CS247: ADVANCED DATA MINING

07: Graph and Network: Random Walk

Instructor: Yizhou Sun

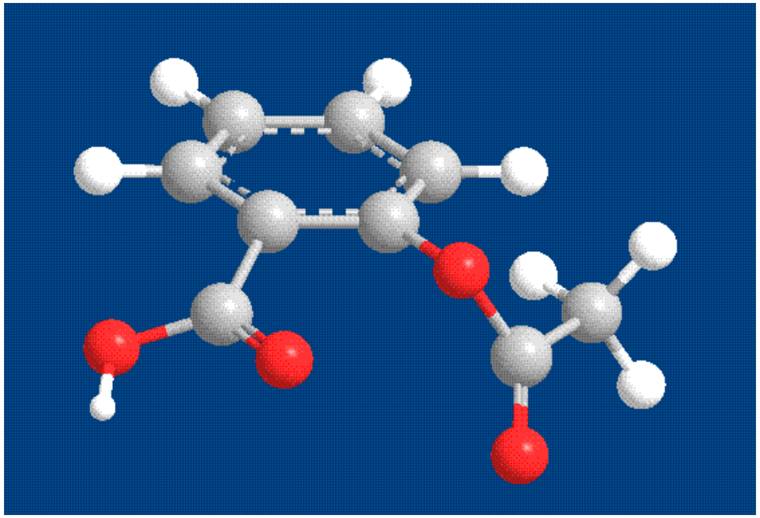
yzsun@cs.ucla.edu

February 7, 2022

Random Walk on Graphs

- Introduction to Graph/Network Data
- Random Walk on Graphs
- PageRank and Personalized PageRank
- Summary

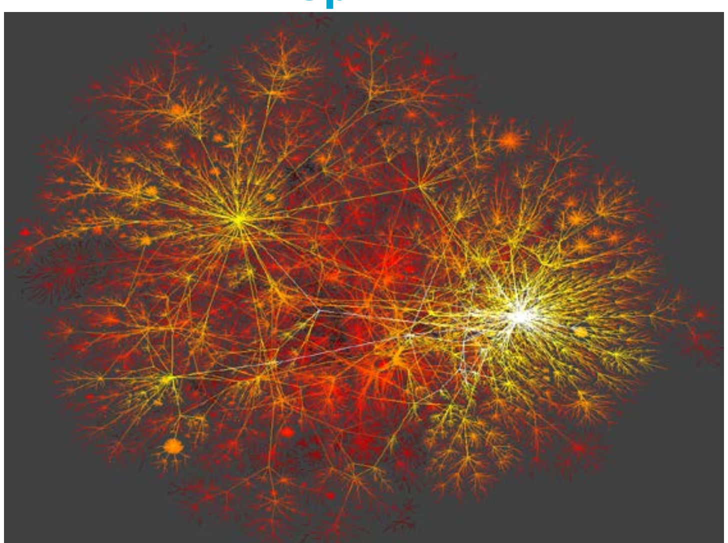
Graph, Graph, Everywhere



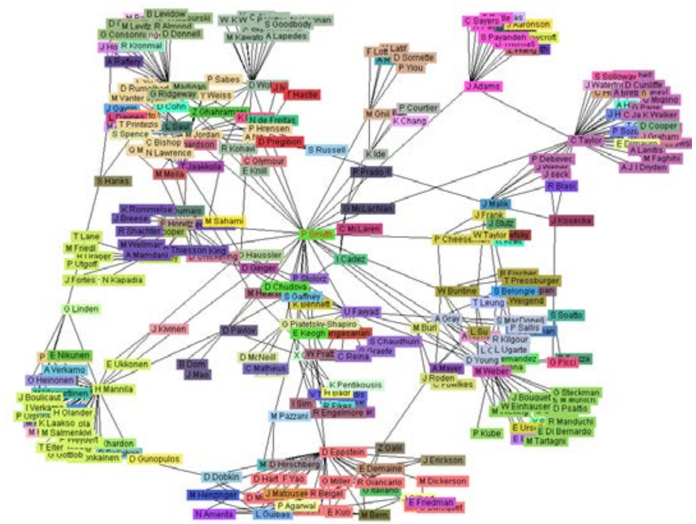
Aspirin

from H. Jeong et al Nature 411, 41 (2001)

Yeast protein interaction network



Internet



Co-author network

Why Graph Mining?

- Graphs are ubiquitous
  - Chemical compounds (Cheminformatics)
  - Protein structures, biological pathways/networks (Bioinformactics)
  - Program control flow, traffic flow, and workflow analysis
  - XML databases, Web, and social network analysis
- Graph is a general model
  - Trees, lattices, sequences, and items are degenerated graphs
- Diversity of graphs
  - Directed vs. undirected, labeled vs. unlabeled (edges & vertices), weighted vs. unweighted, homogeneous vs. heterogeneous
- Complexity of algorithms: many problems are of high complexity

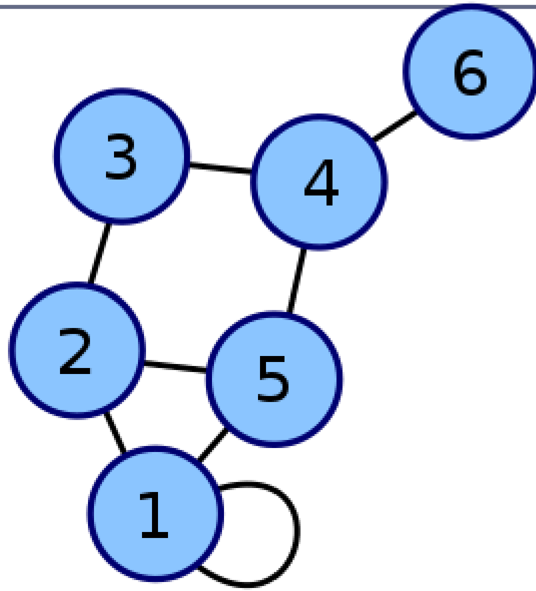
Graph Tasks

- Node Level
  - Node classification
  - Node clustering
  - Link prediction
  - Ranking and similarity search
  - …
- Graph Level
  - Graph classification
  - Graph clustering
  - Graph similarity search
  - …

