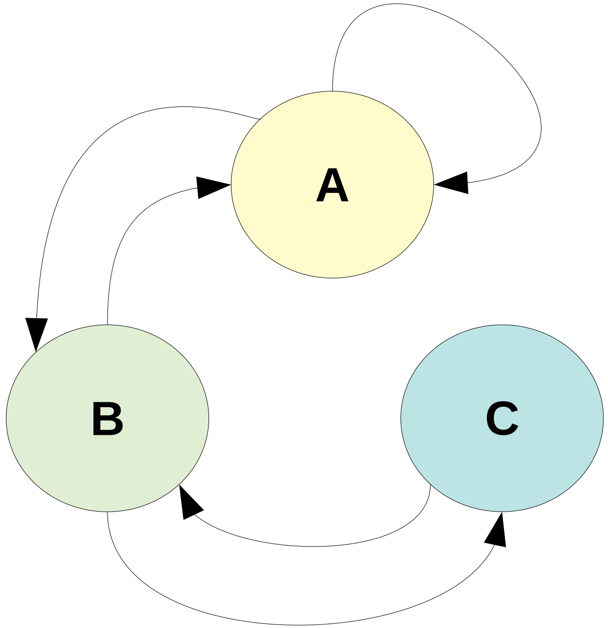
(Simplified) PageRank

$$P_i = c \sum_{j \to i} \frac{P_j}{k_j^{\text{out}}}$$

- k_j^{out} is the number of out-links of page j
- c is a normalization factor to ensure

$$||P||_1 = |P_1| + |P_2| + \cdots + |P_N| = 1$$

Running simplified PageRank on a graph



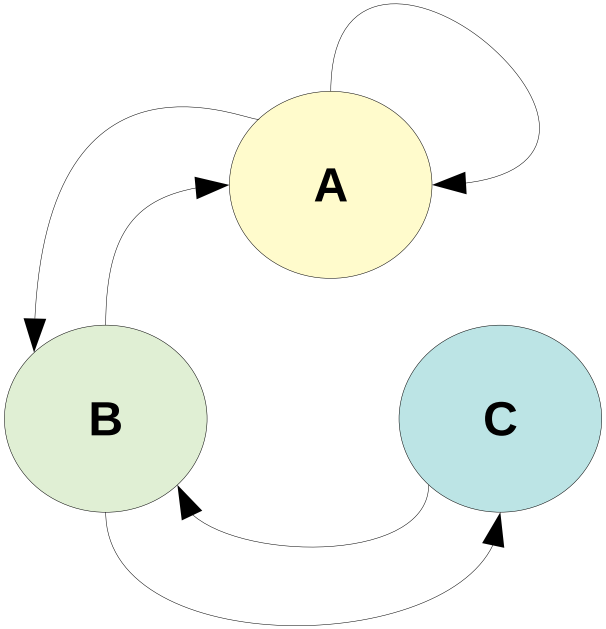
$$M = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Adjacency matrix

$$\hat{M} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \end{bmatrix}$$

Row-stochastic adjacency matrix

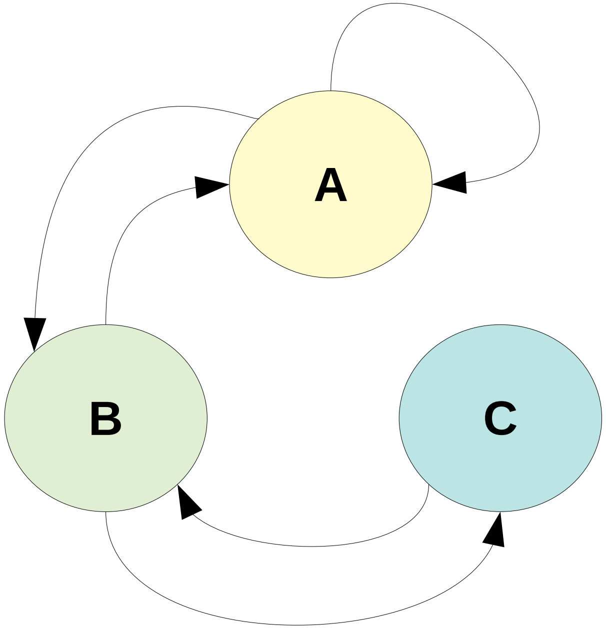
Another example of Simplified PageRank



$$\hat{M}^T = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

First iteration of calculation: $\begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$

