Data and Algorithms of the Web

Link Analysis Algorithms
Page Rank

some slides from:
Anand Rajaraman, Jeffrey D. Ullman
InfoLab (Stanford University)

Link Analysis Algorithms

- Page Rank
- Hubs and Authorities
- Topic-Specific Page Rank
- Spam Detection Algorithms
- Other interesting topics we won't cover
  - Detecting duplicates and mirrors
  - Mining for communities

Ranking web pages

Ranking web pages

- Web pages are not equally “important”

Ranking web pages

- Web pages are not equally "important"
- www.bernard.com and www.stanford.edu contain both the term "stanford" but:
  - www.stanford.edu has 23,400 webpages linking to it
  - www.bernard.com has 10 webpages linking to it

Ranking web pages

- Web pages are not equally "important"
- www.bernard.com and www.stanford.edu contain both the term "stanford" but:
  - www.stanford.edu has 23,400 webpages linking to it
  - www.bernard.com has 10 webpages linking to it
- Are all webpages linking to www.stanford.edu equally important?
  - The webpage of MIT is more "important" than the webpage of a friend of bernard

Ranking web pages

- Web pages are not equally "important"
- www.bernard.com and www.stanford.edu both contain both the term "stanford" but:
  - www.stanford.edu has 23,400 webpages linking to it
  - www.bernard.com has 10 webpages linking to it
- Are all webpages linking to www.stanford.edu equally important?
  - The webpage of MIT is more "important" than the webpage of a friend of bernard

-> Recursive definition of importance

Simple recursive formulation

Simple recursive formulation

- The importance of a page P is proportional to the importance of pages Q where Q -> P (predecessors).

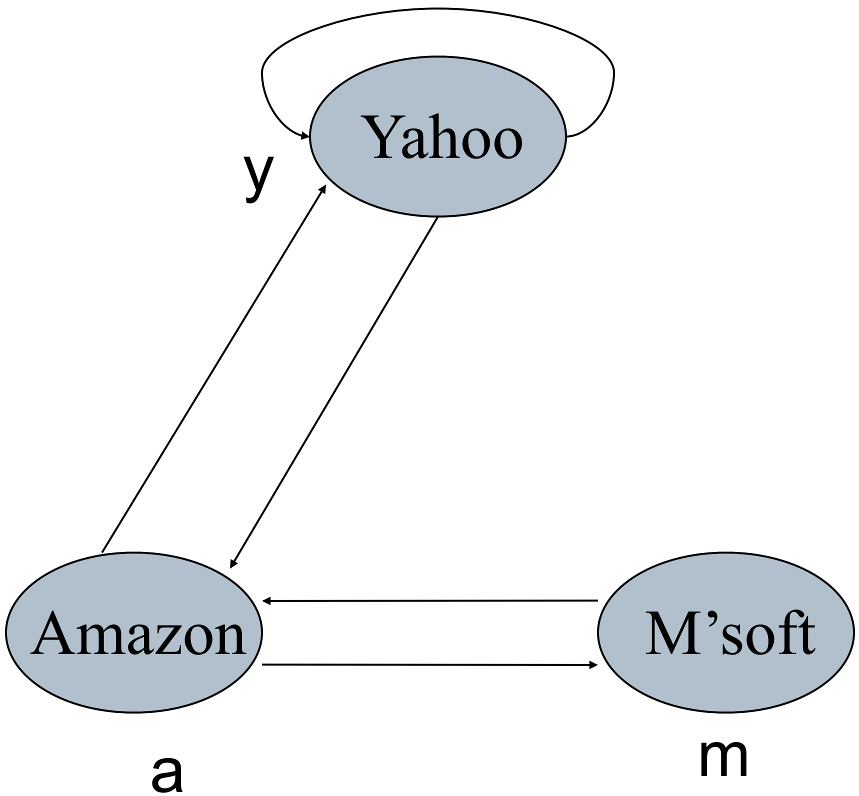
Simple recursive formulation

- The importance of a page P is proportional to the importance of pages Q where Q -> P (predecessors).
- Each page Q votes for its successors. If page Q with importance x has n successors, each succ. P gets x/n votes

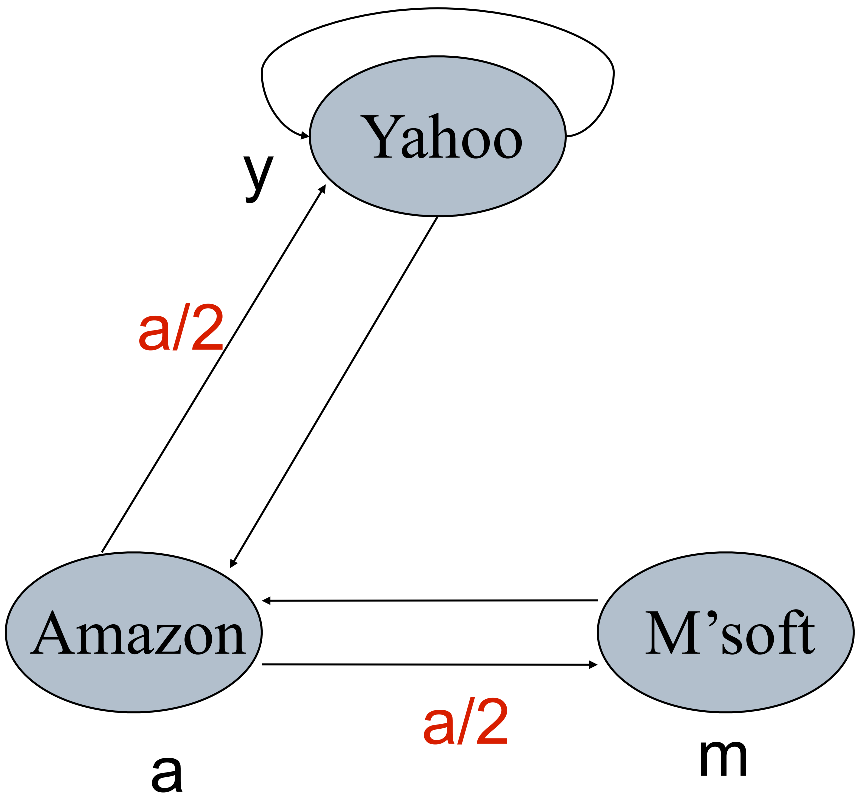
Simple recursive formulation

- The importance of a page P is proportional to the importance of pages Q where Q -> P (predecessors).
- Each page Q votes for its successors. If page Q with importance x has n successors, each succ. P gets x/n votes
- Page P's own importance is the sum of the votes of its predecessors Q.

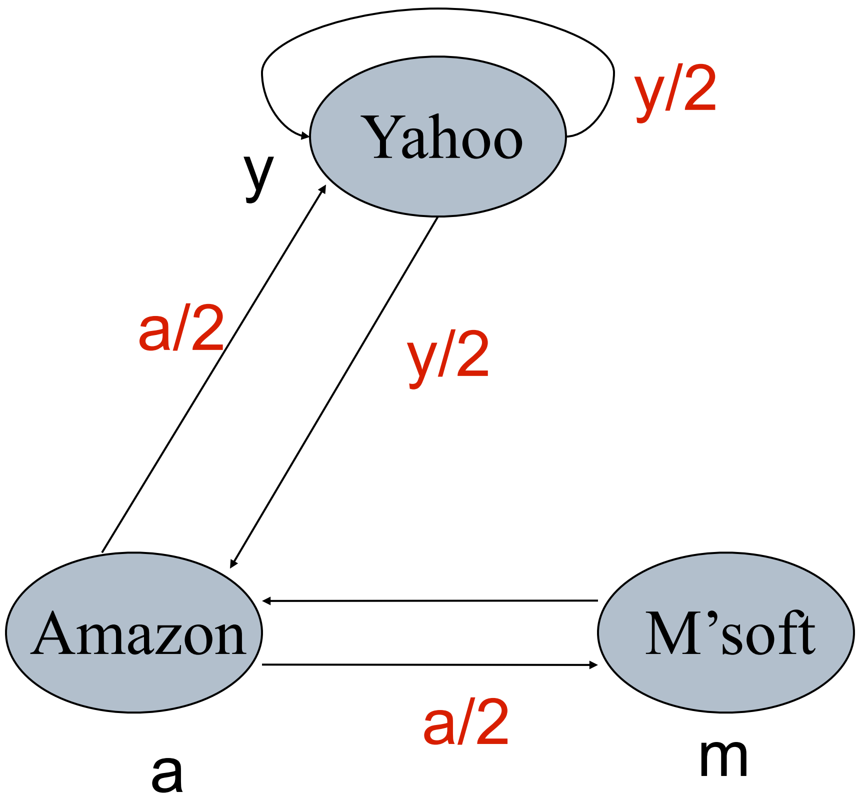
Simple "flow" model



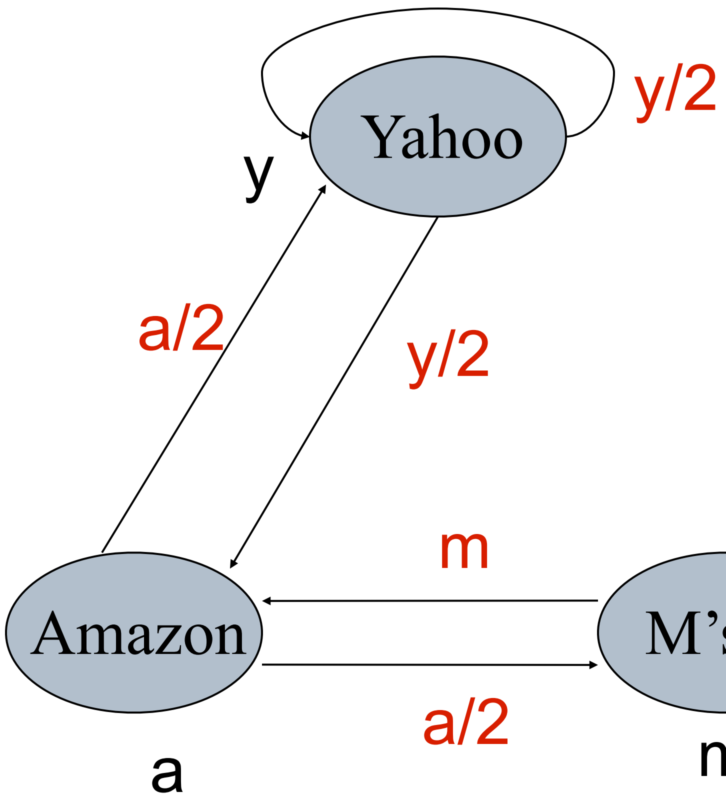
Simple “flow” model



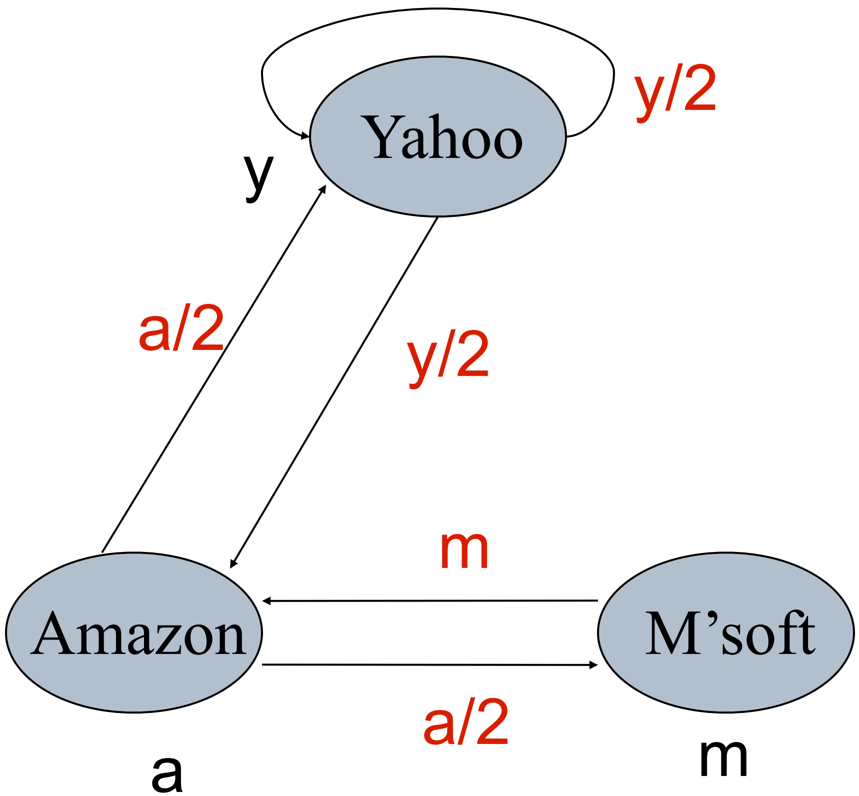
Simple "flow" model



Simple "flow" model



Simple "flow" model



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

$$a = y/2 + m$$

$$m = a/2$$

Solving the flow equations

Solving the flow equations

- 3 equations, 3 unknowns, no constants
  - No unique solution
  - All solutions equivalent modulo scale factor

Solving the flow equations

- 3 equations, 3 unknowns, no constants
  - No unique solution
  - All solutions equivalent modulo scale factor
- Additional constraint forces uniqueness
  - $y+a+m = 1$
  - $y = 2/5$, $a = 2/5$, $m = 1/5$

Solving the flow equations

- 3 equations, 3 unknowns, no constants
  - No unique solution
  - All solutions equivalent modulo scale factor
- Additional constraint forces uniqueness
  - $y+a+m = 1$
  - $y = 2/5$, $a = 2/5$, $m = 1/5$
- Gaussian elimination method works for small examples, but we need a better method for large graphs

Matrix formulation

Matrix formulation

- Matrix **M** has one row and one column for each web page (n x n, where n is the num of pages)

Matrix formulation

- Matrix **M** has one row and one column for each web page (n x n, where n is the num of pages)
- Suppose page j has k successors
  - If j -> i, then $M_{ij}=1/k$
  - Else $M_{ij}=0$

