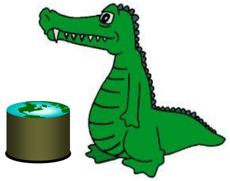
Logistics

- Lab 3 due Friday 11:55pm
  - TA office hour Thursday, possibly additional OH on Friday
- Sample midterm 1 questions
  - Review Lecture Monday
  - Solutions will be discussed and posted then
- Midterm 1 in class on Wed. (10/14)
  - Similar question topics/types as in sample
  - 50 min, close book, paper based, no aids

Midterm 1 topics

1. Data modeling and similarity metrics
   – Vectors, sets, strings, graphs
2. NLP techniques
3. Map-Reduce API
4. Exploratory Analysis using visualizations



Review

- 3 more examples of M/R applications
- Web as a graph
  - Ways to search and rank the Web
- Link Analysis and Page Rank
  - Computing importance scores for each web page based on link structures
- Mathematics abstraction of the problem with flow equations and solutions
  - Gaussian elimination not scalable
- Matrix formulation of the problem

Rank Vector r = Eigenvector of M

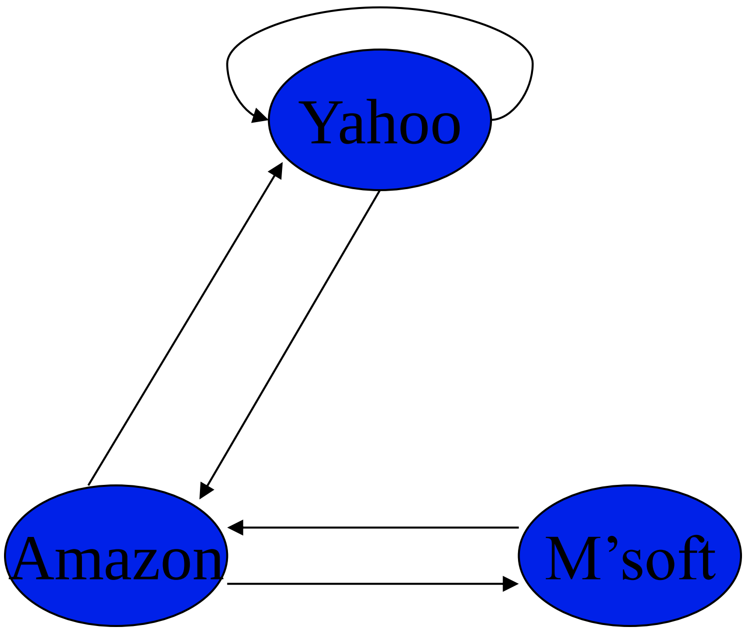
- The flow equations can be written

$$\mathbf{r} = \mathbf{M}\mathbf{r}$$

- The rank vector **r** is an eigenvector of the stochastic web matrix **M**
  - with corresponding eigenvalue 1

- We can now efficiently solve for **r**!
  - The method is called Power iteration

Power Iteration Example



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

$$\begin{array}{c|ccc|}
 & y & a & m \\
\hline
y & 1/2 & 1/2 & 0 \\
a & 1/2 & 0 & 1 \\
m & 0 & 1/2 & 0 \\
\hline
\end{array}$$

r = Mr

$$\begin{bmatrix} y \\ a \\ m \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} y \\ a \\ m \end{bmatrix}$$