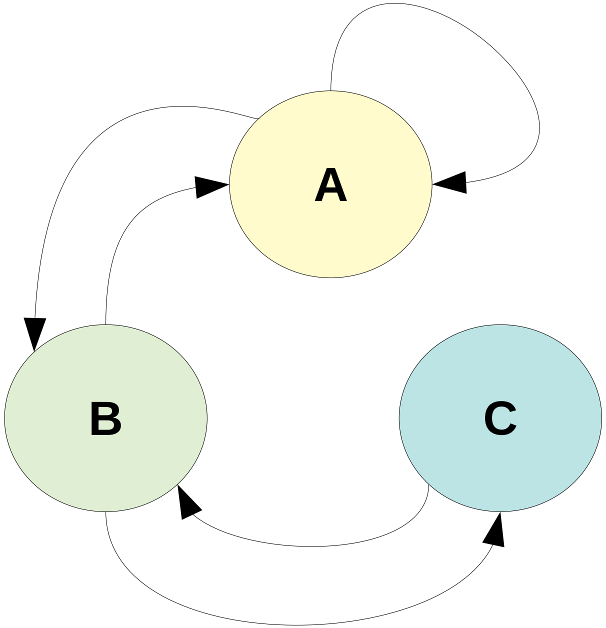
Another example of Simplified PageRank



$$\hat{M}^T = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

Second iteration: $\begin{bmatrix} 5/12 \\ 1/3 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix}$