Matrix formulation

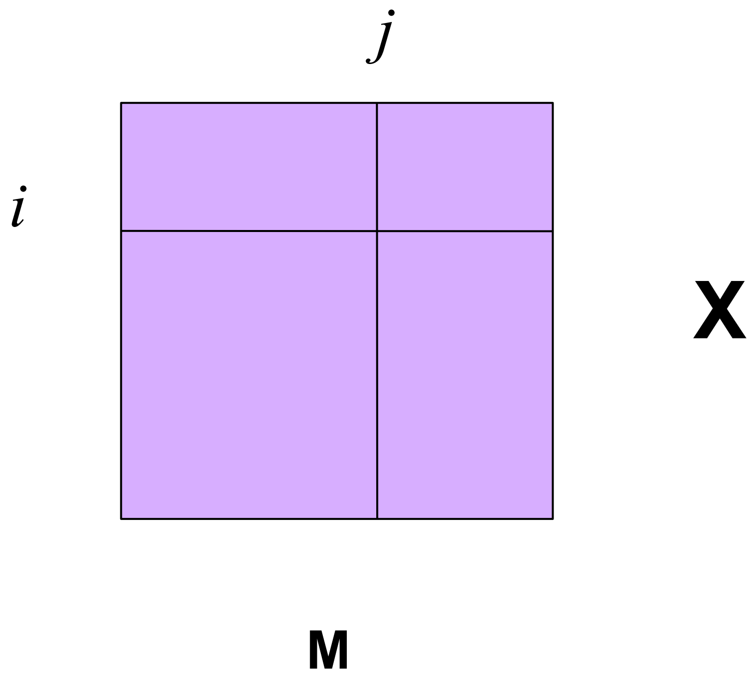
- Matrix **M** has one row and one column for each web page (n x n, where n is the num of pages)
- Suppose page j has k successors
  - If j -> i, then $M_{ij}=1/k$
  - Else $M_{ij}=0$
- **M** is a column stochastic matrix
  - Columns sum to 1

Matrix formulation

- Matrix **M** has one row and one column for each web page (n x n, where n is the num of pages)
- Suppose page j has k successors
  - If j -> i, then $M_{ij}=1/k$
  - Else $M_{ij}=0$
- **M** is a column stochastic matrix
  - Columns sum to 1
- Let **r** be the rank vector where:
  - r_i is the importance score of page i
  - $|\mathbf{r}| = 1$

Example

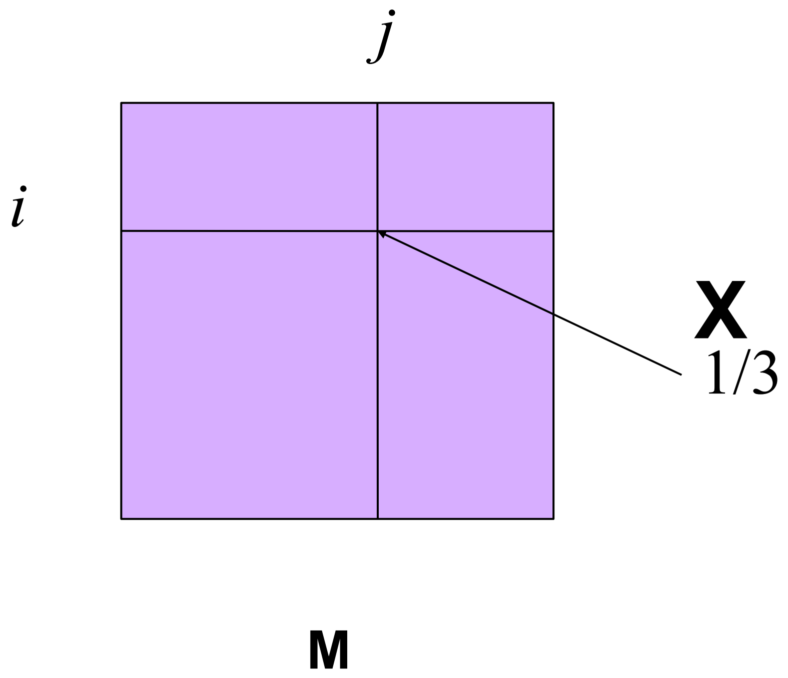
Suppose page *j* links to 3 pages, including *i*



r_i (contribution from predecessors) is obtained by multiplying i*th* row of M with r

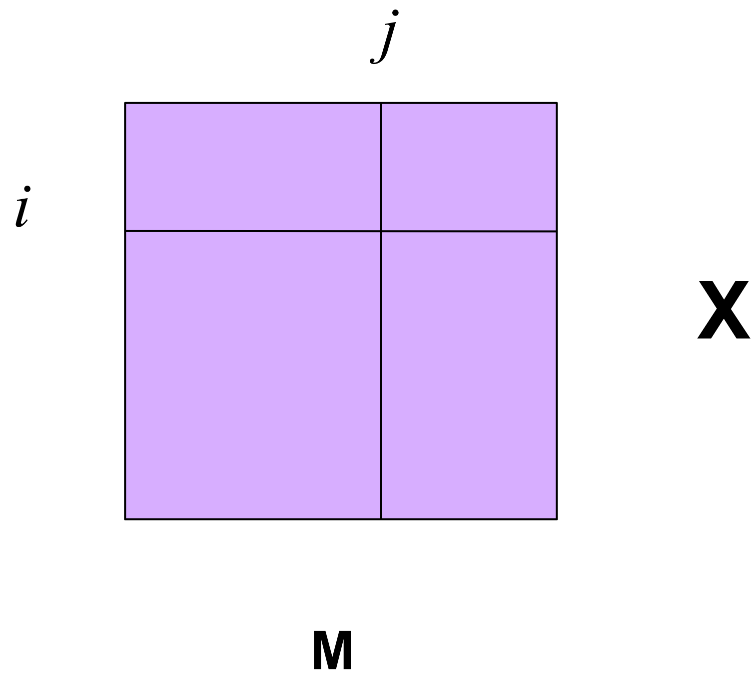
Example

Suppose page j links to 3 pages, including i



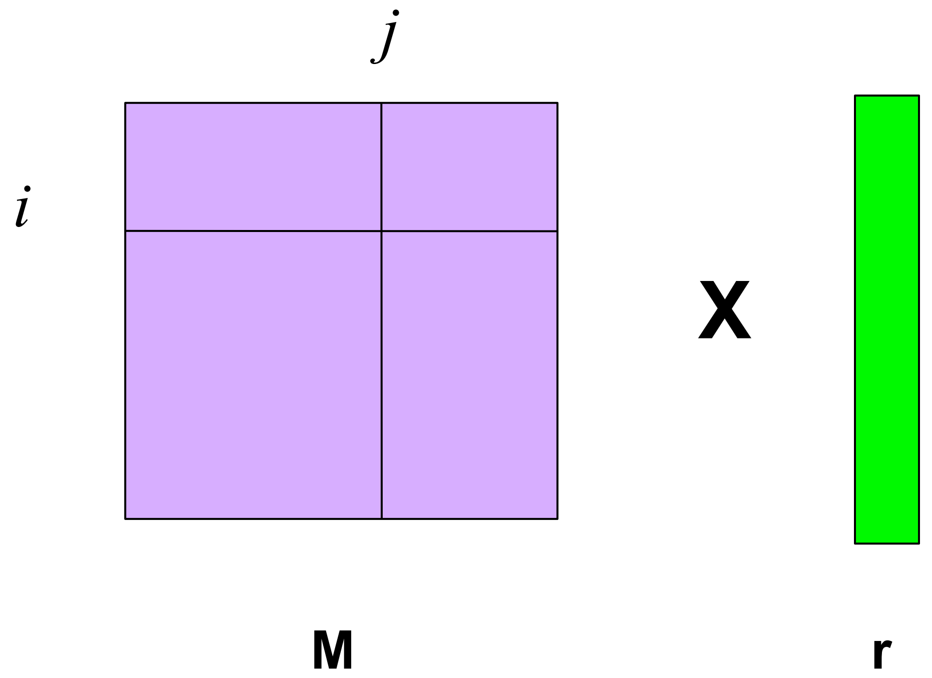
r_i (contribution from predecessors) is obtained by multiplying i*th* row of M with r

Example



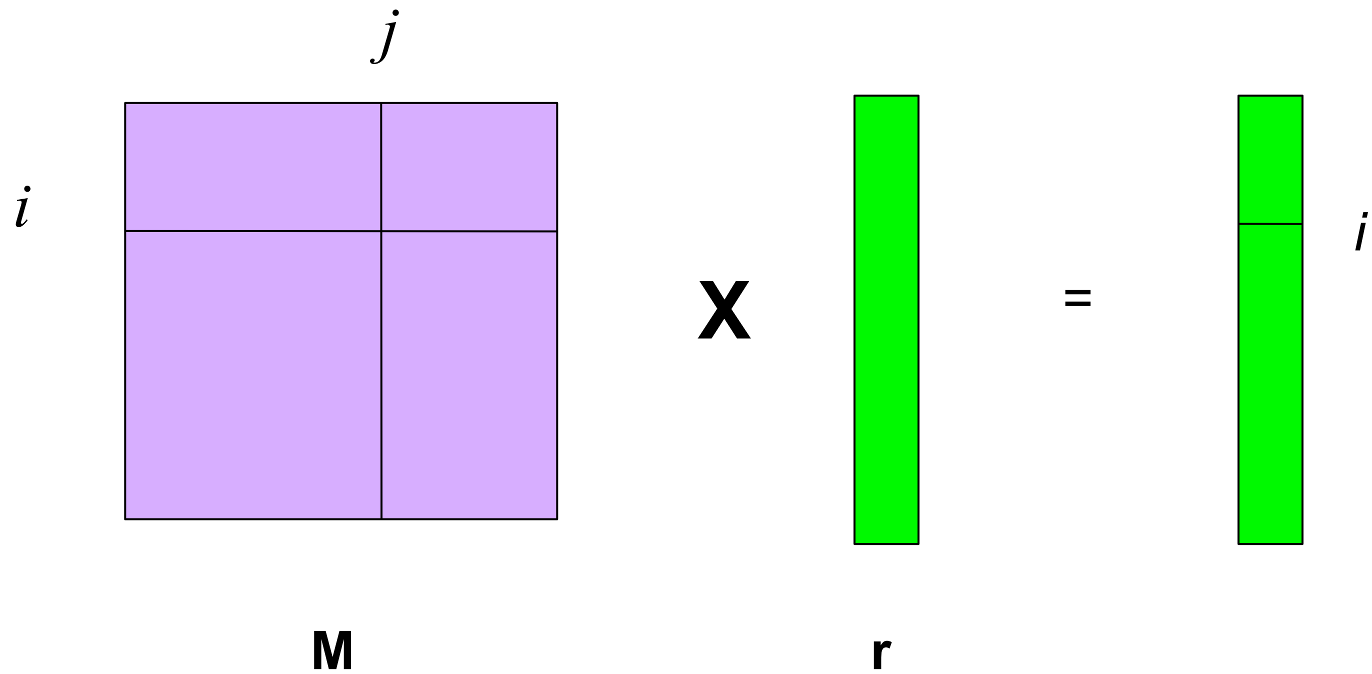
r_i (contribution from predecessors) is obtained by multiplying i*th* row of M with r

Example



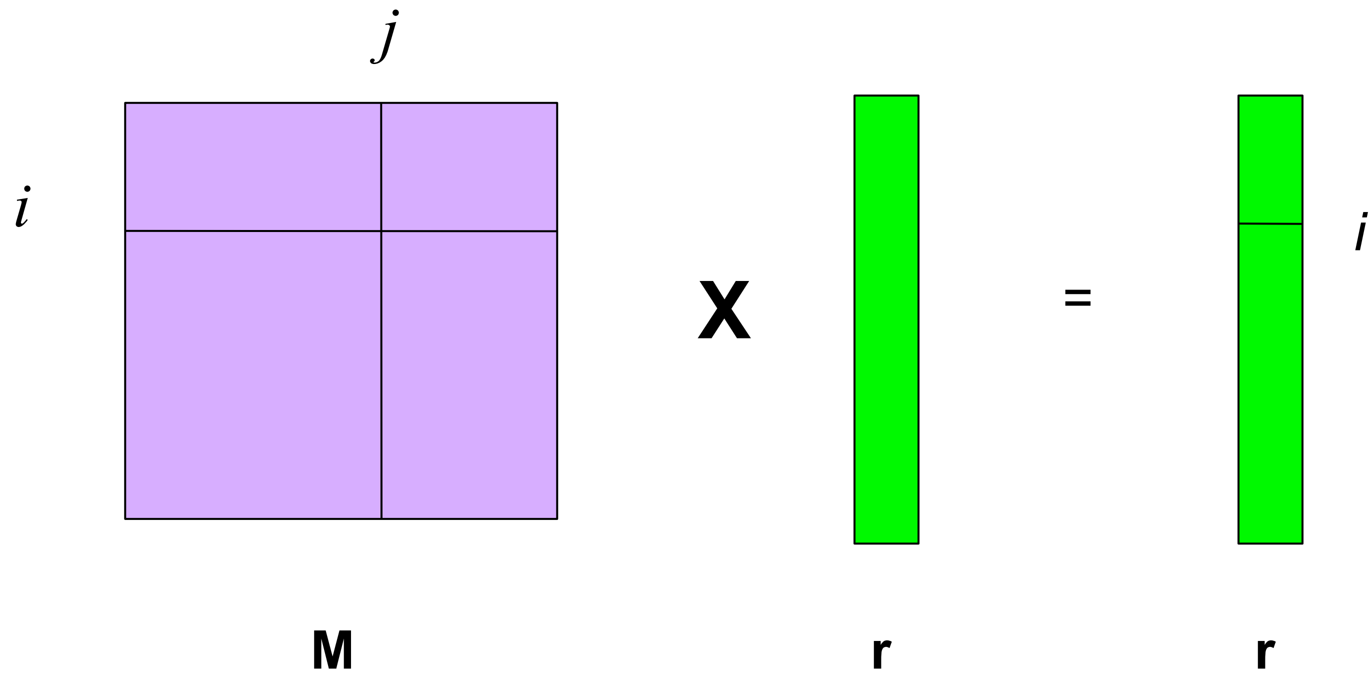
r_i (contribution from predecessors) is obtained by multiplying i*th* row of M with r

Example



r_i (contribution from predecessors) is obtained by multiplying i*th* row of M with r

Example



r_i (contribution from predecessors) is obtained by multiplying i*th* row of M with r

Eigenvector formulation

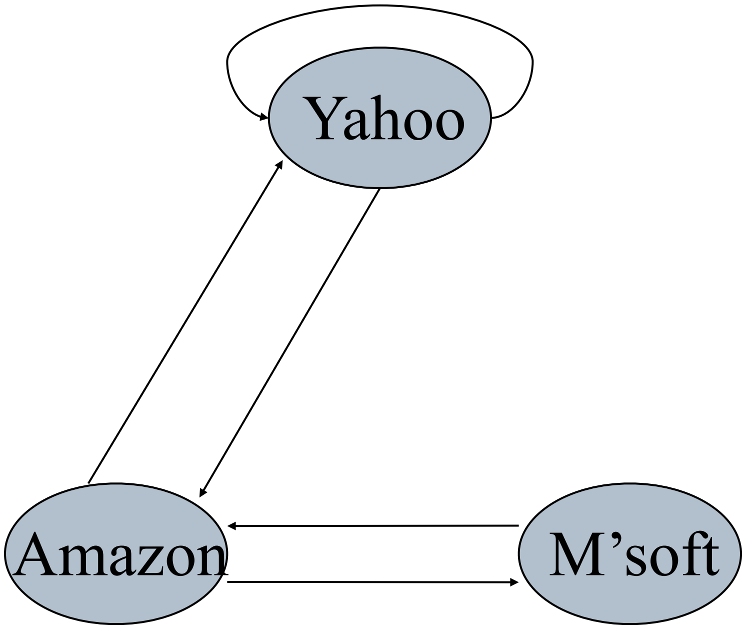
- The system of linear eq. can be written

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

- So the rank vector is an eigenvector of the stochastic web matrix
  - In fact, its first or principal eigenvector, with corresponding eigenvalue...

