Evaluating the Web

PageRank

Hubs and Authorities

PageRank

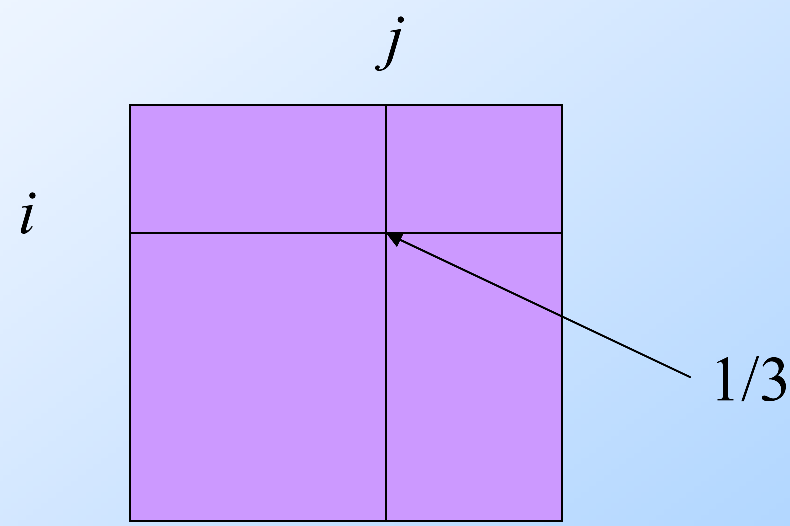
- Intuition: solve the recursive equation: "a page is important if important pages link to it."
- In high-falutin' terms: *importance* = the principal eigenvector of the stochastic matrix of the Web.
  - A few fixups needed.

Stochastic Matrix of the Web

- Enumerate pages.
- Page i corresponds to row and column i.
- $M[i,j] = 1/n$ if page j links to n pages, including page i; 0 if j does not link to i.
  - $M[i,j]$ is the probability we'll next be at page i if we are now at page j.

Example

Suppose page j links to 3 pages, including i



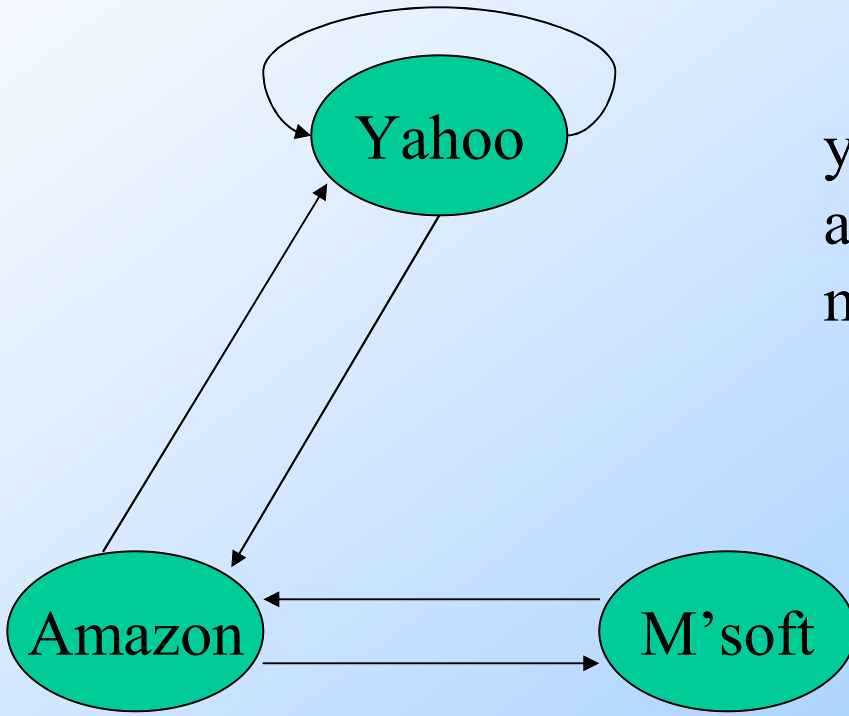
Random Walks on the Web

- Suppose **v** is a vector whose i th component is the probability that we are at page i at a certain time.
- If we follow a link from i at random, the probability distribution for the page we are then at is given by the vector M **v**.