Another example of Simplified PageRank

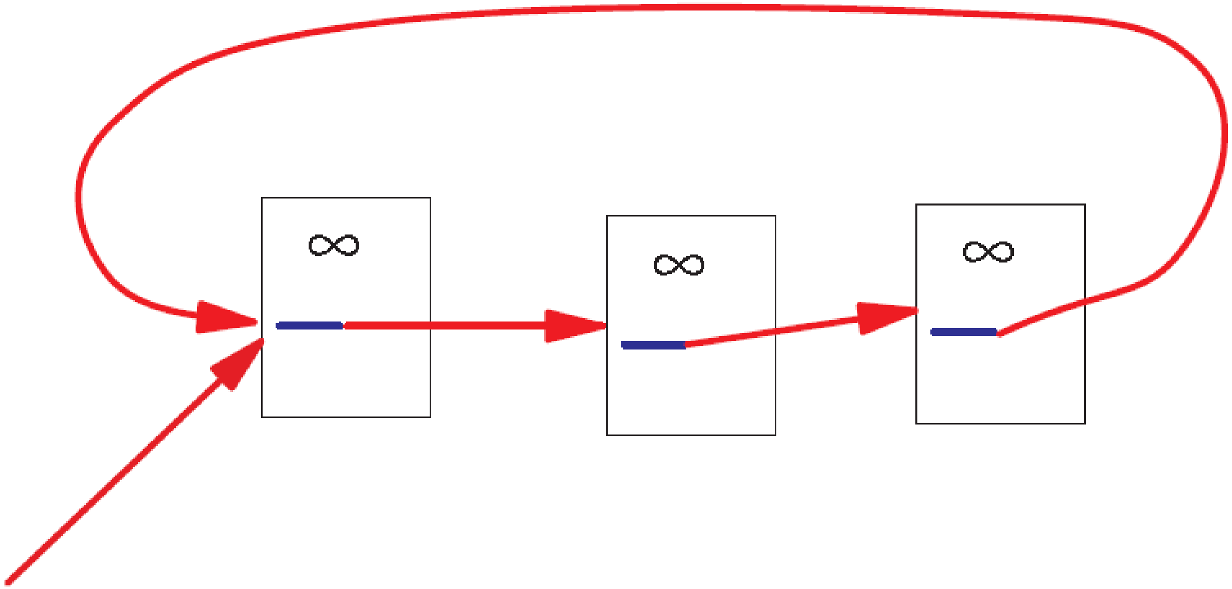


$$\hat{M}^T = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

Following iterations: $\begin{bmatrix} 3/8 \\ 11/24 \\ 1/6 \end{bmatrix} \begin{bmatrix} 5/12 \\ 17/48 \\ 11/48 \end{bmatrix} \ldots \boxed{\begin{bmatrix} 2/5 \\ 2/5 \\ 1/5 \end{bmatrix}}$ Final score

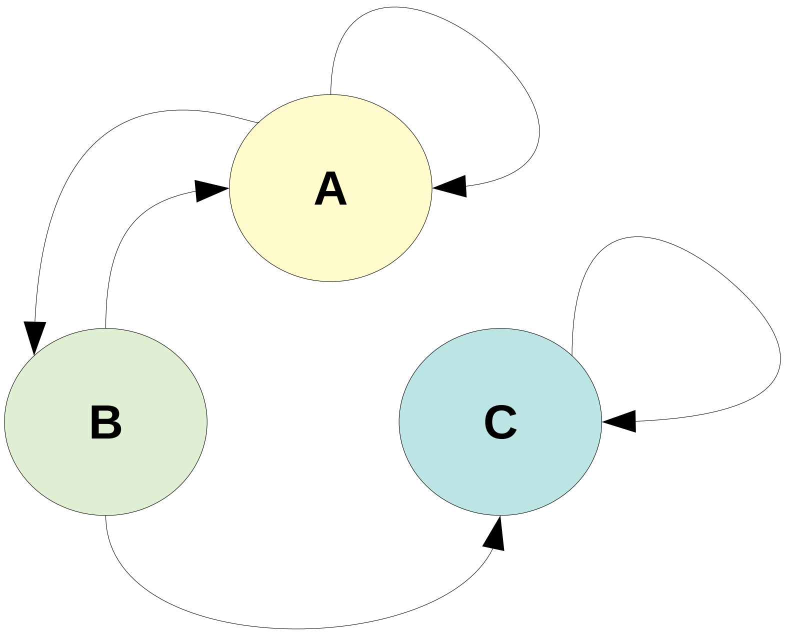
A Problem with Simplified PageRank

A loop:



During each iteration, the loop accumulates score but never distributes score to other pages!

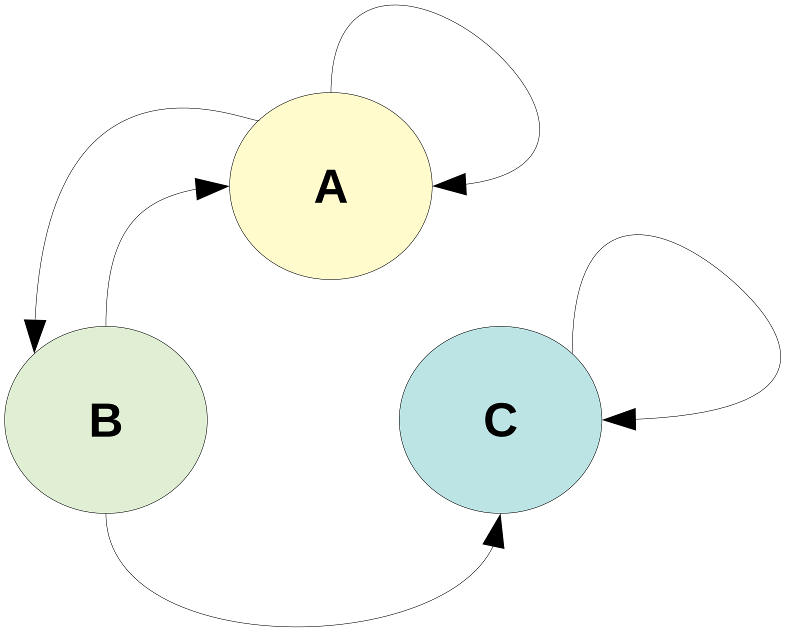
Example of the problem …



$$\hat{M}^T = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

First iteration of calculation: $\begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$