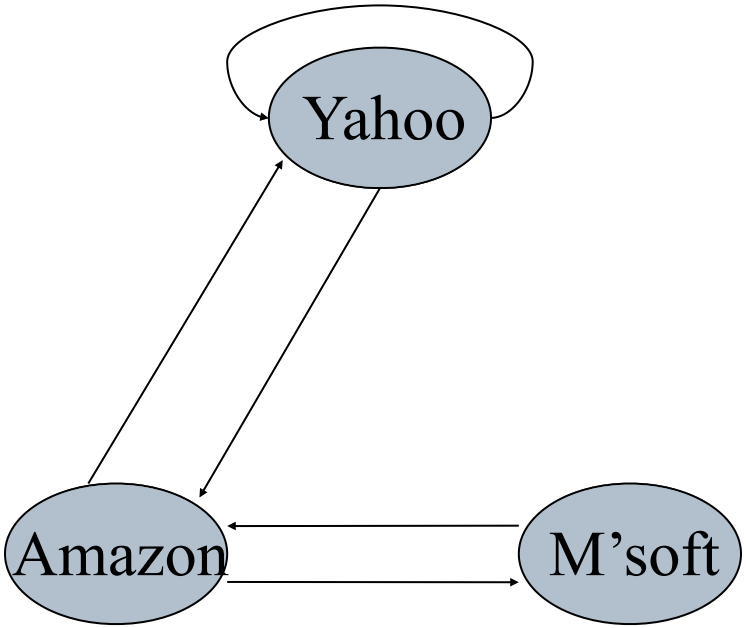
Definition. The vector **x** is an eigenvector of the matrix *A* with eigenvalue λ (lambda) if the following equation holds: $Ax = \lambda x$.

Example



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

Example

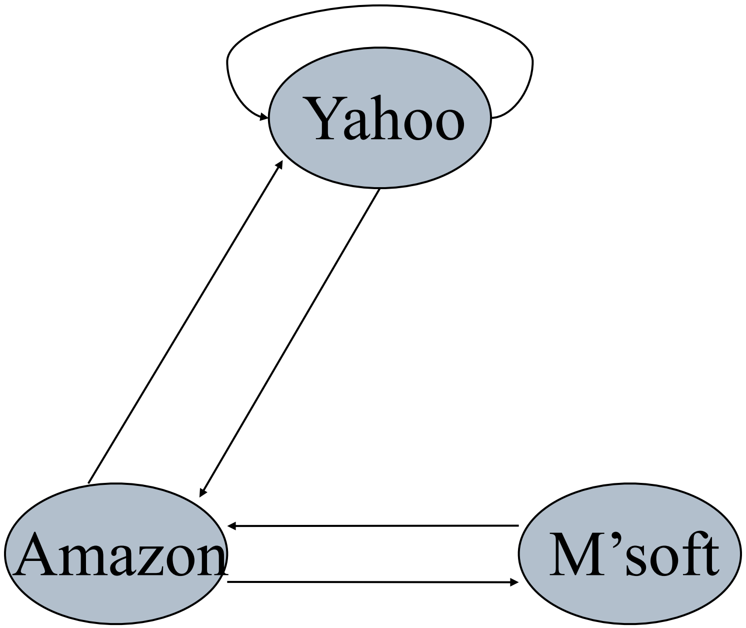


$$
\begin{array}{c|ccc|}
 & y & a & m \\
y & 1/2 & 1/2 & 0 \\
a & 1/2 & 0 & 1 \\
m & 0 & 1/2 & 0
\end{array}
$$

r = Mr

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

Example



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

r = Mr

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

$$a = y/2 + m$$

$$m = a/2$$

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

Power Iteration method

Power Iteration method

- Simple iterative scheme (aka relaxation)

Power Iteration method

- Simple iterative scheme (aka relaxation)
- Suppose there are N web pages

Power Iteration method

- Simple iterative scheme (aka relaxation)
- Suppose there are N web pages
- Initialize: $\mathbf{r}^0 = [1/N,\ldots.,1/N]^T$

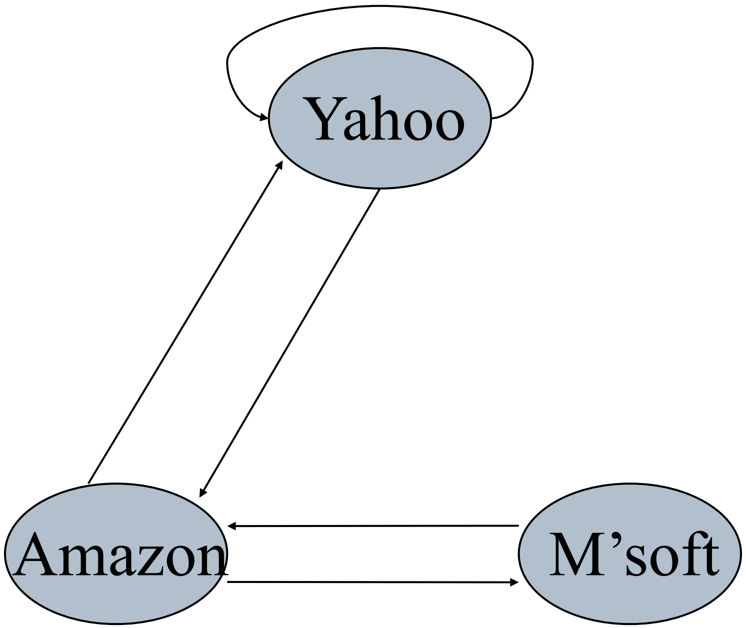
Power Iteration method

- Simple iterative scheme (aka relaxation)
- Suppose there are N web pages
- Initialize: $\mathbf{r}^0 = [1/N,\ldots.,1/N]^T$
- Iterate: $\mathbf{r}^{k+1} = \mathbf{M}\mathbf{r}^k$

Power Iteration method

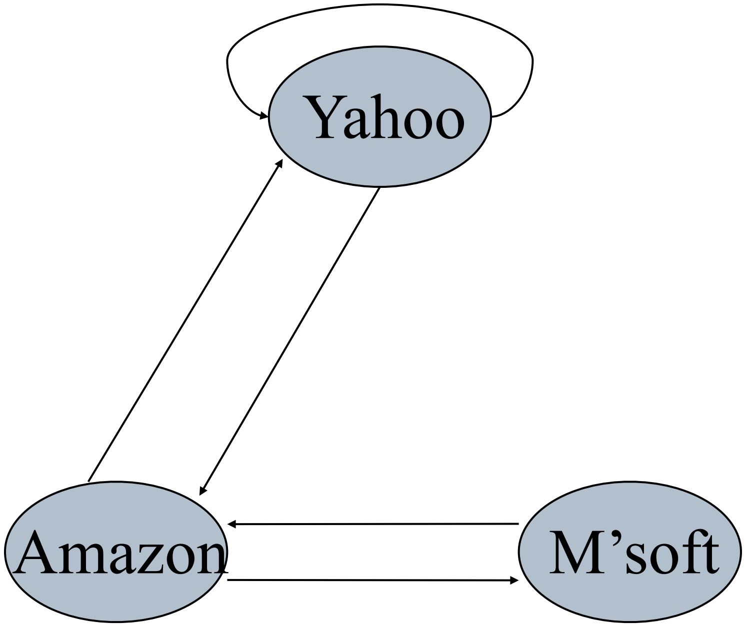
- Simple iterative scheme (aka relaxation)
- Suppose there are N web pages
- Initialize: $\mathbf{r}^0 = [1/N,\ldots.,1/N]^T$
- Iterate: $\mathbf{r}^{k+1} = \mathbf{M}\mathbf{r}^k$
- Stop when $|\mathbf{r}^{k+1} - \mathbf{r}^k|_1 < \varepsilon$
  - $|\mathbf{x}|_1 = \sum_{1 \le i \le N} |x_i|$ is the L_1 norm
  - Can use any other vector norm e.g., Euclidean

Power Iteration Example



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

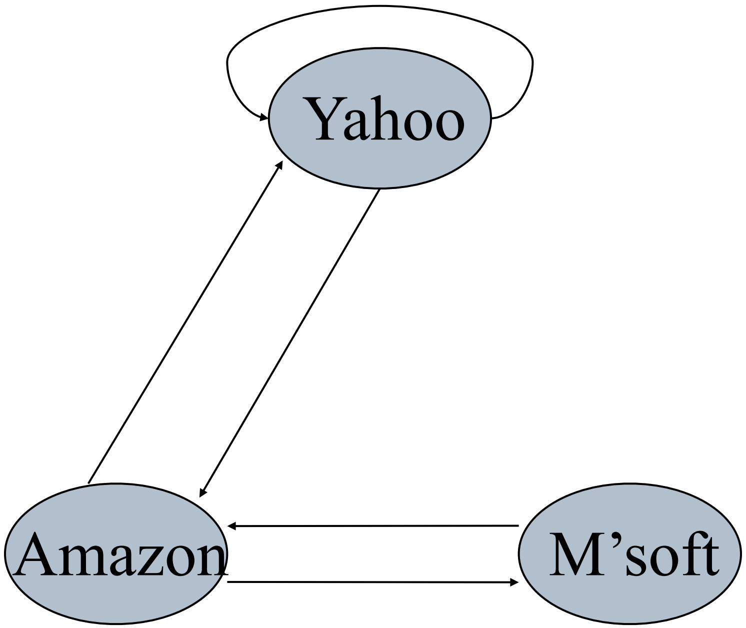
Power Iteration Example



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

$$\begin{array}{c} y \\ a \\ m \end{array} =$$

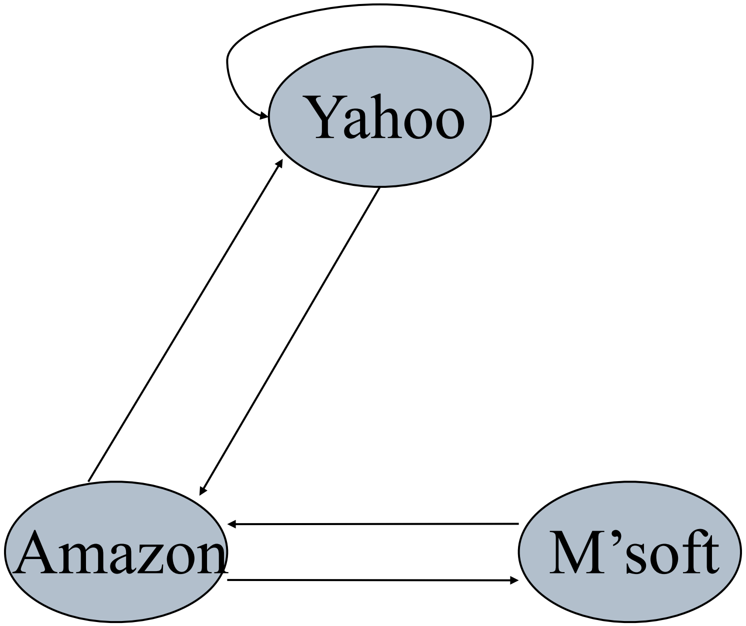
Power Iteration Example



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

$$
\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{c} 1/3 \\ 1/3 \\ 1/3 \end{array}
$$