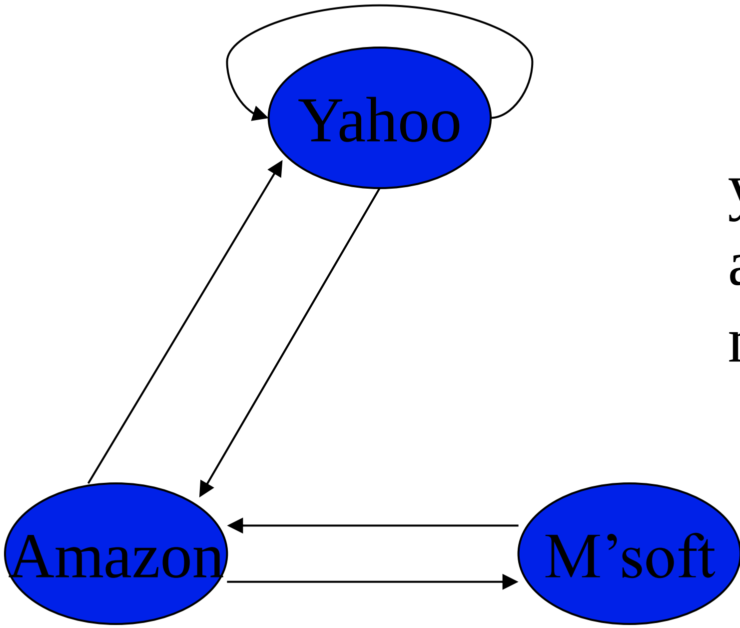
Power Iteration method

- Given a web graph with N nodes, where the nodes are pages and edges are hyperlinks
- Power iteration: a simple iterative scheme
  - Suppose there are N web pages
  - Initialize: $\mathbf{r}^0 = [1/N,....,1/N]^T$
  - Iterate: $\mathbf{r}^{k+1} = \mathbf{M}\mathbf{r}^k$
  - Stop when $|\mathbf{r}^{k+1} - \mathbf{r}^k|_1 < \varepsilon$
    - $|\mathbf{x}|_1 = \sum_{1\leq i\leq N}|x_i|$ is the L_1 norm
    - Can use any other vector norm e.g., Euclidean norm

$$r^{(t+1)}_j = \sum_{i\to j} r^{(t)}_i / d_i$$

d_i *is out-degree of node i*

Power Iteration Example (cont.)



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

Power Iteration:

- Set $r_j = 1/N$
- **1:** $r'_j = \sum_{i \to j} r_i / d_i$
  (i.e., r' = Mr)
- **2:** $r = r'$
- Goto **1**

y		1/3	1/3	5/12	3/8		2/5
a	=	1/3	1/2	1/3	11/24	. . .	2/5
m		1/3	1/6	1/4	1/6		1/5

Random Walk Interpretation

- Imagine a random web surfer
  - At any time t, surfer is on some page P
  - At time t+1, the surfer follows an outlink from P uniformly at random
  - Ends up on some page Q linked from P
  - Process repeats indefinitely
- Let **p**(t) be a vector whose i^{th} component is the probability that the surfer is at page i at time t
  - **p**(t) is a probability distribution on pages
  - **p**(t) is a link-based importance rank t->∞

PageRank: Problems

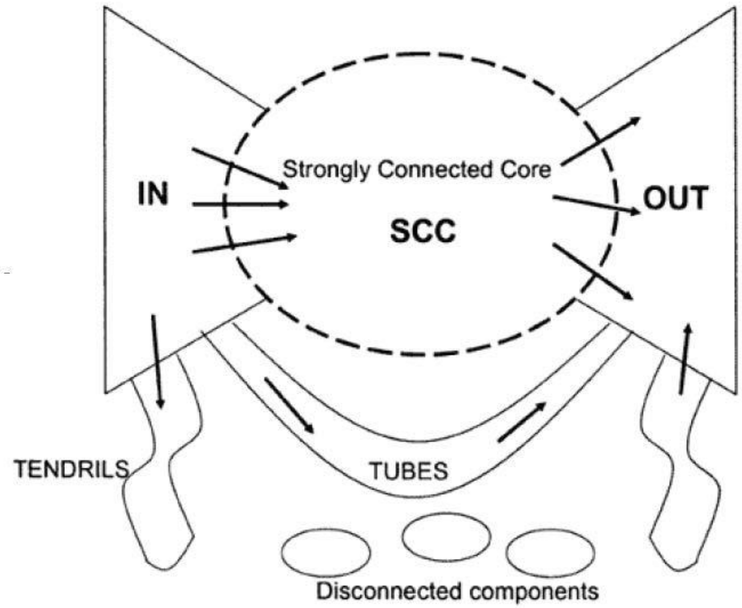
2 problems:

(1) Spider traps (all out-links are within the group)

- Eventually spider traps absorb all importance

(2) Some pages are **dead ends** (have no out-links)