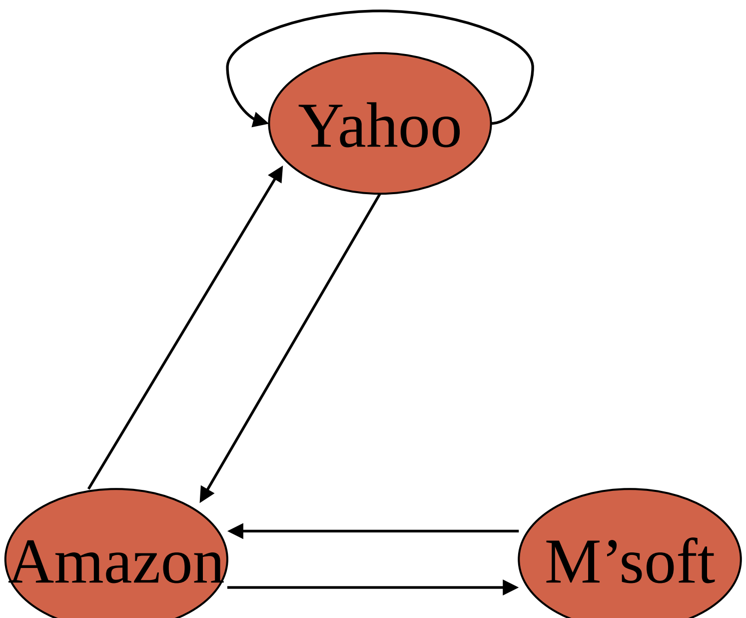
Representation of a Graph

- $G =< V, E >$
  - $V = \{u_1, \dots, u_N\}$: node set
  - $E \subseteq V \times V$: edge set
- Adjacency matrix
  - $A = \{a_{ij}\}, i, j = 1, \dots, N$
    - $a_{ij} = 1, if\ < u_i, u_j > \in E$
    - $a_{ij} = 0, if\ < u_i, u_j > \notin E$
  - Undirected graph vs. Directed graph
    - $A = A^{\mathrm{T}}\ vs. A \neq A^{\mathrm{T}}$
  - Weighted graph
    - Use *W* instead of *A*, where w_{ij} represents the weight of edge $< u_i, u_j >$