Random Walks --- (2)

- Starting from any vector **v**, the limit $M(M(\ldots M(M\mathbf{v})\ldots))$ is the distribution of page visits during a random walk.
- Intuition: pages are important in proportion to how often a random walker would visit them.
- The math: limiting distribution = principal eigenvector of M = PageRank.

Example: The Web in 1839



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

Simulating a Random Walk

- Start with the vector **v** = [1,1,…,1] representing the idea that each Web page is given one unit of *importance*.
- Repeatedly apply the matrix M to **v**, allowing the importance to flow like a random walk.
- Limit exists, but about 50 iterations is sufficient to estimate final distribution.

Example

- Equations $\mathbf{v} = M\mathbf{v}$:

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

$$a = y/2 + m$$

$$m = a/2$$

y		1	1	5/4	9/8		6/5
a	=	1	3/2	1	11/8	. . .	6/5
m		1	1/2	3/4	1/2		3/5

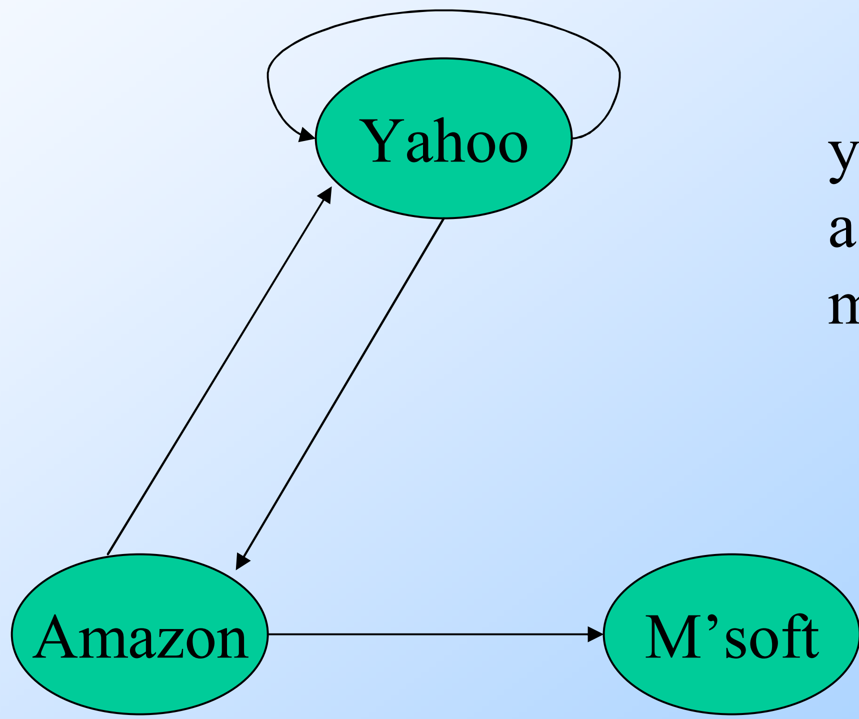
Solving The Equations

- Because there are no constant terms, these 3 equations in 3 unknowns do not have a unique solution.
- Add in the fact that $y+a+m = 3$ to solve.
- In Web-sized examples, we cannot solve by Gaussian elimination; we need to use *relaxation* (= iterative solution).

Real-World Problems

- Some pages are "dead ends" (have no links out).
  - Such a page causes importance to leak out.
- Other (groups of) pages are *spider traps* (all out-links are within the group).
  - Eventually spider traps absorb all importance.