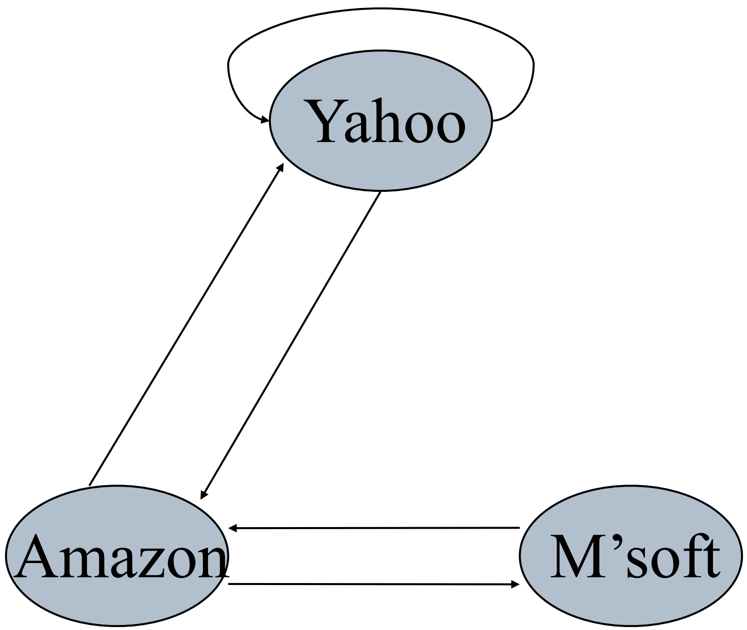
Power Iteration Example



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

$$
\begin{array}{c}
y \\ a \\ m
\end{array}
=
\begin{array}{c}
1/3 \\ 1/3 \\ 1/3
\end{array}
\quad
\begin{array}{c}
1/3 \\ 1/2 \\ 1/6
\end{array}
$$

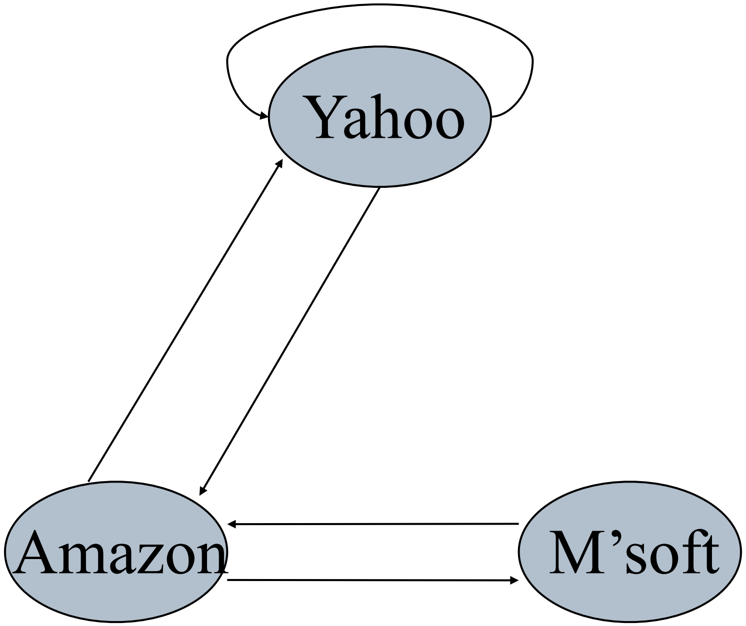
Power Iteration Example



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

$$
\begin{array}{c}
y \\ a \\ m
\end{array}
=
\begin{array}{c}
1/3 \\ 1/3 \\ 1/3
\end{array}
\quad
\begin{array}{c}
1/3 \\ 1/2 \\ 1/6
\end{array}
\quad
\begin{array}{c}
5/12 \\ 1/3 \\ 1/4
\end{array}
$$

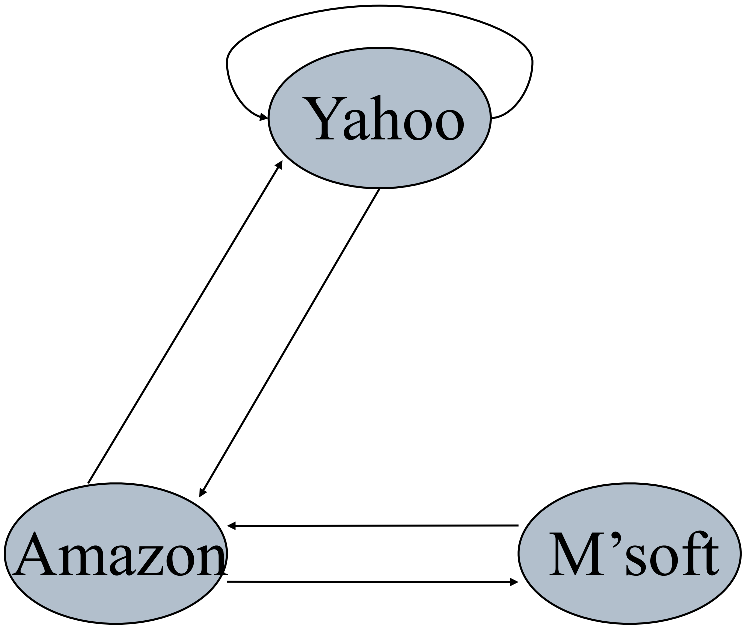
Power Iteration Example



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

$$
\begin{array}{c}
y \\ a \\ m
\end{array}
=
\begin{array}{cccc}
1/3 & 1/3 & 5/12 & 3/8 \\
1/3 & 1/2 & 1/3 & 11/24 \\
1/3 & 1/6 & 1/4 & 1/6
\end{array}
$$

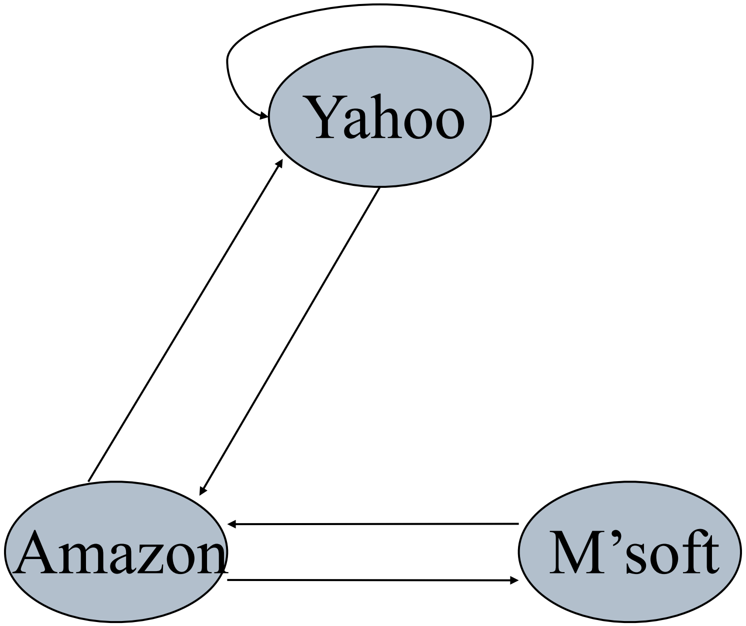
Power Iteration Example



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

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

Power Iteration Example



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

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

Random Walk Interpretation

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

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

The stationary distribution

The stationary distribution

- Where is the surfer at time t+1?
  - Follows a link uniformly at random
  - $\mathbf{p}(t+1) = \mathbf{M}\mathbf{p}(t)$

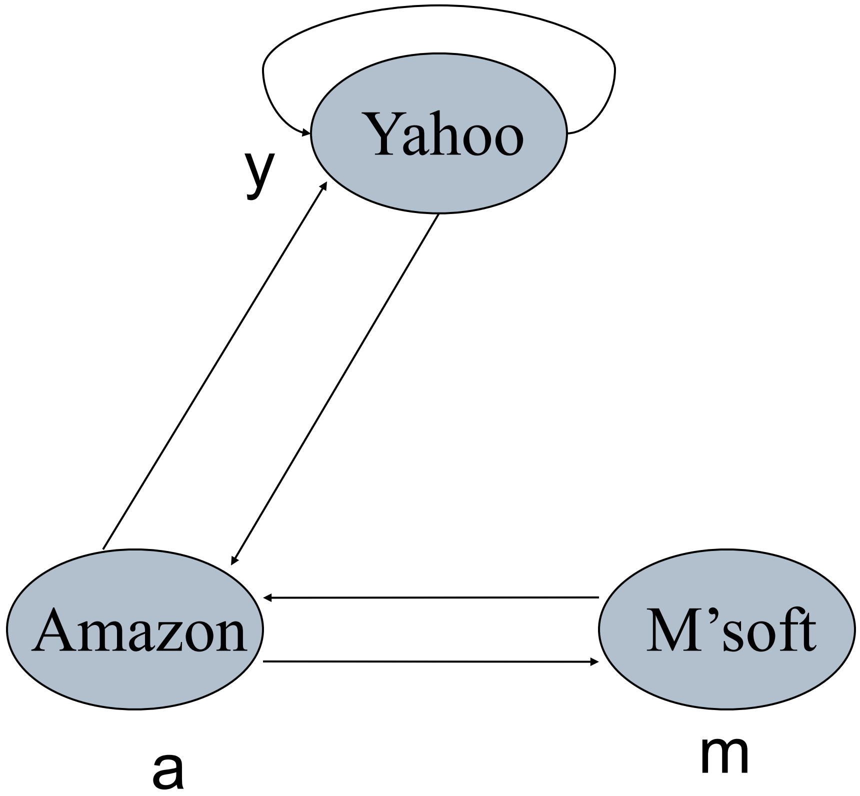
The stationary distribution

- Where is the surfer at time t+1?
  - Follows a link uniformly at random
  - $\mathbf{p}(t+1) = \mathbf{M}\mathbf{p}(t)$
- Suppose the random walk reaches a state such that $\mathbf{p}(t+1) = \mathbf{M}\mathbf{p}(t) = \mathbf{p}(t)$
  - Then $\mathbf{p}(t)$ is called a stationary distribution for the random walk

The stationary distribution

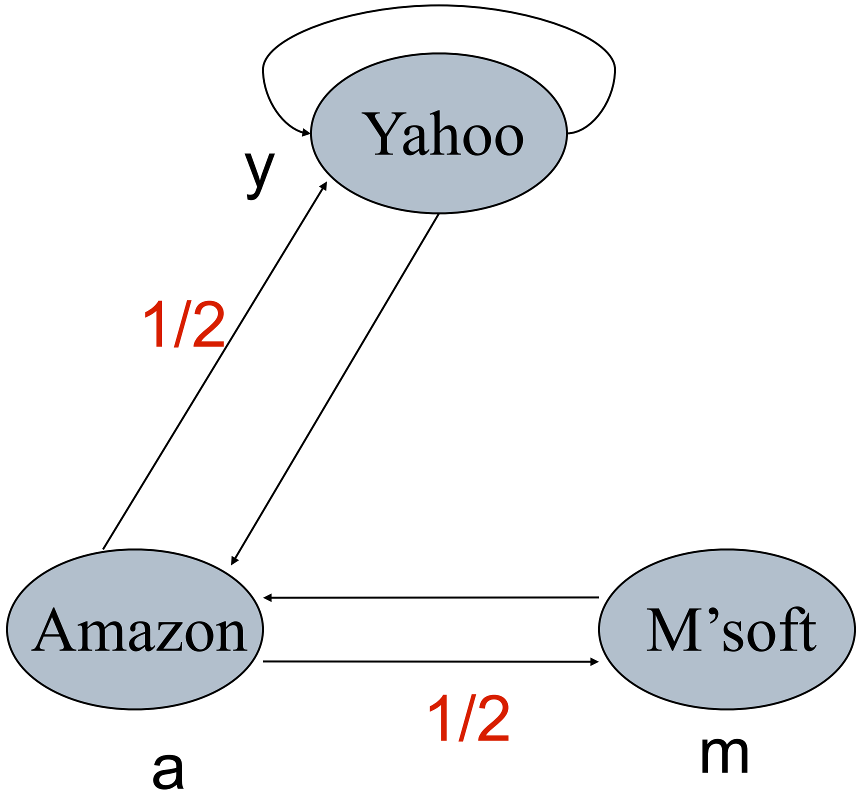
- Where is the surfer at time t+1?
  - Follows a link uniformly at random
  - $\mathbf{p}(t+1) = \mathbf{M}\mathbf{p}(t)$
- Suppose the random walk reaches a state such that $\mathbf{p}(t+1) = \mathbf{M}\mathbf{p}(t) = \mathbf{p}(t)$
  - Then $\mathbf{p}(t)$ is called a stationary distribution for the random walk
- Our rank vector $\mathbf{r}$ satisfies $\mathbf{r} = \mathbf{M}\mathbf{r}$
  - So it is a stationary distribution for the random surfer