Examples



$$\begin{pmatrix} 2 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$



	y	a	m
y	1	1	0
a	1	0	1
m	0	1	0

Adjacency matrix A

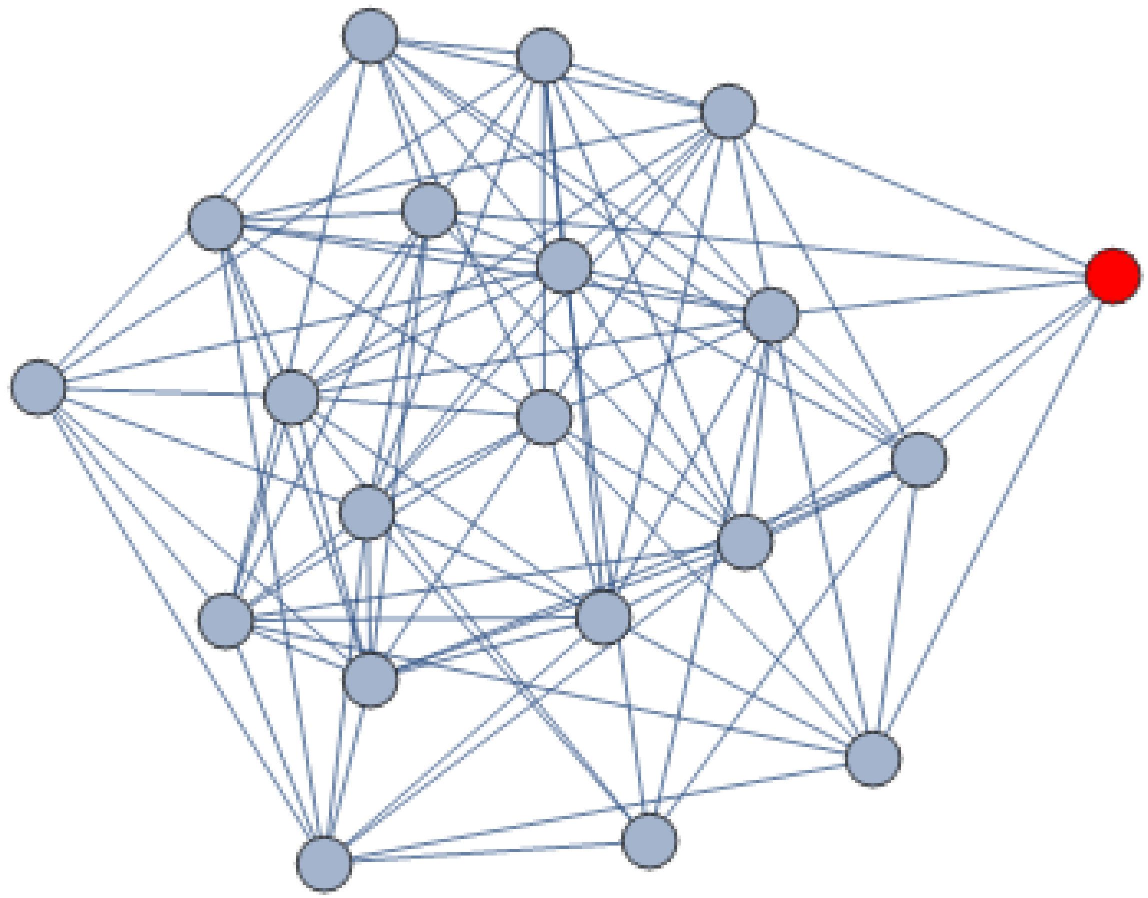
Neighbors and Degree

- Neighbor set of node i
  - $N(i) = \{j | a_{ij} > 0\ and\ j \in V\}$
- Degree of node i
  - $d_i = \sum_j a_{ij}$

Random Walk on Graphs

- Introduction to Graph/Network Data
- Random Walk on Graphs
- PageRank and Personalized PageRank
- Summary

Random Walk Example



https://mathematica.stackexchange.com/questions/156626/generate-random-walk-on-a-graph

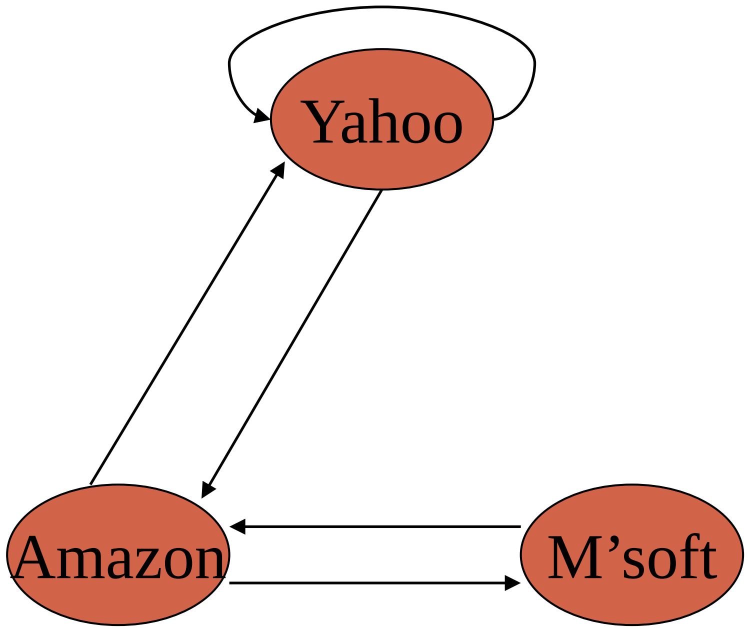
Connection to Stochastic Process

- Finite State Discrete Time Stochastic Process
  - $\{X_n; n \geq 0\} = \{X_0, X_1, X_2, \dots,\}$
  - n: discrete time step; $X_n \in \{1,2, \dots M\}$, i.e., discrete states
- Finite-State Markov Chain
  - *The current depends only on the most recent*
    - $\Pr(X_n = j | X_{n-1} = i, X_{n-1} = k, \dots, X_0 = m) = \Pr(X_n = j | X_{n-1} = i)$
- Transition Probability
  - Jump from state i to state j: $P_{ij} := \Pr(X_n = j | X_{n-1} = i)$

In the graph setting

- States: nodes
- Transition probability:
  - $P_{ij} = a_{ij}/d_i$
- A random walk with length l:
  - $\{X_0, X_1, X_2, \dots, X_{l-1}\}$
    - *Where* $X_n \in \{1,2,\dots,N\}$ and $\Pr(X_n = j | X_{n-1} = i) = P_{ij}$ *defined above*

Example



	y	a	m
y	1	1	0
a	1	0	1
m	0	1	0

Adjacency matrix A

	y	a	m
y	$\frac{1}{2}$	$\frac{1}{2}$	0
a	$\frac{1}{2}$	0	$\frac{1}{2}$
m	0	1	0

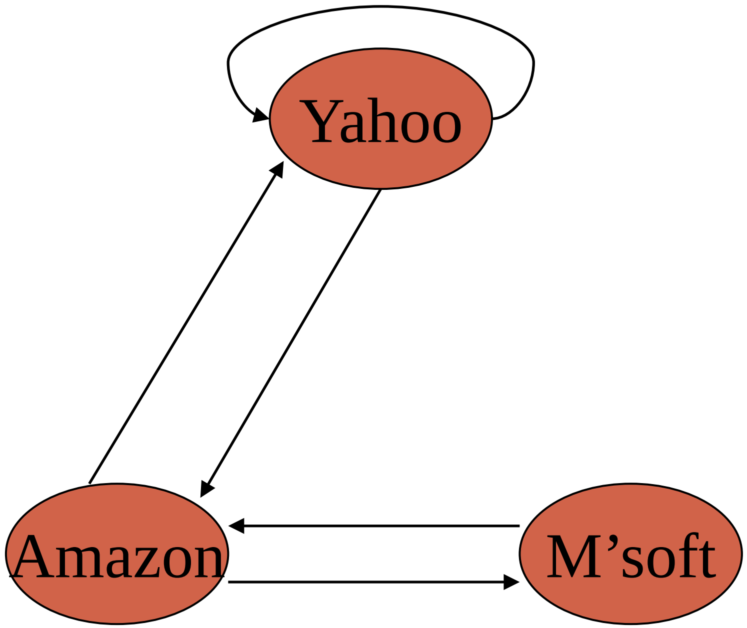
Transition matrix P

Stationary Distribution

- $\pi = (\pi_1, \pi_2, \ldots, \pi_N)$ $is\ called\ a\ stationary\ distribution\ of\ a\ markov\ chain\ with\ transition\ matrix\ P$, if
  - $\pi_i \geq 0\ and\ \sum_i \pi_i = 1$
  - $\pi P = \pi$

Any aperiodic and irreducible finite Markov chain has precisely one stationary distribution.