Microsoft Becomes Dead End



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

Example

- Equations $\mathbf{v} = M\mathbf{v}$:

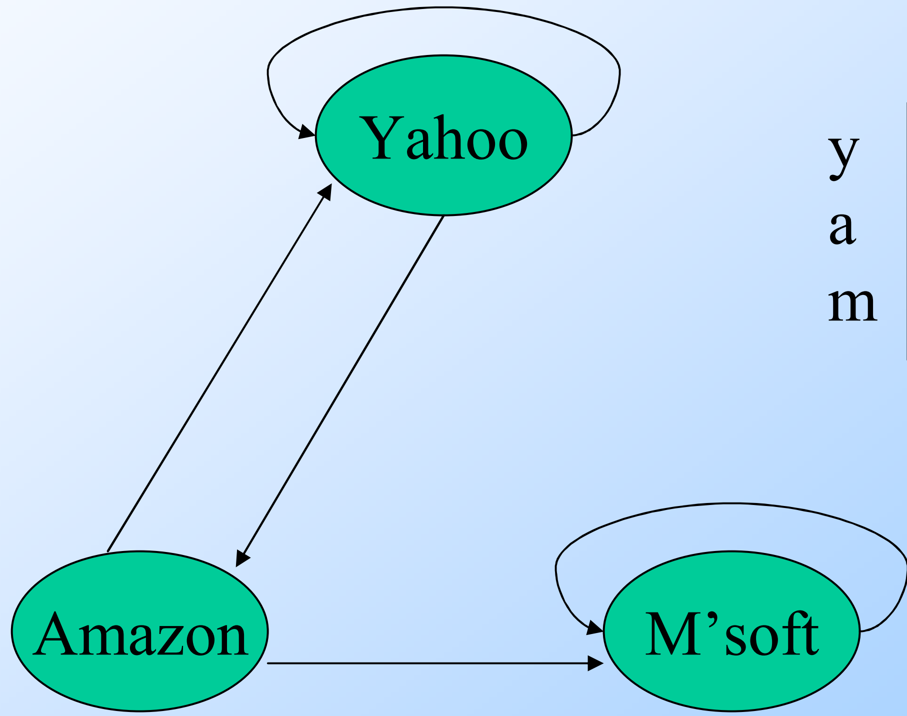
  $y = y/2 + a/2$

  $a = y/2$

  $m = a/2$

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

M’soft Becomes Spider Trap



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

Example

- Equations $\mathbf{v} = M\mathbf{v}$:

$y = y/2 + a/2$

$a = y/2$

$m = a/2 + m$

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

Google Solution to Traps, Etc.

- "Tax" each page a fixed percentage at each interation.
- Add the same constant to all pages.
- Models a random walk with a fixed probability of going to a random place next.

Example: Previous with 20% Tax

- Equations $\mathbf{v} = 0.8(M\mathbf{v}) + 0.2$:

$$y = 0.8(y/2 + a/2) + 0.2$$

$$a = 0.8(y/2) + 0.2$$

$$m = 0.8(a/2 + m) + 0.2$$

$$\begin{array}{l} y \\ a \\ m \end{array} = \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \quad \begin{array}{c} 1.00 \\ 0.60 \\ 1.40 \end{array} \quad \begin{array}{c} 0.84 \\ 0.60 \\ 1.56 \end{array} \quad \begin{array}{c} 0.776 \\ 0.536 \\ 1.688 \end{array} \quad \cdots \quad \begin{array}{r} 7/11 \\ 5/11 \\ 21/11 \end{array}$$

General Case

- In this example, because there are no dead-ends, the total importance remains at 3.
- In examples with dead-ends, some importance leaks out, but total remains finite.