Random walk interpretation



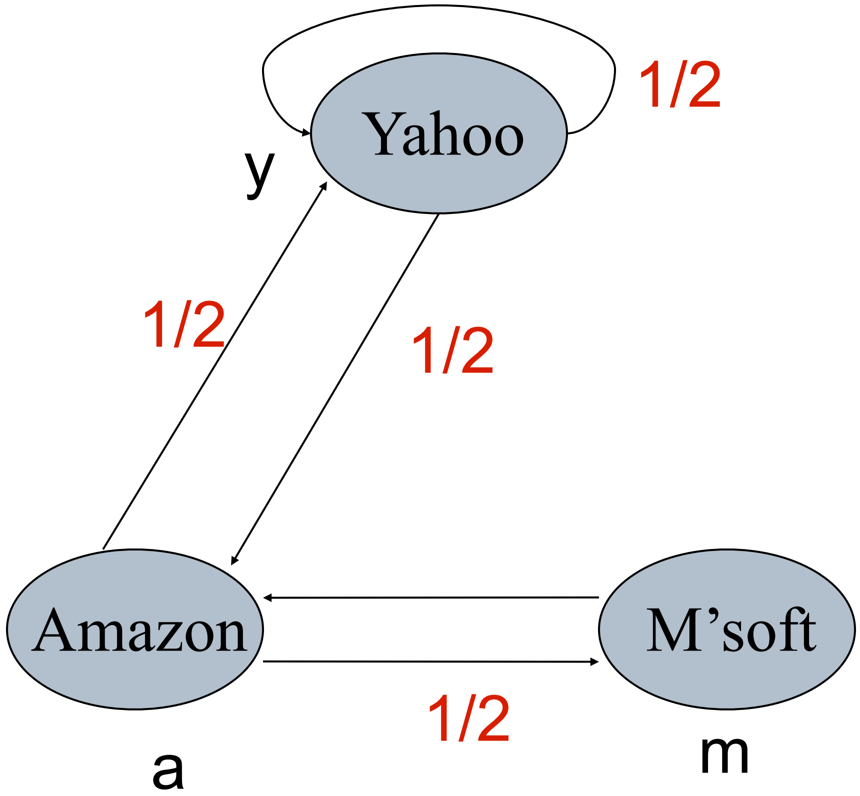
Stationary distribution

Random walk interpretation



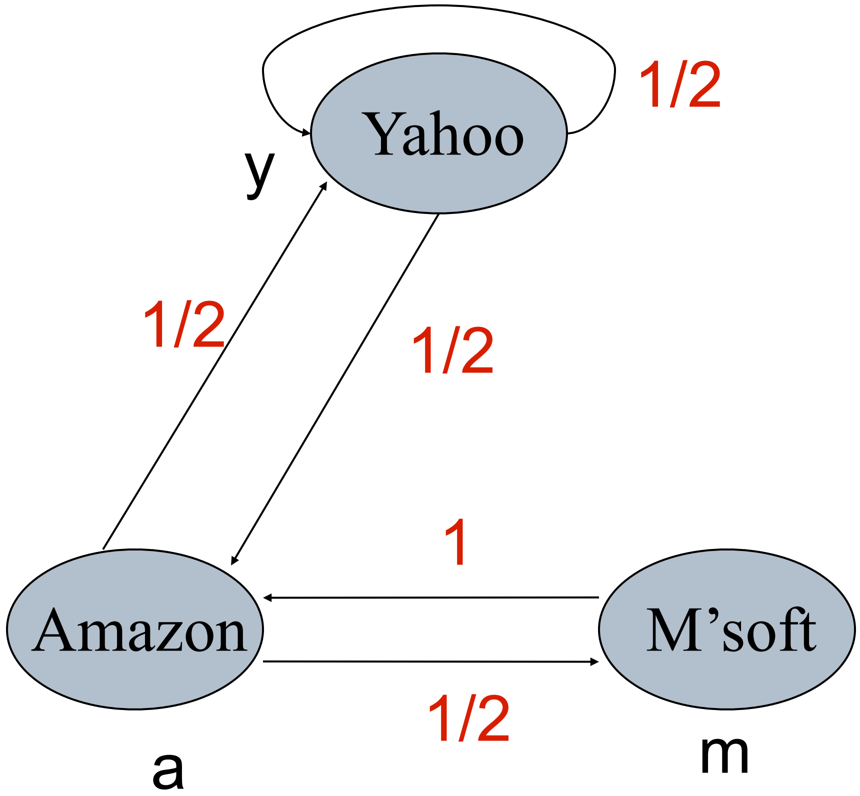
Stationary distribution

Random walk interpretation



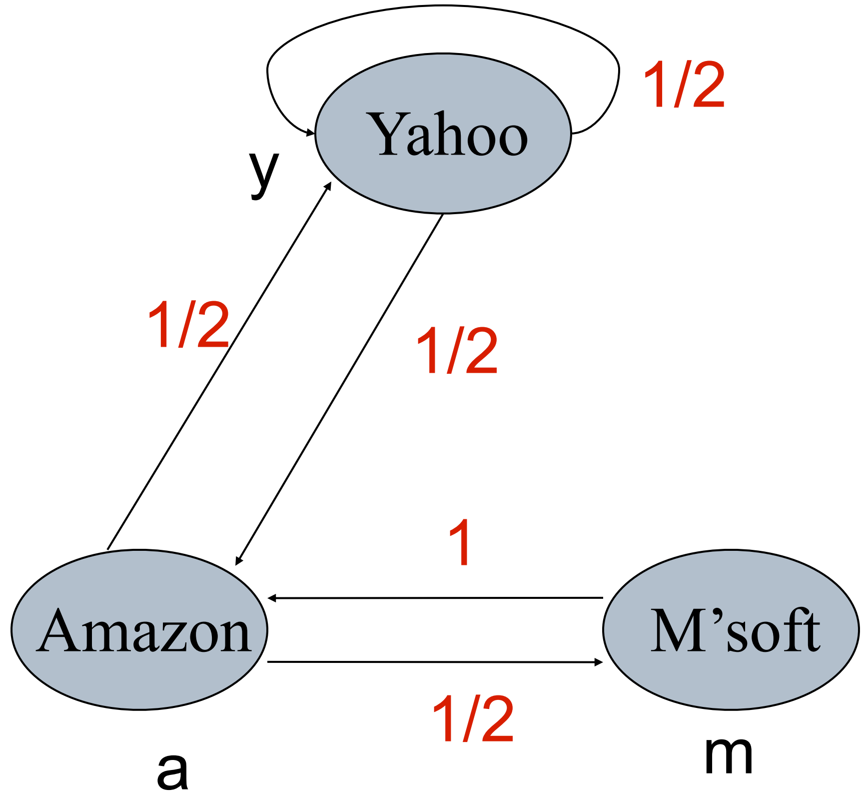
Stationary distribution

Random walk interpretation



Stationary distribution

Random walk interpretation



2/5
2/5 Stationary distribution
1/5

Existence and Uniqueness

A central result from the theory of random walks (aka Markov processes):

For graphs that satisfy certain conditions, the stationary distribution is unique and eventually will be reached no matter what the initial probability distribution at time t = 0.

Spider traps

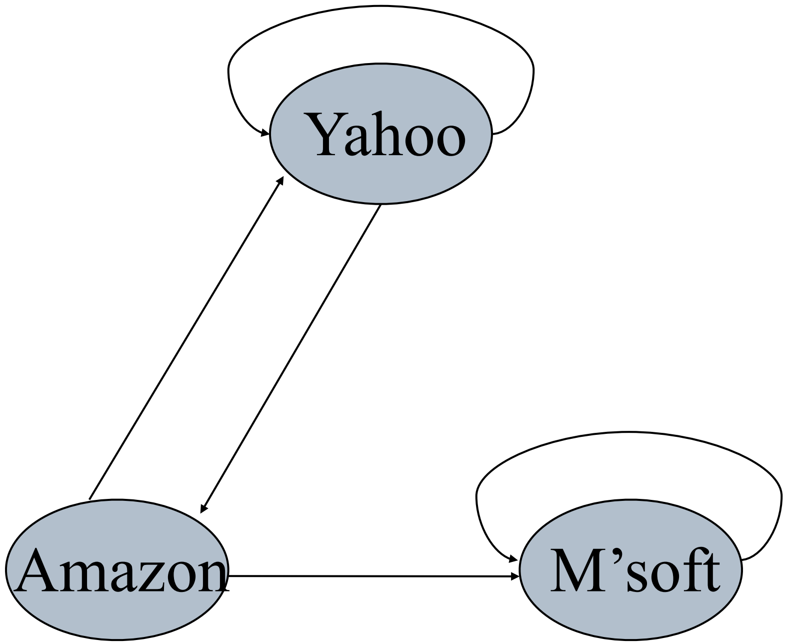
Spider traps

- 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

- 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

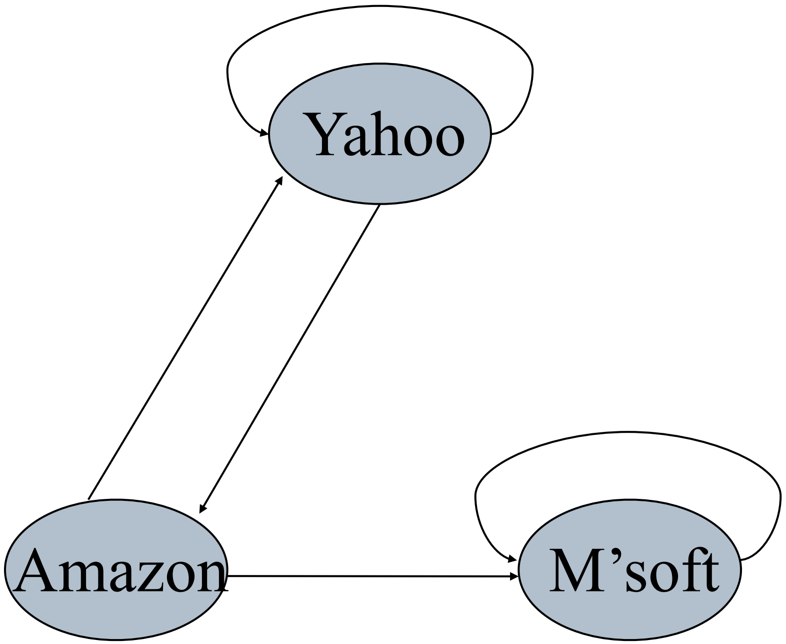
Microsoft becomes a spider trap



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

$$\begin{matrix} y \\ a \\ m \end{matrix} = \begin{matrix} 1/3 \\ 1/3 \\ 1/3 \end{matrix}$$

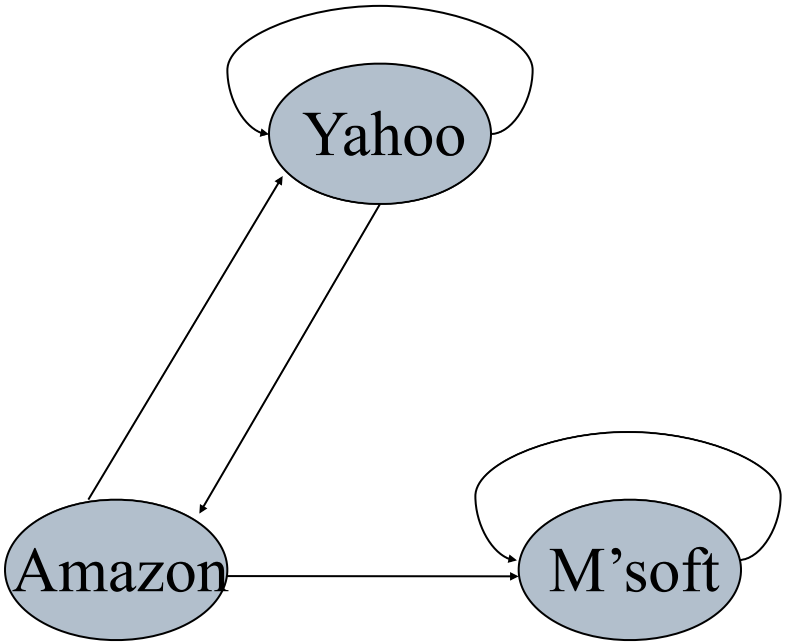
Microsoft becomes a spider trap



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

$$
\begin{array}{c}
y \\ a \\ m
\end{array}
=
\begin{array}{cc}
1/3 & 1/3 \\
1/3 & 1/6 \\
1/3 & 1/2
\end{array}
$$

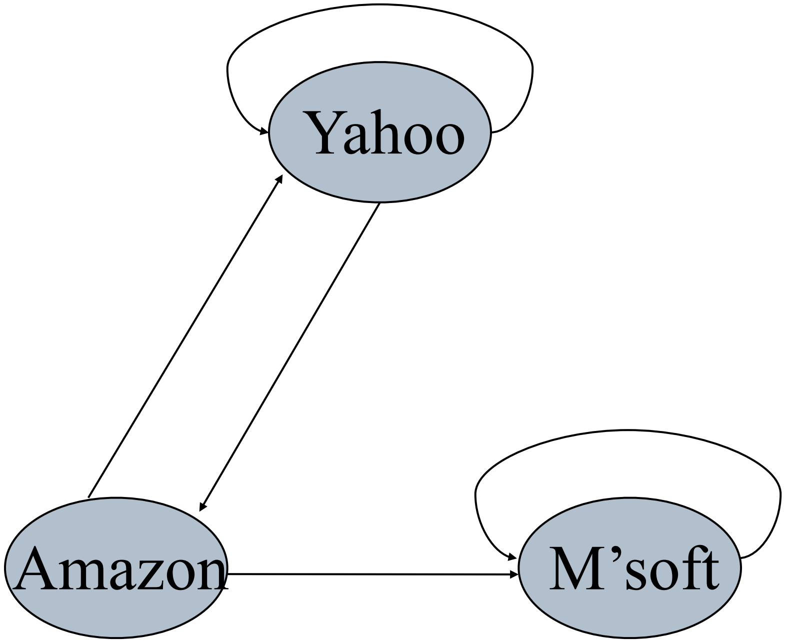
Microsoft becomes a spider trap



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

$$\begin{array}{l} y \\ a \\ m \end{array} = \begin{array}{ccccc} 1/3 & 1/3 & 1/4 \\ 1/3 & 1/6 & 1/6 \\ 1/3 & 1/2 & 7/12 \end{array}$$

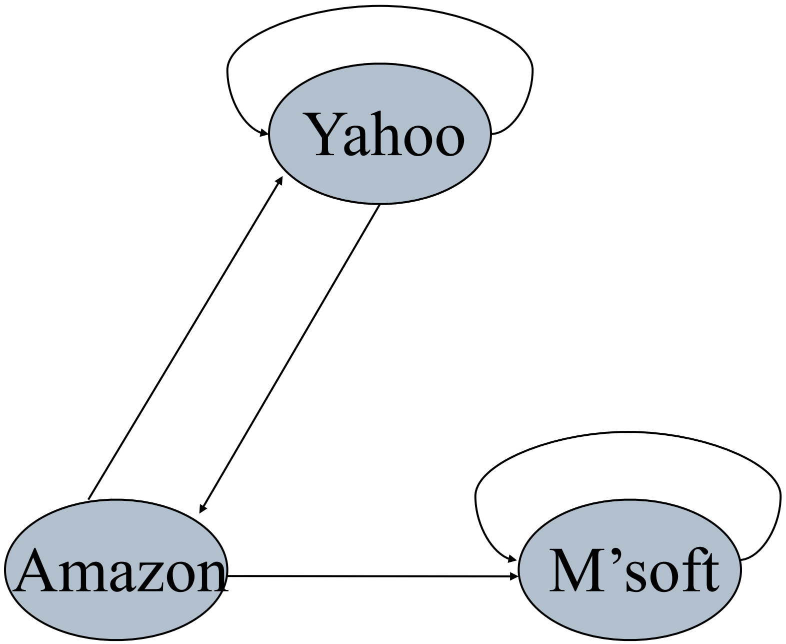
Microsoft becomes a spider trap



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

$$
\begin{array}{c}
y \\
a \\
m
\end{array}
=
\begin{array}{cccc}
1/3 & 1/3 & 1/4 & 5/24 \\
1/3 & 1/6 & 1/6 & 1/8 \\
1/3 & 1/2 & 7/12 & 2/3
\end{array}
$$