Example

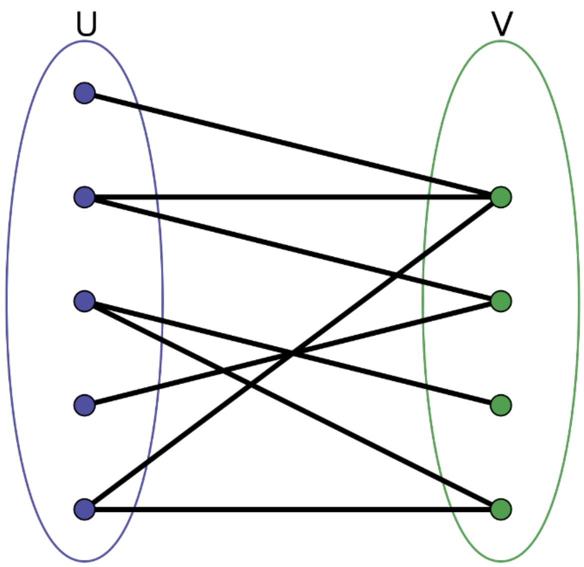


$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{pmatrix}$$

$$\pi = (\frac{2}{5}, \frac{2}{5}, \frac{1}{5})$$

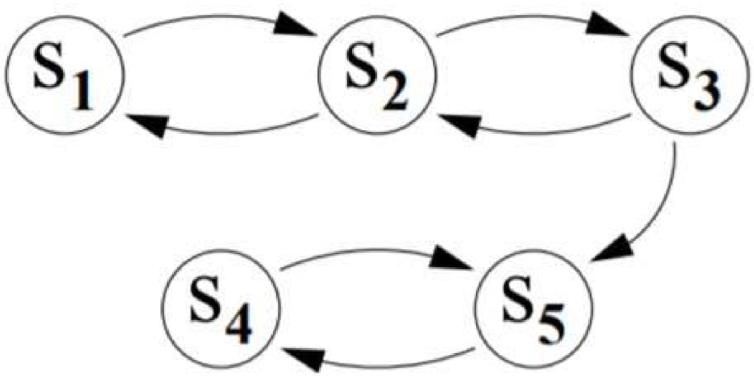
Example on Periodic Graph

- Periodic, e.g., Bipartite graph

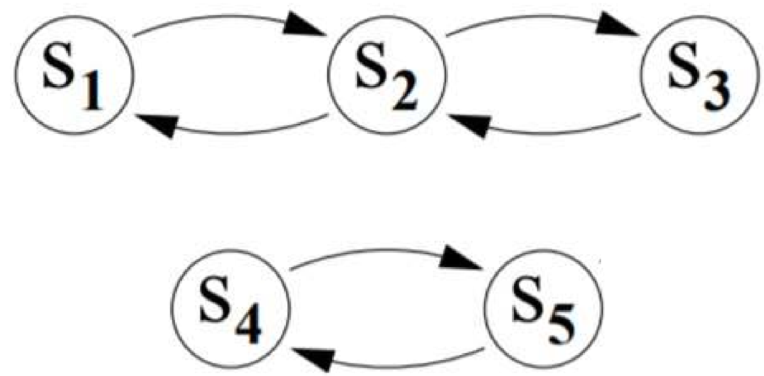


Examples on Reducible Graphs

- Irreducible: any nodes can reach to any nodes
- Reducible: Not irreducible



Not Irreducible

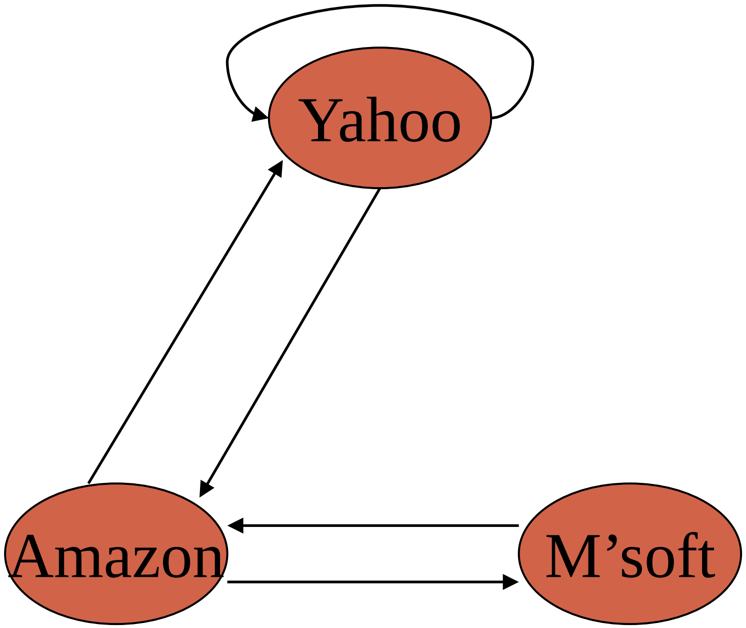


Not Irreducible

Stationary Distribution Calculation

- Method 1: $\pi P = \pi$
  - Eigen-decomposition
  - π is the first or principal eigenvector, with corresponding eigenvalue 1
- Method 2: Power's iteration
  - $\pi^{(t+1)} \leftarrow \pi^{(t)} P$