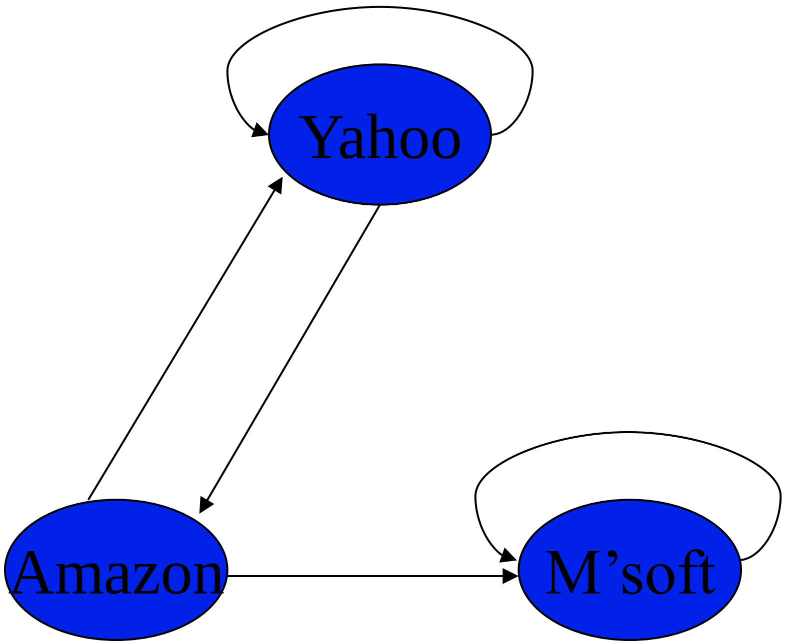
- Such pages cause importance to “leak out”



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
- Spider traps violate the conditions needed for the random walk theorem

Microsoft becomes a spider trap



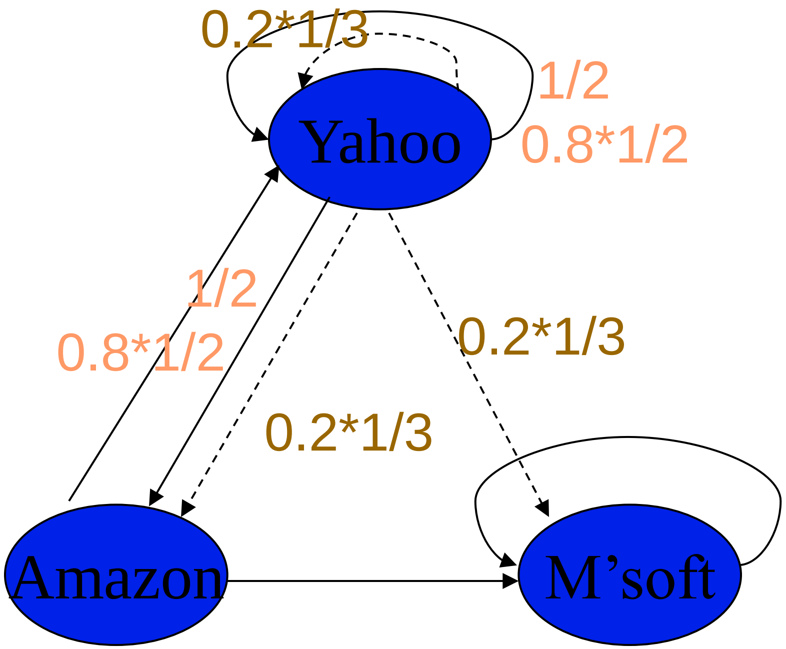
	y	a	m
y	1/2	1/2	0
a	1/2	0	0
m	0	1/2	1

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

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-β, jump to some page uniformly at random
  - Common values for β are in the range 0.8 to 0.9
- Results: Surfer will teleport out of spider trap within a few time steps

Random teleports ($\beta = 0.8$)