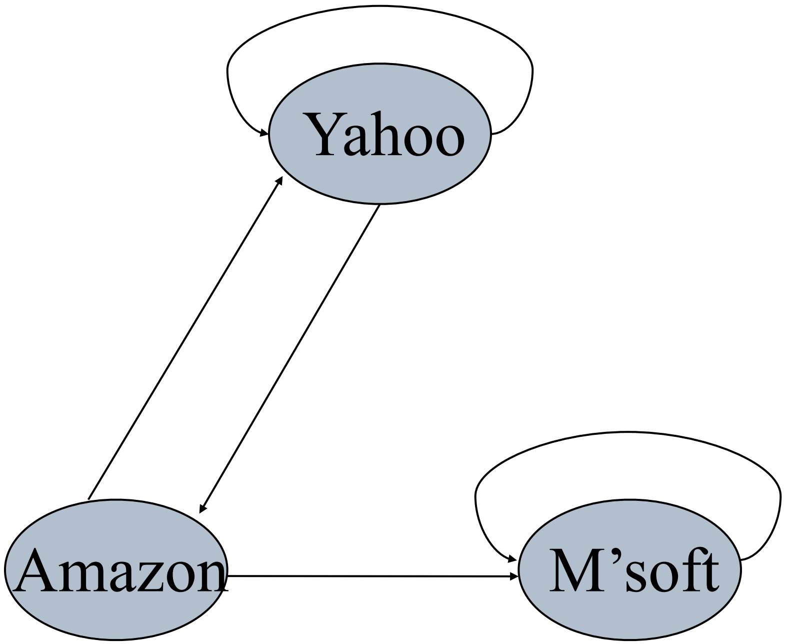
Microsoft becomes a spider trap



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

$$
\begin{array}{c} y \\ a \\ m \end{array} \; = \;
\begin{array}{c} 1/3 \\ 1/3 \\ 1/3 \end{array} \quad
\begin{array}{c} 1/3 \\ 1/6 \\ 1/2 \end{array} \quad
\begin{array}{c} 1/4 \\ 1/6 \\ 7/12 \end{array} \quad
\begin{array}{c} 5/24 \\ 1/8 \\ 2/3 \end{array} \quad \ldots
$$

Microsoft becomes a spider trap



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

$$
\begin{matrix} y \\ a \\ m \end{matrix} = \begin{matrix} 1/3 \\ 1/3 \\ 1/3 \end{matrix} \quad \begin{matrix} 1/3 \\ 1/6 \\ 1/2 \end{matrix} \quad \begin{matrix} 1/4 \\ 1/6 \\ 7/12 \end{matrix} \quad \begin{matrix} 5/24 \\ 1/8 \\ 2/3 \end{matrix} \quad \ldots \quad \begin{matrix} 0 \\ 0 \\ 1 \end{matrix}
$$

Random teleports

Random teleports

- The Google solution for spider traps

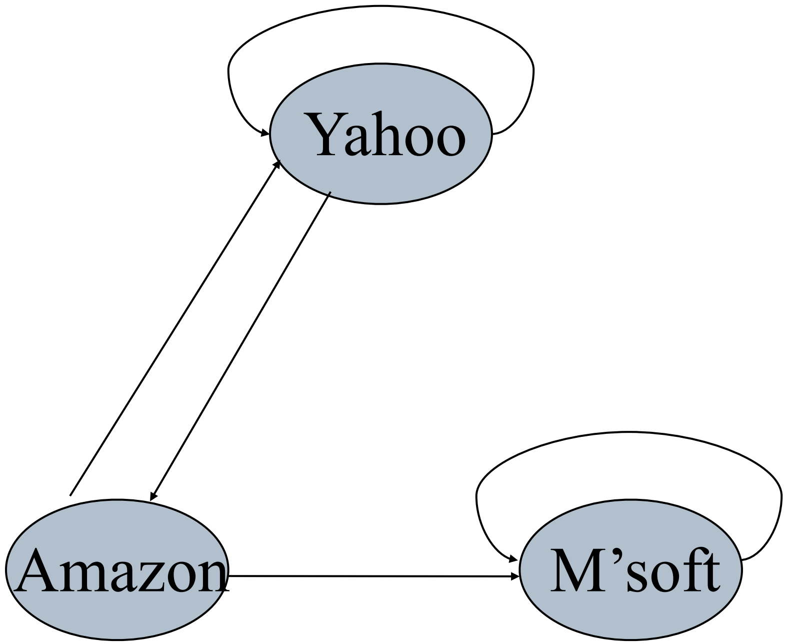
Random teleports

- 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

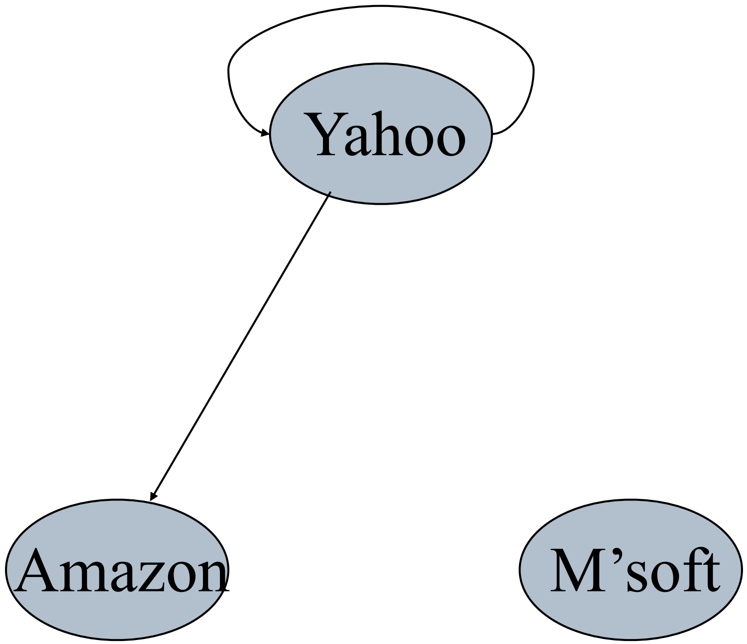
Random teleports

- 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

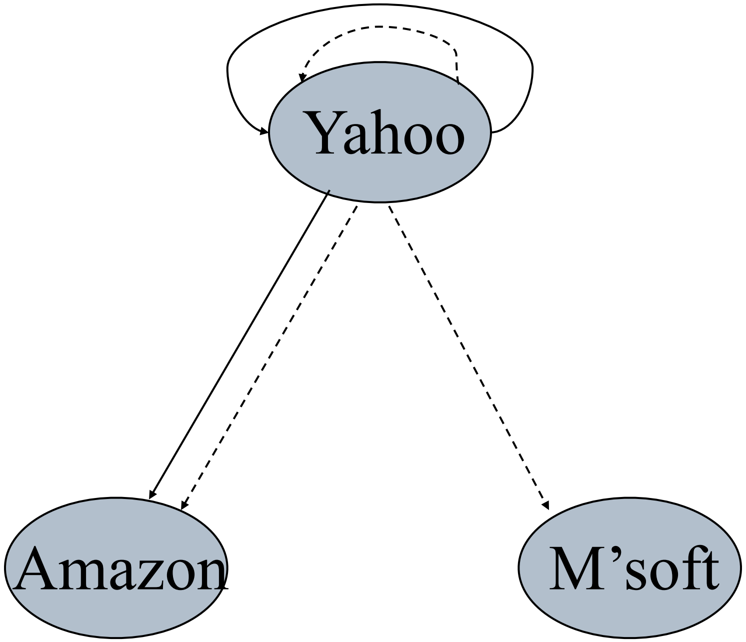
Random teleports ($\beta = 0.8$)



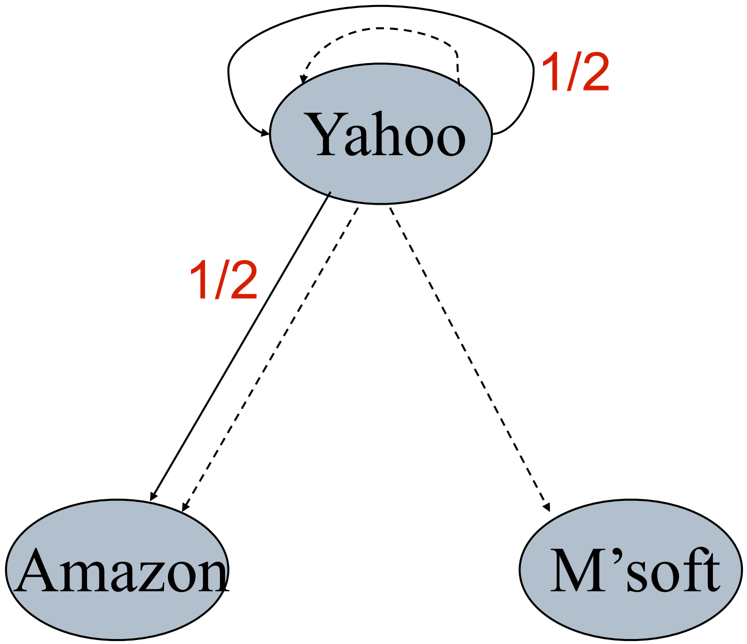
Random teleports ($\beta = 0.8$)



Random teleports ($\beta = 0.8$)

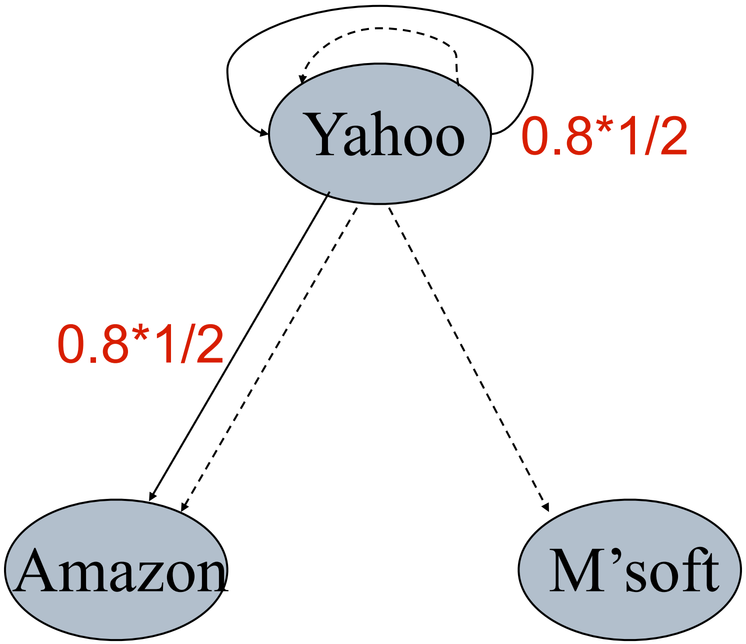


Random teleports ($\beta = 0.8$)

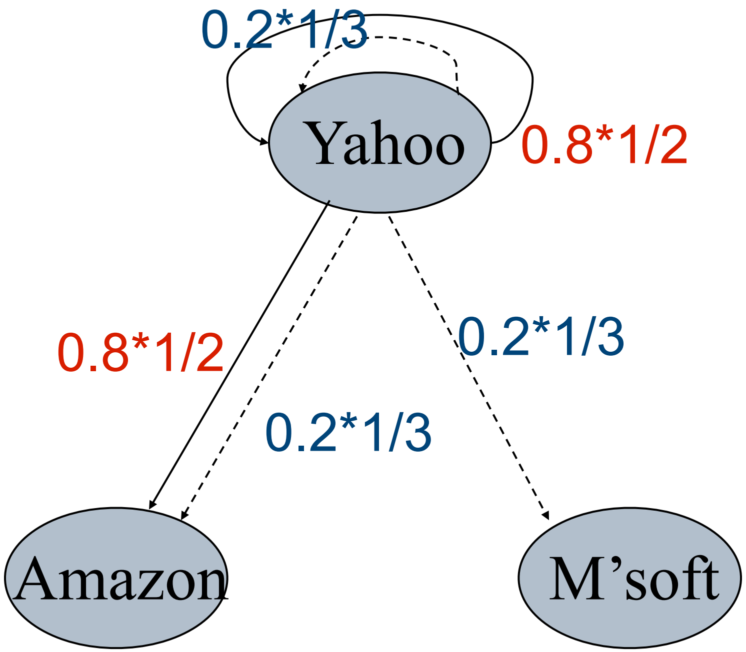


$\longrightarrow$

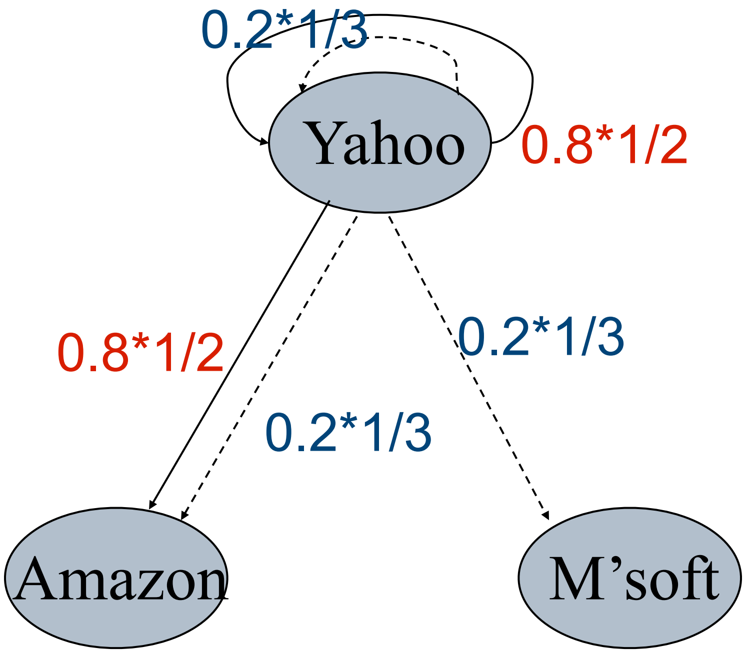
Random teleports ($\beta = 0.8$)