Solving the Equations

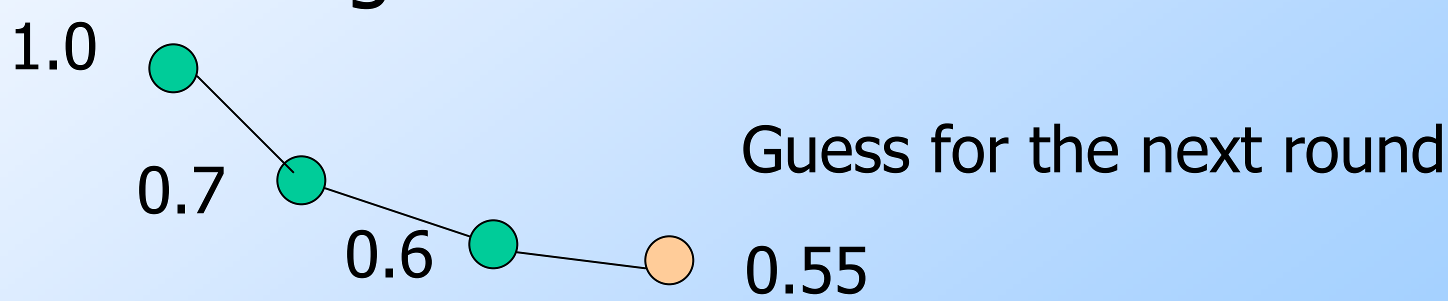
- Because there are constant terms, we can expect to solve small examples by Gaussian elimination.
- Web-sized examples still need to be solved by relaxation.

Speeding Convergence

- Newton-like prediction of where components of the principal eigenvector are heading.
- Take advantage of locality in the Web.
- Each technique can reduce the number of iterations by 50%.
  - Important --- PageRank takes time!

Predicting Component Values

- Three consecutive values for the importance of a page suggests where the limit might be.