Power Iteration Example



$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 1 & 0 \end{pmatrix}$$

$$\begin{matrix} y \\ a \\ m \end{matrix} = \begin{matrix} 1/3 \\ 1/3 \\ 1/3 \end{matrix} \quad \begin{matrix} 1/3 \\ 1/2 \\ 1/6 \end{matrix} \quad \begin{matrix} 5/12 \\ 1/3 \\ 1/4 \end{matrix} \quad \begin{matrix} 3/8 \\ 11/24 \\ 1/6 \end{matrix} \quad \cdots \quad \begin{matrix} 2/5 \\ 2/5 \\ 1/5 \end{matrix}$$

$$\pi^{(0)\prime} \quad \pi^{(1)\prime} \quad \pi^{(2)\prime} \quad \pi^{(t)\prime} \quad \pi^{*\prime}$$

Random Walk on Graphs

- Introduction to Graph/Network Data
- Random Walk on Graphs
- PageRank and Personalized PageRank
- Summary

The History of PageRank

- PageRank was developed by Larry Page (hence the name *Page*-Rank) and Sergey Brin.
- It is first as part of a research project about a new kind of search engine. That project started in 1995 and led to a functional prototype in 1998.
- Shortly after, Page and Brin founded Google.

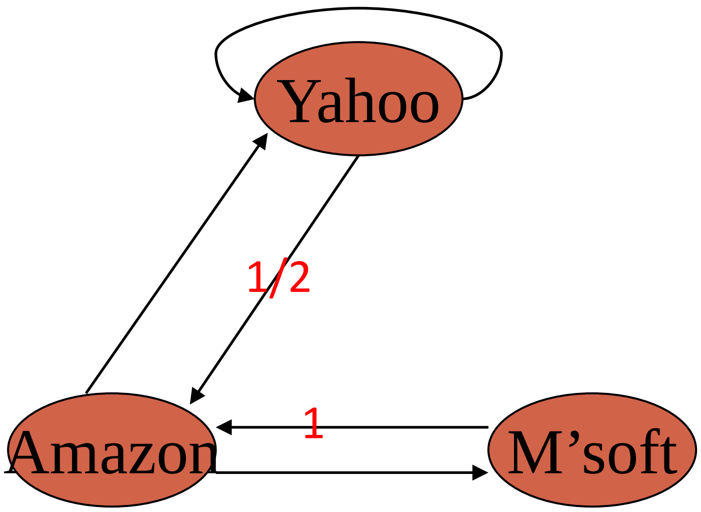
Ranking web pages

- Web pages are not equally "important"
  - www.cnn.com vs. a personal webpage
- Inlinks as votes
  - The more inlinks, the more important
- Are all inlinks equal?
  - Higher ranked inlink should play a more important role
  - Recursive question!

Simple recursive formulation

- Each link's vote is proportional to the importance of its source page
- If page P with importance x has n outlinks, each link gets x/n votes
- Page P's own importance is the sum of the votes on its inlinks

$$y = y/2 + a/2$$
$$a = y/2 + m$$
$$m = a/2$$



Connection to Stationary Distribution

- Given the web network adjacency matrix A, Compute transition matrix P
  - $P = D^{-1}A$, where D is the diagonal matrix and $D_{ii} = d_i$
    - PageRank vector corresponds to the stationary distribution of P
  - $M = P^T$: column stochastic matrix
    - We usually use $\boldsymbol{r} = M\boldsymbol{r}$ to denote the PageRank vector

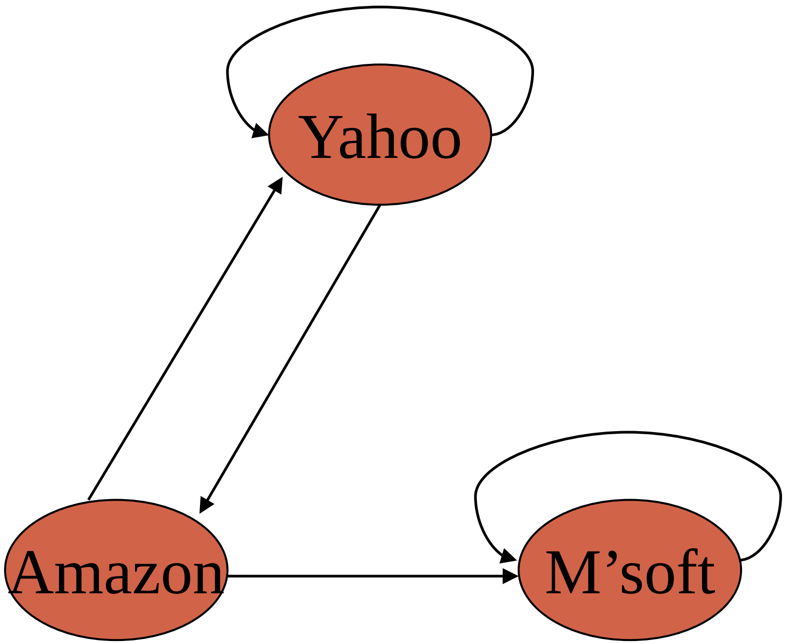
Corner cases

- What if the transition matrix is not aperiodic and not irreducible?

Spider traps

- A group of pages is a spider trap if there are no links from within the group to outside the group
  - Random surfer gets trapped!