Random teleports ($\beta = 0.8$)

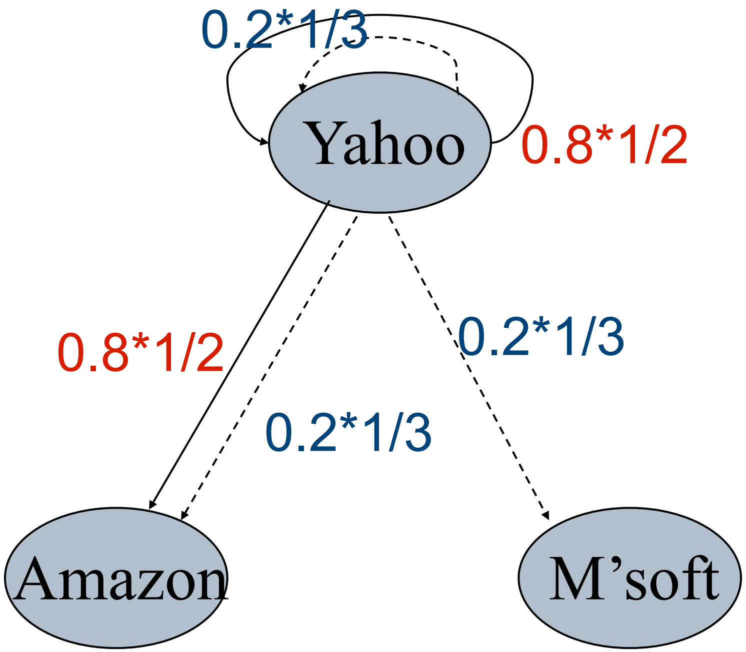


Random teleports ($\beta = 0.8$)



$$\begin{array}{cc} & y \\ \begin{array}{c} y \\ a \\ m \end{array} & \begin{bmatrix} 1/2 \\ 1/2 \\ 0 \end{bmatrix} \end{array} \longrightarrow 0.8* \begin{array}{c} y \\ \begin{bmatrix} 1/2 \\ 1/2 \\ 0 \end{bmatrix} \end{array} + 0.2* \begin{array}{c} y \\ \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \end{array}$$

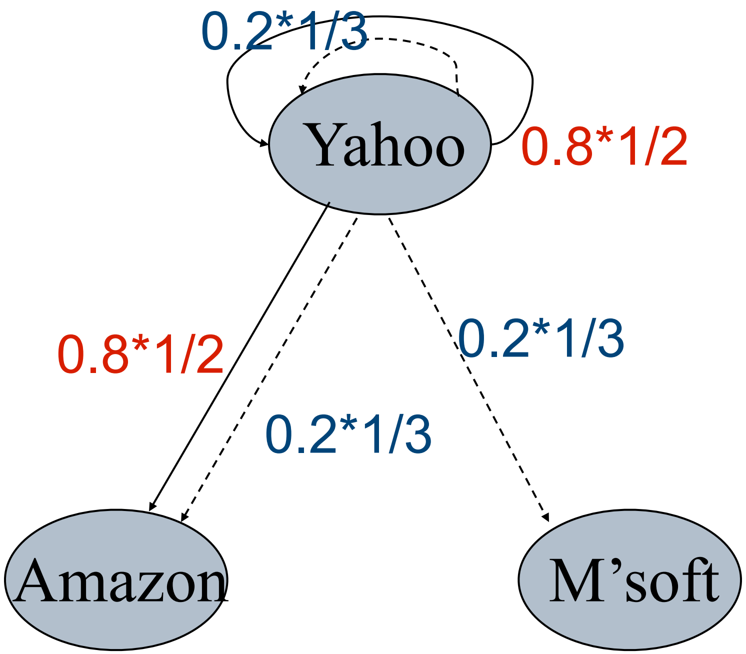
Random teleports ($\beta = 0.8$)



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

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

Random teleports ($\beta = 0.8$)

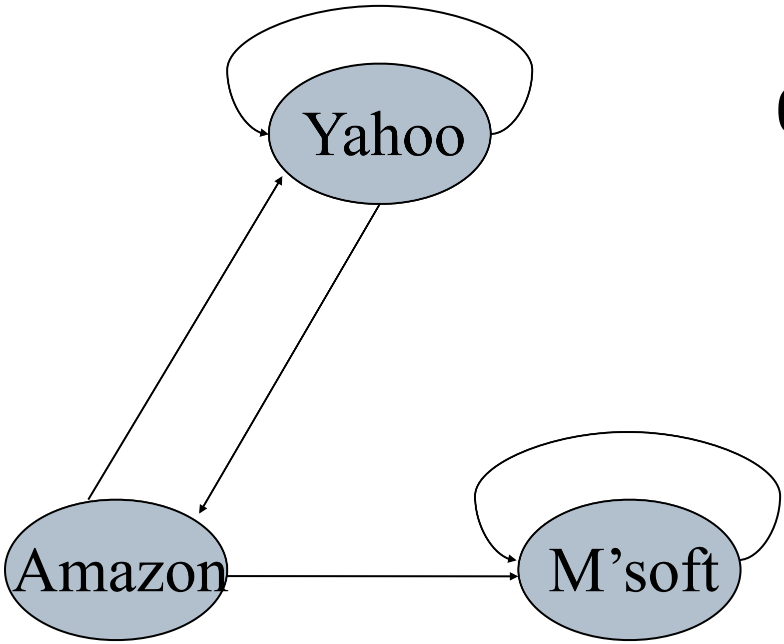


$$\begin{array}{c} \\ y \\ a \\ m \end{array} \begin{array}{c} y \\ \boxed{\begin{array}{c} 1/2 \\ 1/2 \\ 0 \end{array}} \end{array} \longrightarrow 0.8* \begin{array}{c} y \\ \boxed{\begin{array}{c} 1/2 \\ 1/2 \\ 0 \end{array}} \end{array} + 0.2* \begin{array}{c} y \\ \boxed{\begin{array}{c} 1/3 \\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

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

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

Random teleports ($\beta = 0.8$)

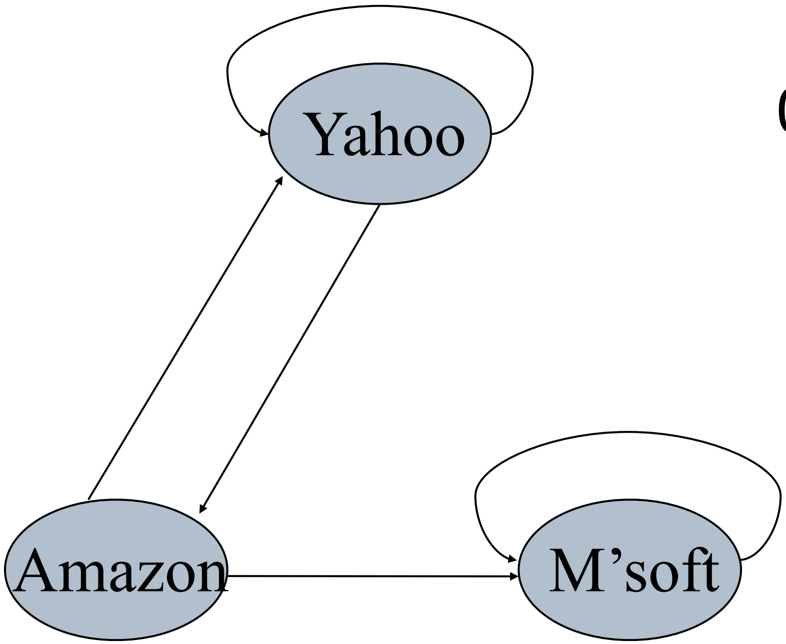


$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

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

$$\begin{array}{ccc} y & & 1/3 \\ a & = & 1/3 \\ m & & 1/3 \end{array}$$

Random teleports ($\beta = 0.8$)

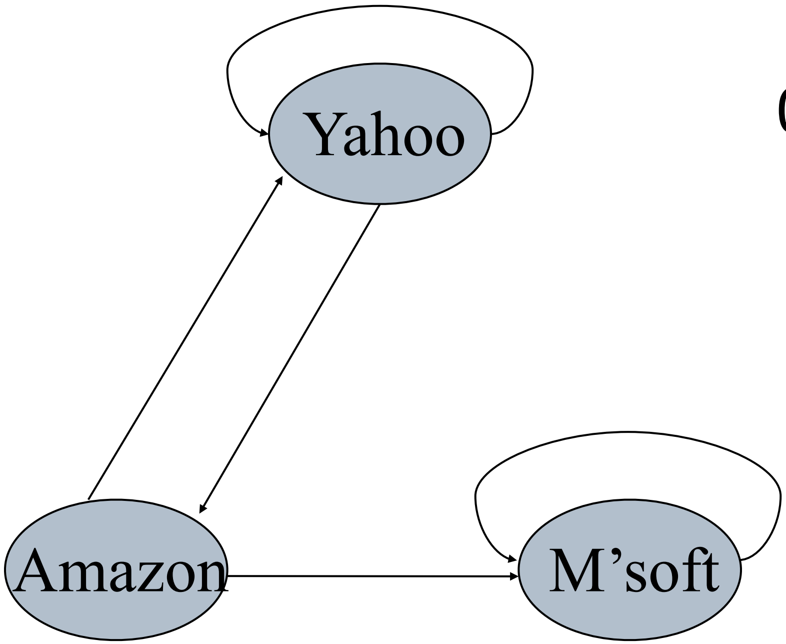


$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

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

$$\begin{array}{ccccc} y & & 1/3 & 1/3 & \\ a & = & 1/3 & 0.20 & \\ m & & 1/3 & 0.47 & \end{array}$$

Random teleports ($\beta = 0.8$)

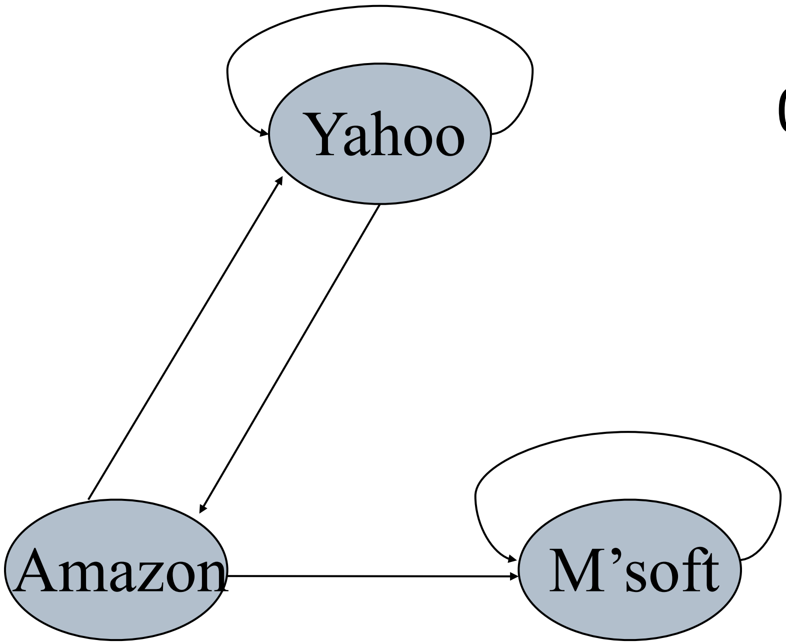


$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

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

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{cccc} 1/3 & 1/3 & 0.27 \\ 1/3 & 0.20 & 0.20 \\ 1/3 & 0.47 & 0.52 \end{array}$$

Random teleports ($\beta = 0.8$)

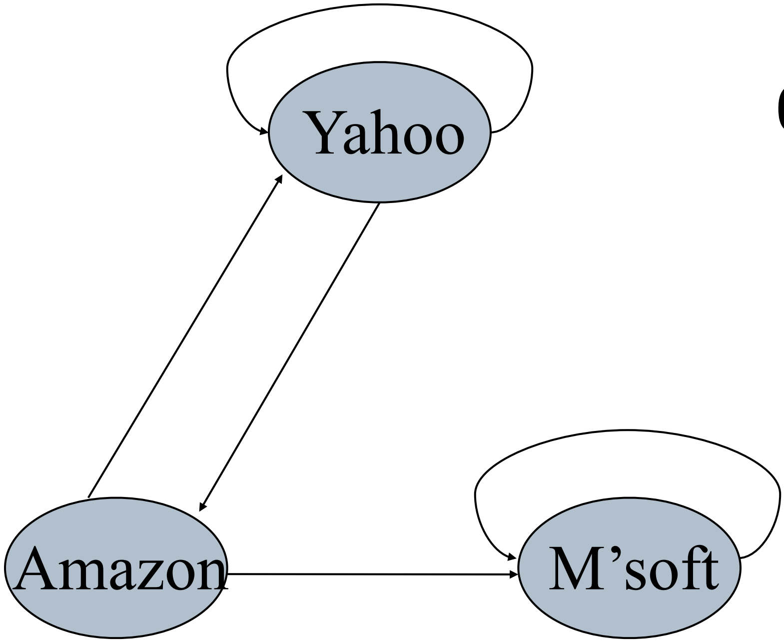


$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

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

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{cccc} 1/3 & 1/3 & 0.27 & 0.258 \\ 1/3 & 0.20 & 0.20 & 0.178 \\ 1/3 & 0.47 & 0.52 & 0.562 \end{array}$$

Random teleports ($\beta = 0.8$)