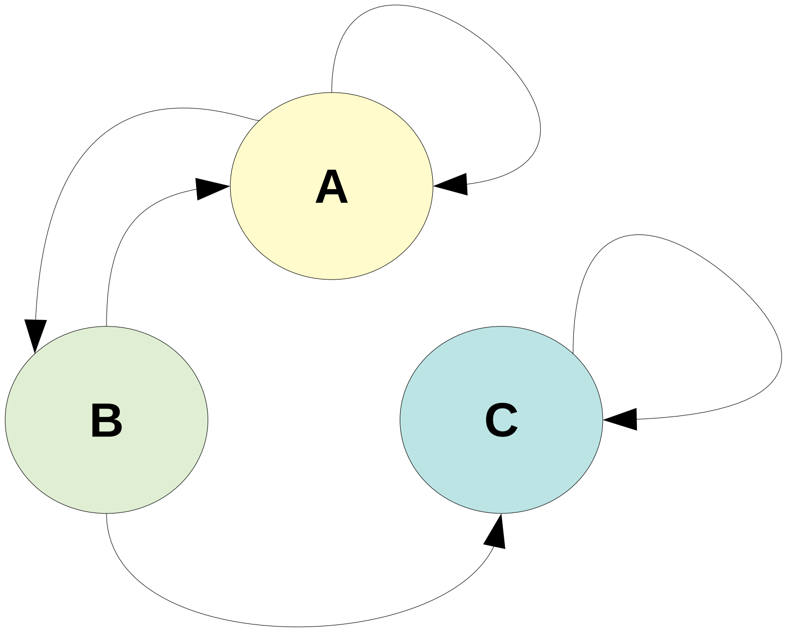
Example of the problem ...



$$\hat{M}^T = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

Second iteration: $\begin{bmatrix} 1/4 \\ 1/6 \\ 7/12 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix}$

Example of the problem ...



$$\hat{M}^T = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

Following iterations: $\begin{bmatrix} 5/24 \\ 1/8 \\ 2/3 \end{bmatrix} \begin{bmatrix} 1/6 \\ 5/48 \\ 35/48 \end{bmatrix} \cdots\cdots \boxed{\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}$ The winner takes all!

Why is PageRank also refered to as “Eigen” centrality

What are we computing?

$$
\begin{aligned}
p^t &= Ap^{t-1} \\
\text{after convergence}: \; p &= Ap
\end{aligned}
$$

A is the transposed row-stochastic adjacency matrix

What is p?

How do you call this method to compute p?

What are we computing?

$$\begin{aligned} p^t &= Ap^{t-1} \\ \text{after convergence}: \; p &= Ap \end{aligned}$$

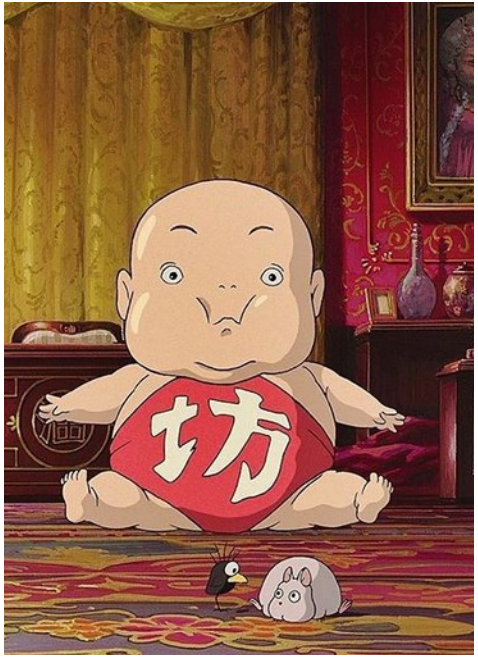
- This will converge if A is:
  - Left-stochastic (each column adds up to one)
  - Irreducible (represents a strongly connected graph)
  - Aperiodic (does not represent a bipartite graph)