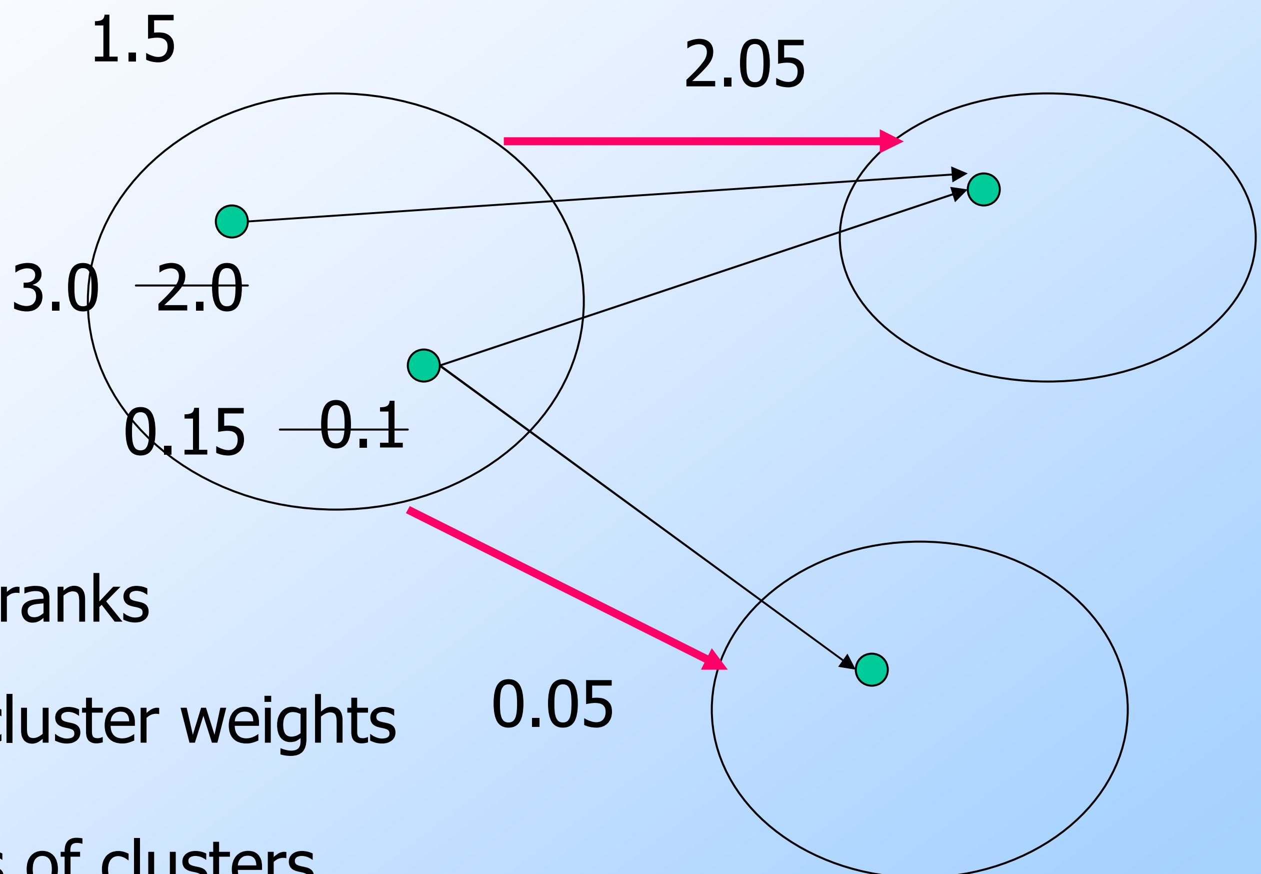
Exploiting Substructure

- Pages from particular domains, hosts, or paths, like `stanford.edu` or `www-db.stanford.edu/~ullman` tend to have higher density of links.
- Initialize PageRank using ranks within your local cluster, then ranking the clusters themselves.

Strategy

- Compute local PageRanks (in parallel?).
- Use local weights to establish intercluster weights on edges.
- Compute PageRank on graph of clusters.
- Initial rank of a page is the product of its local rank and the rank of its cluster.
- “Clusters” are appropriately sized regions with common domain or lower-level detail.

In Pictures



Local ranks

Intercluster weights

Ranks of clusters

Initial eigenvector

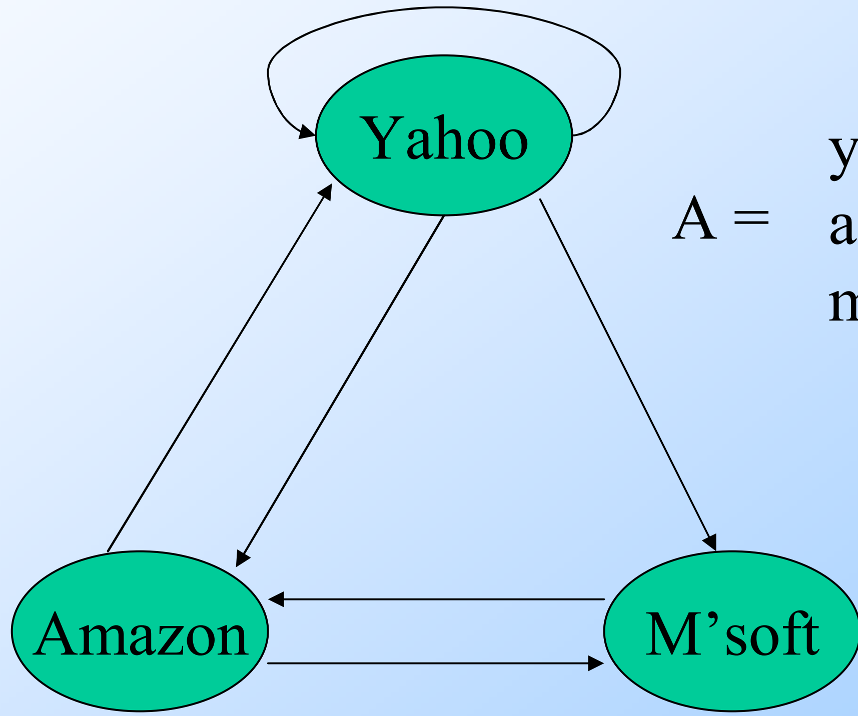
Hubs and Authorities

- Mutually recursive definition:
  - A *hub* links to many authorities;
  - An *authority* is linked to by many hubs.
- Authorities turn out to be places where information can be found.
  - Example: course home pages.
- Hubs tell where the authorities are.
  - Example: CSD course-listing page.