Microsoft becomes a spider trap



	y	a	m
y	1/2	1/2	0
a	1/2	0	0
m	0	1/2	1

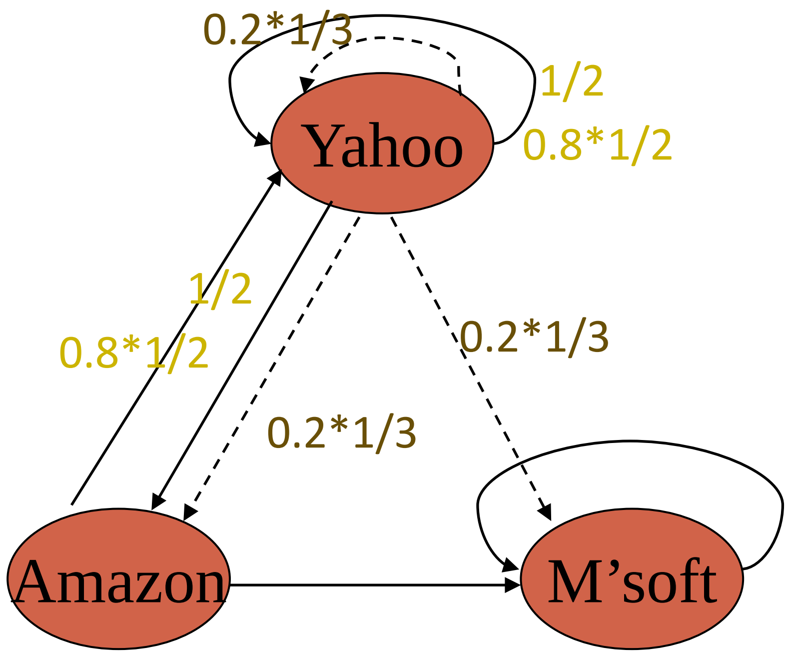
M=Transpose of P

y		1/3	1/3	1/4	5/24		0
a	=	1/3	1/6	1/6	1/8	. . .	0
m		1/3	1/2	7/12	2/3		1

Random teleports

- The Google solution for spider traps
- At each time step, the random surfer has two options:
  - With probability β, follow a link at random
  - With probability $1-\beta$, jump to some page uniformly at random
  - Common values for β are in the range 0.8 to 0.9
- Surfer will teleport out of spider trap within a few time steps

Random teleports ($\beta = 0.8$)



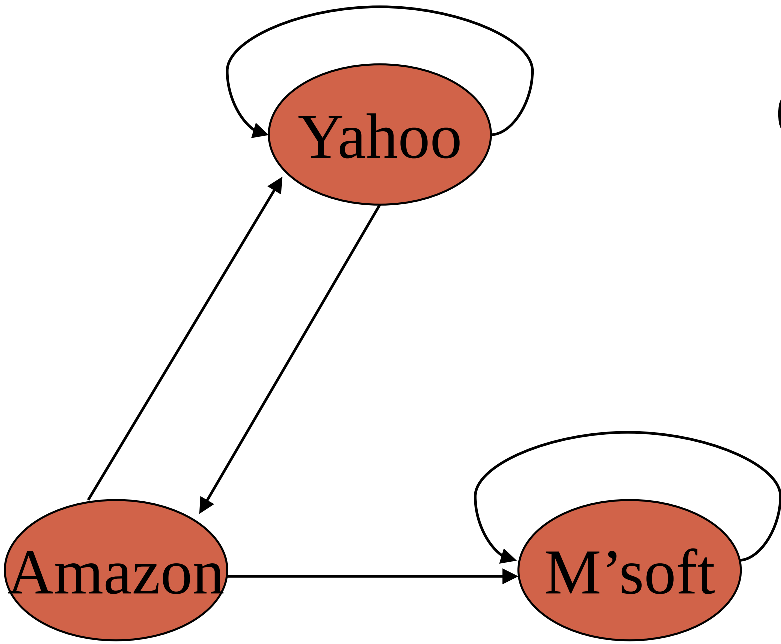
$$\begin{array}{c} \\ y \\ a \\ m \end{array} \begin{array}{c} y \\ \boxed{\begin{array}{c} 1/2 \\ 1/2 \\ 0 \end{array}} \end{array} \qquad 0.8* \begin{array}{c} y \\ \boxed{\begin{array}{c} 1/2 \\ 1/2 \\ 0 \end{array}} \end{array} \quad + 0.2* \begin{array}{c} y \\ \boxed{\begin{array}{c} 1/3 \\ 1/3 \\ 1/3 \end{array}} \end{array}$$

$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

----> : teleport links from "Yahoo"

$$\begin{array}{c} y \\ a \\ m \end{array} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

Random teleports ($\beta = 0.8$)



$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

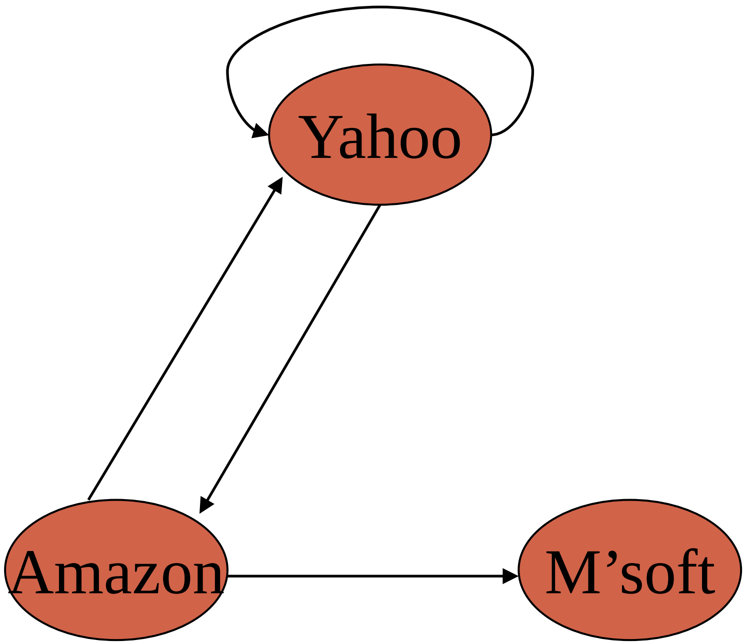
$$\begin{matrix} y \\ a \\ m \end{matrix} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