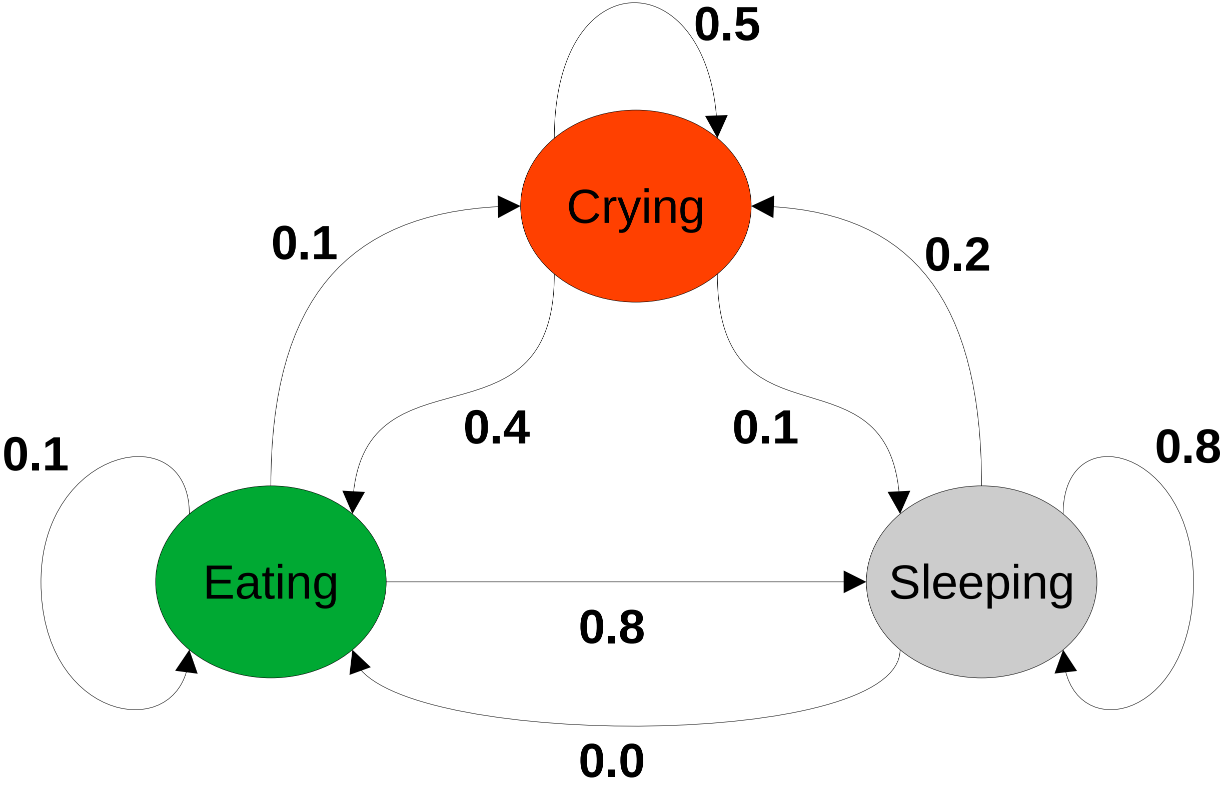
“Random walk” interpretation

Markov Chain

- Discrete process over a set of **states**
- Next state computed from **current state** only (no memory of older states)
  - Higher-order Markov chains can be defined
- Stationary distribution of Markov chain is a probability distribution such that $p = Ap$
- Intuitively, p represents "the average time spent" at each node if the process continues forever