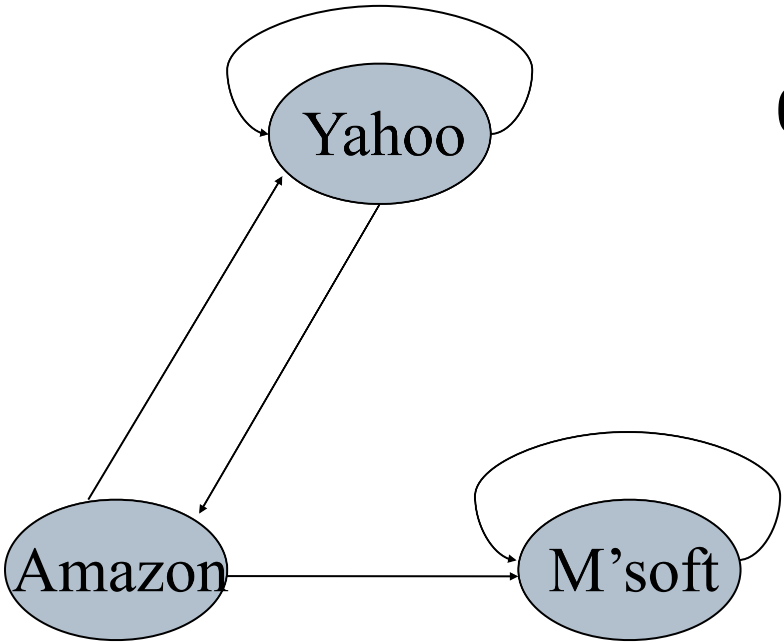


$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

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

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{c} 1/3 \\ 1/3 \\ 1/3 \end{array} \quad \begin{array}{c} 1/3 \\ 0.20 \\ 0.47 \end{array} \quad \begin{array}{c} 0.27 \\ 0.20 \\ 0.52 \end{array} \quad \begin{array}{c} 0.258 \\ 0.178 \\ 0.562 \end{array} \quad \ldots$$

Random teleports ($\beta = 0.8$)



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

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

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{c} 1/3 \\ 1/3 \\ 1/3 \end{array} \quad \begin{array}{c} 1/3 \\ 0.20 \\ 0.47 \end{array} \quad \begin{array}{c} 0.27 \\ 0.20 \\ 0.52 \end{array} \quad \begin{array}{c} 0.258 \\ 0.178 \\ 0.562 \end{array} \quad \ldots \quad \begin{array}{c} 7/33 \\ 5/33 \\ 21/33 \end{array}$$

Page Rank

Page Rank

- Construct the NxN matrix **A** as follows
  - $A_{ij} = \beta M_{ij} + (1-\beta)/N$

Page Rank

- Construct the NxN matrix **A** as follows
  - $A_{ij} = \beta M_{ij} + (1-\beta)/N$
- Verify that **A** is a stochastic matrix

Page Rank

- Construct the NxN matrix **A** as follows
  - $A_{ij} = \beta M_{ij} + (1-\beta)/N$
- Verify that **A** is a stochastic matrix
- The page rank vector **r** is the principal eigenvector of this matrix
  - satisfying $\mathbf{r} = \mathbf{A}\mathbf{r}$

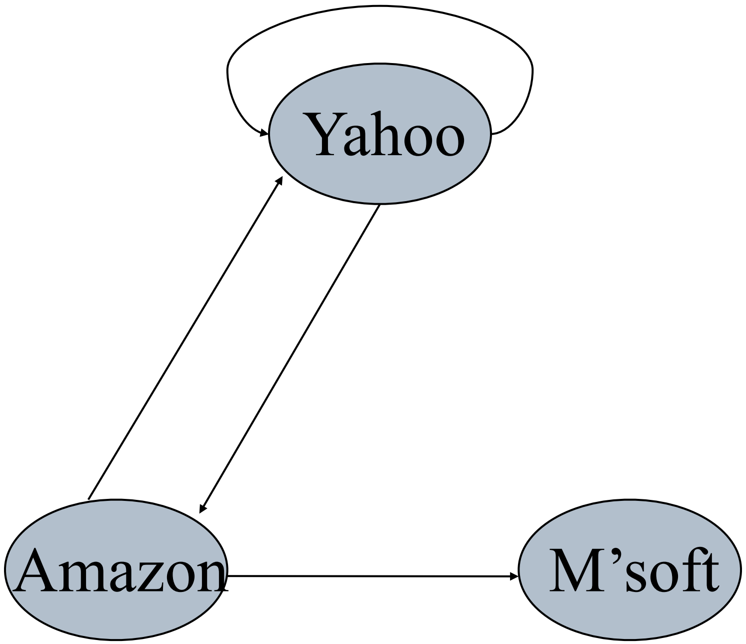
Page Rank

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

Dead ends

- The description of the PageRank algorithm is essentially complete. Minor problem with "dead ends".
- Pages with no outlinks are "dead ends" for the random surfer -> Nowhere to go in the next step.
- Our algorithm so far is not well-defined when the number of successors k=0 (we would have 1/0!).

Microsoft becomes a dead end

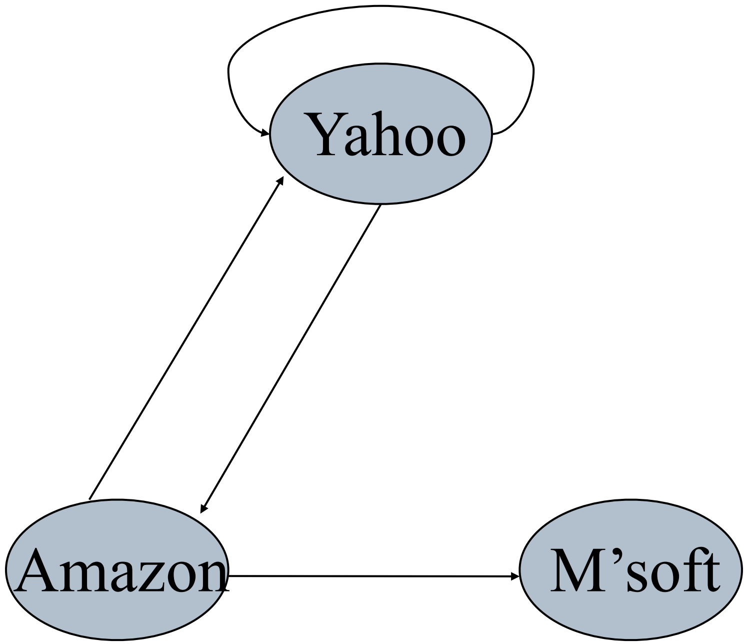


$$0.8 \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \end{bmatrix} + 0.2 \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

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

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{c} 1/3 \\ 1/3 \\ 1/3 \end{array}$$

Microsoft becomes a dead end



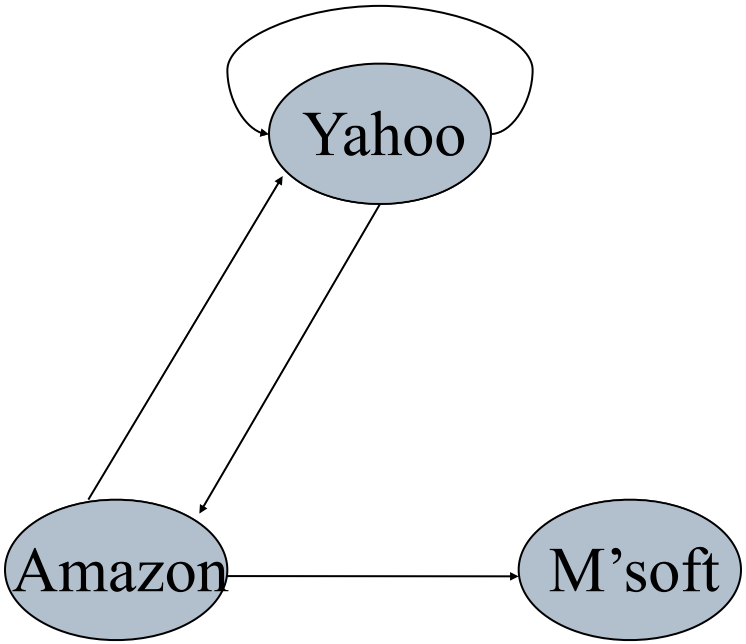
$$0.8\begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \end{bmatrix} + 0.2\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

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

Non-stochastic!

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{c} 1/3 \\ 1/3 \\ 1/3 \end{array}$$

Microsoft becomes a dead end



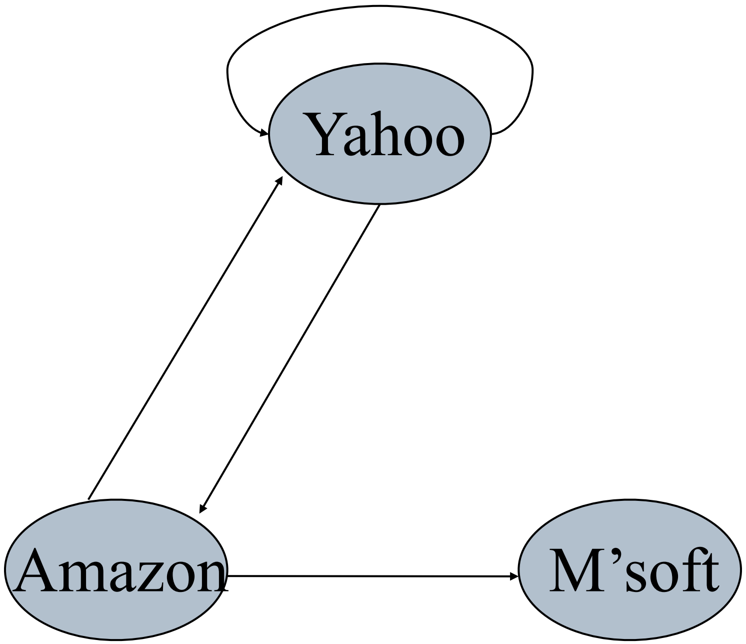
$$0.8\begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \end{bmatrix} + 0.2\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}$$

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

Non-stochastic!

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{c} 1/3 \\ 1/3 \\ 1/3 \end{array} \quad \ldots$$

Microsoft becomes a dead end



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

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

Non-stochastic!

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{c} 1/3 \\ 1/3 \\ 1/3 \end{array} \quad \cdots \quad \begin{array}{c} 0 \\ 0 \\ 0 \end{array}$$

Dealing with dead-ends

Dealing with dead-ends

- Teleport
  - Follow random teleport links with probability 1.0 from dead-ends
  - Adjust matrix accordingly

Dealing with dead-ends

- Teleport
  - Follow random teleport links with probability 1.0 from dead-ends
  - Adjust matrix accordingly
- More efficient: 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

Efficiency issues

Efficiency issues

- Key step is matrix-vector multiplication
  - $\mathbf{r}^{new} = \mathbf{A}\mathbf{r}^{old}$

Efficiency issues

- Key step is matrix-vector multiplication
  - $\mathbf{r}^{new} = \mathbf{A}\mathbf{r}^{old}$
- Easy if we have enough main memory to hold $\mathbf{A}$, $\mathbf{r}^{old}$, $\mathbf{r}^{new}$

Efficiency issues

- Key step is matrix-vector multiplication
  - $\mathbf{r}^{new} = \mathbf{A}\mathbf{r}^{old}$
- Easy if we have enough main memory to hold $\mathbf{A}$, $\mathbf{r}^{old}$, $\mathbf{r}^{new}$
- Say N = 1 billion pages
  - Matrix A has N^2 entries
    - 10^{18} is a large number!

Rearranging the equation

Rearranging the equation

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

Rearranging the equation

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

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

Rearranging the equation

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

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

$r_i = \sum_{1 \leq j \leq N} A_{ij}\, r_j$

Rearranging the equation

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

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

$r_i = \sum_{1 \leq j \leq N} A_{ij} r_j$

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

Rearranging the equation

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

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

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

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

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

Rearranging the equation

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

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

$r_i = \sum_{1 \leq j \leq N} A_{ij} r_j$

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

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

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

Rearranging the equation

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

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

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

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

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

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

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

Rearranging the equation

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

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

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

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

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

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

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

where $[x]_N$ is a vector with N entries equal to x

Sparse matrix formulation

Sparse matrix formulation

- We can rearrange the page rank equation:
  - $\mathbf{r} = \beta\mathbf{M}\mathbf{r} + [(1-\beta)/N]_N$
  - $[(1-\beta)/N]_N$ is an N-vector with all entries $(1-\beta)/N$

Sparse matrix formulation

- We can rearrange the page rank equation:
  - $\mathbf{r} = \beta \mathbf{M} \mathbf{r} + [(1-\beta)/N]_N$
  - $[(1-\beta)/N]_N$ is an N-vector with all entries $(1-\beta)/N$
- **M** is a sparse matrix!
  - 10 links per node, approx 10N entries

Sparse matrix formulation

- We can rearrange the page rank equation:
  - $\mathbf{r} = \beta \mathbf{M}\mathbf{r} + [(1-\beta)/N]_N$
  - $[(1-\beta)/N]_N$ is an N-vector with all entries $(1-\beta)/N$
- **M** is a sparse matrix!
  - 10 links per node, approx 10N entries
- So in each iteration, we need to:
  - Compute $\mathbf{r}^{new} = \beta \mathbf{M}\mathbf{r}^{old}$
  - Add a constant value $(1-\beta)/N$ to each entry in $\mathbf{r}^{new}$

Sparse matrix encoding

- Encode sparse matrix using only nonzero entries
  - Space proportional roughly to number of links
  - say 10N, or 4*10*1 billion = 40GB
  - still won’t fit in memory, but will fit on disk

source node	dest. node	probability
0	1	1/4
0	5	1/4
2	17	1/12

PageRank: summary

PageRank: summary

- Remove iteratively dead ends from G

PageRank: summary

- Remove iteratively dead ends from G
- Build stochastic matrix M_G (M for short)

PageRank: summary

- Remove iteratively dead ends from G
- Build stochastic matrix M_G (M for short)
- Initialize: $\mathbf{r}^0 = [1/N, \ldots, 1/N]^T$

PageRank: summary

- Remove iteratively dead ends from G
- Build stochastic matrix M_G (M for short)
- Initialize: $\mathbf{r}^0 = [1/N,\ldots.,1/N]^T$
- Iterate:
  - $\mathbf{r}^{\mathbf{k+1}} = \beta \mathbf{M}\mathbf{r}^{\mathbf{k}} + [(1-\beta)/N]_N$
  - Stop when $|\mathbf{r}^{k+1} - \mathbf{r}^k|_1 < \varepsilon$