$$\begin{matrix} y \\ a \\ m \end{matrix} = \begin{bmatrix} 0.333 \\ 0.333 \\ 0.333 \end{bmatrix} \begin{bmatrix} 0.333 \\ 0.200 \\ 0.467 \end{bmatrix} \begin{bmatrix} 0.280 \\ 0.200 \\ 0.520 \end{bmatrix} \begin{bmatrix} 0.259 \\ 0.179 \\ 0.563 \end{bmatrix} \ldots \begin{bmatrix} 7/33 \\ 5/33 \\ 21/33 \end{bmatrix}$$

Dead ends

- Pages with no outlinks are "dead ends" for the random surfer
  - Nowhere to go on next step

Microsoft becomes a dead end



$$0.8\begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \end{bmatrix} + 0.2\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

$$\begin{matrix} y \\ a \\ m \end{matrix}\begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 1/15 \end{bmatrix}$$

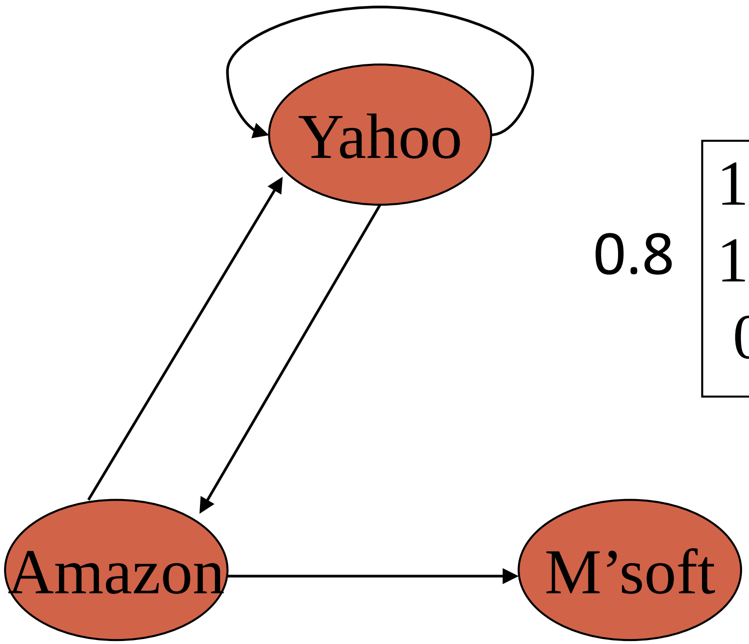
Non-stochastic!

$$\begin{matrix} y \\ a \\ m \end{matrix} = \begin{matrix} 1/3 \\ 1/3 \\ 1/3 \end{matrix} \quad \begin{matrix} 1/3 \\ 0.2 \\ 0.2 \end{matrix} \quad \ldots \quad \begin{matrix} 0 \\ 0 \\ 0 \end{matrix}$$

Dealing with dead-ends

- Method 1: Teleport
  - Follow random teleport links with probability 1.0 from dead-ends
  - Adjust matrix accordingly
- Method 2: Prune and propagate
  - Preprocess the graph to eliminate dead-ends
    - Might require multiple passes
  - Compute pagerank on reduced graph
  - Approximate values for deadends by propagating values from reduced graph

Dealing dead end: teleport

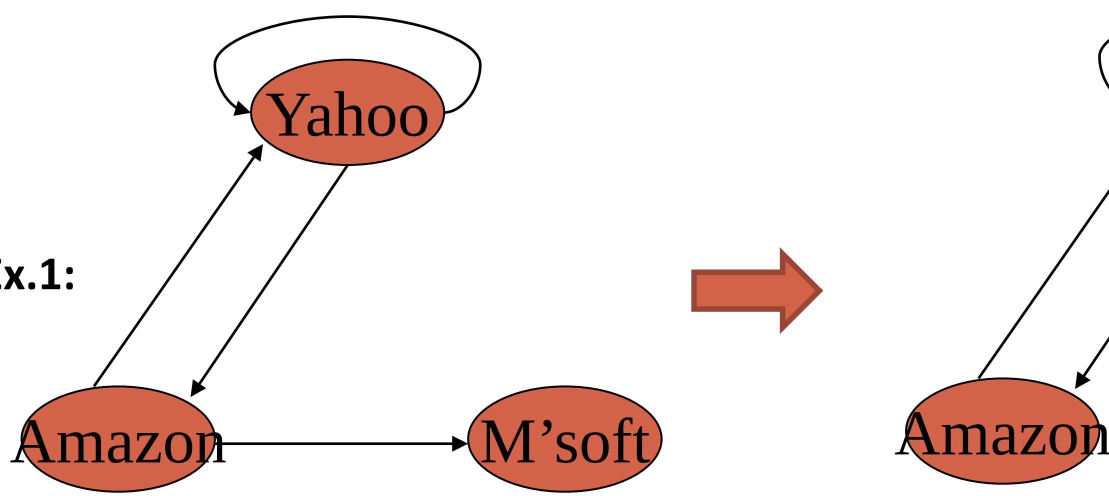


$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \end{bmatrix} + \begin{bmatrix} 0.2*1/3 & 0.2*1/3 & 1*1/3 \\ 0.2*1/3 & 0.2*1/3 & 1*1/3 \\ 0.2*1/3 & 0.2*1/3 & 1*1/3 \end{bmatrix}$$