Example Markov Chain: a baby (think of 1-hour time steps)





Random Walks in Graphs

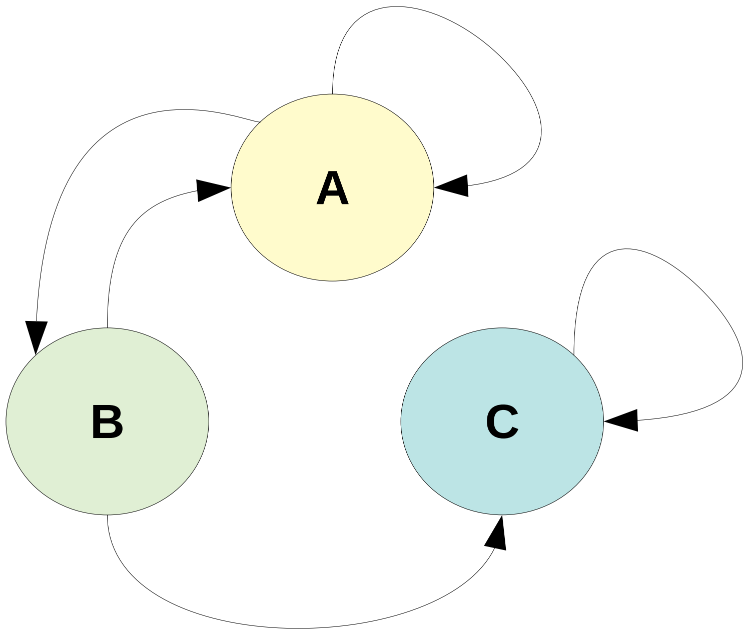
- **Random Surfer Model** $\rightarrow$ Simplified PageRank
  - The simplified model: the standing probability distribution of a random walk on the graph of the web. simply keeps clicking successive links at random
- **Modified Random Surfer** $\rightarrow$ PageRank
  - The modified model: the “random surfer” simply keeps clicking successive links at random, but periodically “gets bored” and jumps to a random page based on a distribution R (e.g., uniform)
  - This guarantees **irreducibility**
  - Pages without out-links (dangling nodes) are a row of zeros, can be replaced by R, or by a row of $1/N$

PageRank

$$P_i = \alpha \sum_{j \to i} \frac{P_j}{k_j^{\mathrm{out}}} + (1-\alpha) R(i)$$

R(i): web pages that "users" jump to when they "get bored";
Uniform preferences => *R(i) = 1/N*

An example of PageRank $\alpha = 0.8$



$$\hat{M}^T = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\alpha \hat{M}^T + (1-\alpha)R = 0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 0.333 \\ 0.333 \\ 0.333 \end{bmatrix} \begin{bmatrix} 0.333 \\ 0.200 \\ 0.467 \end{bmatrix} \begin{bmatrix} 0.280 \\ 0.200 \\ 0.520 \end{bmatrix} \begin{bmatrix} 0.259 \\ 0.179 \\ 0.563 \end{bmatrix} \ldots \boxed{\begin{bmatrix} 7/33 \\ 5/33 \\ 21/33 \end{bmatrix}}$$

Was: $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

Summary

Things to remember

- Simplified PageRank
- PageRank

Practice on your own

- Consider a directed graph $G = (V, E)$ in which $V = \{1, 2, \ldots, N\}$ and $(i, j) \in E \iff i \in V \wedge j \in V \wedge (j = i + 1 \vee j = i = N)$
  - 1. Indicate the value of Simplified PageRank S(i) for each node i in the graph, justifying your answer.
  - 2. Indicate the value of PageRank P (i) for each node i in the graph as a function of i and the parameter α.
- Tip: write P(1), then write P(2), then write P(3), then write P(i).