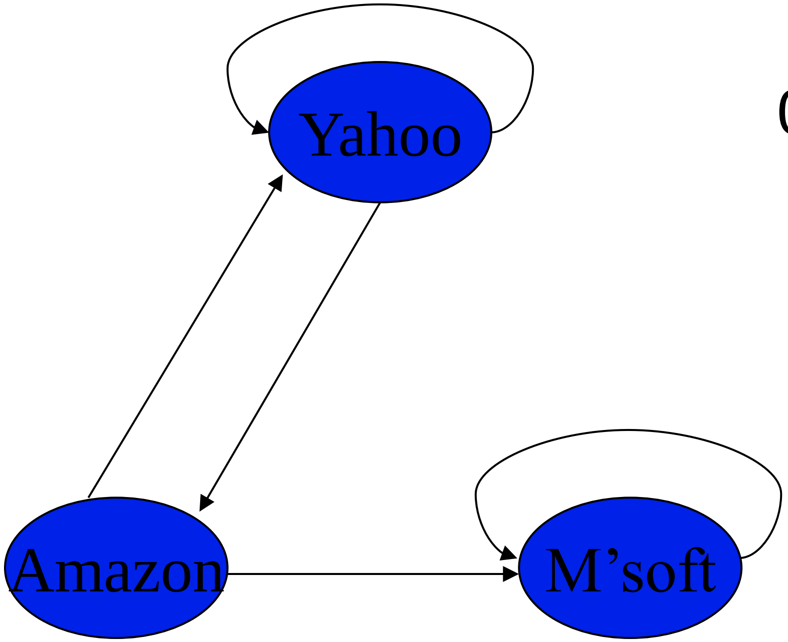


$$
\begin{array}{c} \\ y \\ a \\ m \end{array}
\begin{array}{c} y \\ \boxed{\begin{array}{c} 1/2 \\ 1/2 \\ 0 \end{array}} \end{array}
\qquad 0.8* \begin{array}{c} y \\ \boxed{\begin{array}{c} 1/2 \\ 1/2 \\ 0 \end{array}} \end{array}
+ 0.2* \begin{array}{c} y \\ \boxed{\begin{array}{c} 1/3 \\ 1/3 \\ 1/3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

$$
0.8 \boxed{\begin{array}{ccc} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{array}}
+ 0.2 \boxed{\begin{array}{ccc} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{array}}
$$

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

Random teleports ($\beta = 0.8$)



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

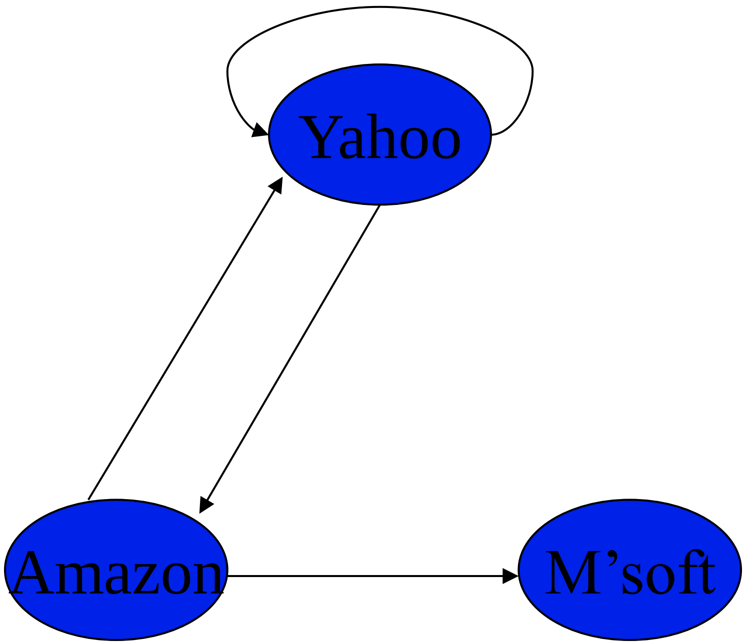
$$\begin{array}{c} y \\ a \\ m \end{array} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

$$\begin{array}{cccccccc} y & & 1 & 1.00 & 0.84 & 0.776 & & 7/11 \\ a & = & 1 & 0.60 & 0.60 & 0.536 & \ldots & 5/11 \\ m & & 1 & 1.40 & 1.56 & 1.688 & & 21/11 \end{array}$$

Dead ends

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

Microsoft becomes a dead end



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

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

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \quad \begin{array}{c} 1 \\ 0.6 \\ 0.6 \end{array} \quad \begin{array}{c} 0.787 \\ 0.547 \\ 0.387 \end{array} \quad \begin{array}{c} 0.648 \\ 0.430 \\ 0.333 \end{array} \quad \ldots \quad \begin{array}{c} 0 \\ 0 \\ 0 \end{array}$$

Dealing with dead-ends

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