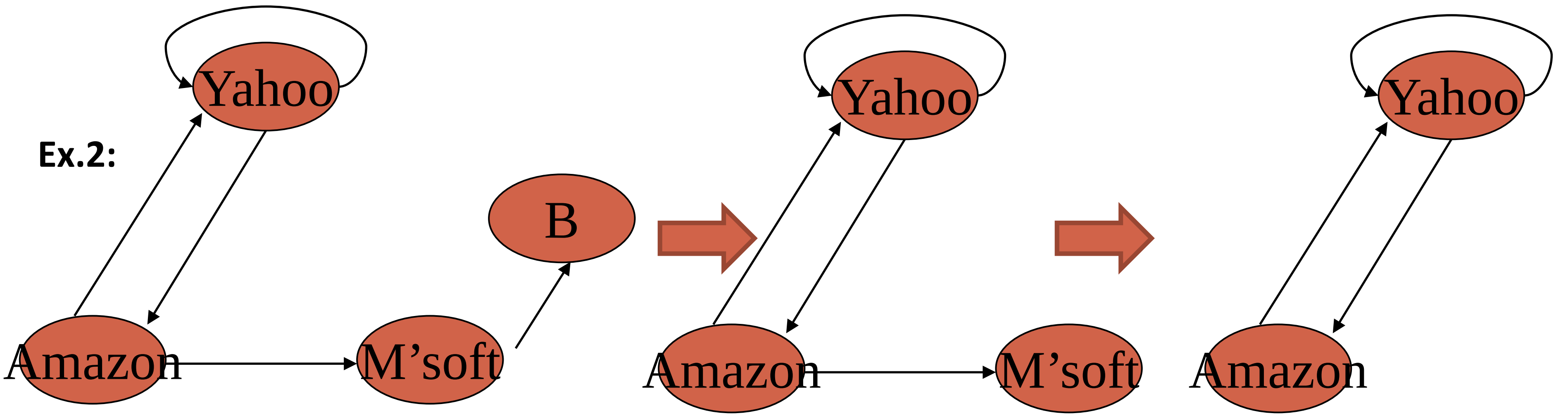
$$\begin{matrix} y \\ a \\ m \end{matrix} \begin{bmatrix} 7/15 & 7/15 & 1/3 \\ 7/15 & 1/15 & 1/3 \\ 1/15 & 7/15 & 1/3 \end{bmatrix}$$

Dealing dead end: reduce graph

Ex.1:



Ex.2:



PageRank

- Construct the N-by-N matrix **B** as follows
  - $B_{ij} = \beta M_{ij} + (1-\beta)/N$
- Verify that **B** is a stochastic matrix
- The page rank vector **r** is the principal eigenvector of this matrix
  - satisfying $\mathbf{r} = \mathbf{B}\mathbf{r}$
- Equivalently, **r** is the stationary distribution of the random walk with teleports

Computing PageRank

- Key step is matrix-vector multiplication
  - $\mathbf{r}^{\text{new}} = \mathbf{B}\mathbf{r}^{\text{old}}$
- Easy if we have enough main memory to hold $\mathbf{B}$, $\mathbf{r}^{\text{old}}$, $\mathbf{r}^{\text{new}}$
- Say N = 1 billion pages
  - We need 4 bytes for each entry (say)
  - 2 billion entries for vectors, approx 8GB
  - Matrix B has N^2 entries
    - 10^{18} is a large number!

Rearranging the equation

$\mathbf{r} = \mathbf{B}\mathbf{r}$, where

$B_{ij} = \beta M_{ij} + (1-\beta)/N$

$r_i = \sum_{1 \le j \le N} B_{ij}\, r_j$

$r_i = \sum_{1 \le j \le N} [\beta M_{ij} + (1-\beta)/N]\, r_j$

$\quad = \beta \sum_{1 \le j \le N} M_{ij}\, r_j + (1-\beta)/N \sum_{1 \le j \le N} r_j$

$\quad = \beta \sum_{1 \le j \le N} M_{ij}\, r_j + (1-\beta)/N$, since $|\mathbf{r}| = 1$

$\mathbf{r} = \beta \mathbf{M}\mathbf{r} + [(1-\beta)/N]_N$

where $[x]_N$ is an N-vector with all entries x

Personalized PageRank

- Query-dependent Ranking
  - For a query webpage u, which webpages are most important to u?
    - We need a measure s(u,v)
  - The relative important webpages to different queries would be different

Calculation of P-PageRank

- Recall PageRank calculation:
  - $\mathbf{r} = \beta\mathbf{M}\mathbf{r} + [(1-\beta)/N]_N$ or
  - $\mathbf{r} = \beta\mathbf{M}\mathbf{r} + (1-\beta)\, r_0$, where $r_0 = \begin{pmatrix} 1/N \\ 1/N \\ \dots \\ 1/N \end{pmatrix}$
- For P-PageRank, s(u,v) $= \mathbf{r}_u(v)$, where $\mathbf{r}_u = \beta\mathbf{M}\,\mathbf{r}_u + (1-\beta)\, r_0$

  by replacing r_0 with $r_0 = \begin{pmatrix} 0 \\ 0 \\ \dots \\ 1 \\ \dots \\ 0 \end{pmatrix}$ ← uth webpage

  - Teleport to webpage u

Random Walk on Graphs

- Introduction to Graph/Network Data
- Random Walk on Graphs
- PageRank and Personalized PageRank
- Summary

Summary

- Random walk on graph
  - PageRank for ranking
  - P-PageRank for similarity search