Transition Matrix A

- H&A uses a matrix $A[i,j] = 1$ if page i links to page j, 0 if not.
- A^T, the transpose of A, is similar to the PageRank matrix M, but A^T has 1's where M has fractions.

Example



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

Using Matrix A for H&A

- Powers of A and A^T diverge in size of elements, so we need scale factors.
- Let **h** and **a** be vectors measuring the "hubbiness" and authority of each page.
- Equations: $\mathbf{h} = \lambda A\mathbf{a}$; $\mathbf{a} = \mu A^T\mathbf{h}$.
  - Hubbiness = scaled sum of authorities of linked pages.
  - Authority = scaled sum of hubbiness of predecessor pages.

Consequences of Basic Equations

- From $\mathbf{h} = \lambda A\mathbf{a}$; $\mathbf{a} = \mu A^T \mathbf{h}$ we can derive:
  - $\mathbf{h} = \lambda\mu A A^T \mathbf{h}$
  - $\mathbf{a} = \lambda\mu A^T A\, \mathbf{a}$
- Compute **h** and **a** by iteration, assuming initially each page has one unit of hubbiness and one unit of authority.
  - Pick an appropriate value of λμ.

Example

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad A^T = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix} \qquad AA^T = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 2 & 0 \\ 1 & 0 & 1 \end{bmatrix} \qquad A^TA = \begin{bmatrix} 2 & 1 & 2 \\ 1 & 2 & 1 \\ 2 & 1 & 2 \end{bmatrix}$$

a(yahoo)	=	1	5	24	114	. . .	$1+\sqrt{3}$
a(amazon)	=	1	4	18	84	. . .	2
a(m'soft)	=	1	5	24	114	. . .	$1+\sqrt{3}$

h(yahoo)	=	1	6	28	132	. . .	1.000
h(amazon)	=	1	4	20	96	. . .	0.735
h(m'soft)	=	1	2	8	36	. . .	0.268

Solving the Equations

- Solution of even small examples is tricky, because the value of λμ is one of the unknowns.
  - Each equation like $y = \lambda\mu(3y + 2a + m)$ lets us solve for λμ in terms of y, a, m; equate each expression for λμ.
- As for PageRank, we need to solve big examples by relaxation.

Details for **h** --- (1)

$$y = \lambda\mu(3y + 2a + m)$$

$$a = \lambda\mu(2y + 2a)$$

$$m = \lambda\mu(y + m)$$

- Solve for $\lambda\mu$:

$$\lambda\mu = y/(3y + 2a + m) = a/(2y + 2a) = m/(y + m)$$

Details for **h** --- (2)

- Assume $y = 1$.

$\lambda\mu = 1/(3 + 2a + m) = a/(2 + 2a) =$
$m/(1+m)$

- Cross-multiply second and third:

$a + am = 2m + 2am$ or $a = 2m/(1-m)$

- Cross multiply first and third:

$1+m = 3m + 2am + m^2$ or $a = (1-2m - m^2)/2m$

Details for **h** --- (3)

- Equate formulas for a:

$a = 2m/(1-m) = (1-2m-m^2)/2m$

- Cross-multiply:

$1 - 2m - m^2 - m + 2m^2 + m^3 = 4m^2$

- Solve for m: $m = .268$
- Solve for a: $a = 2m/(1-m) = .735$

Solving H&A in Practice

- Iterate as for PageRank; don't try to solve equations.
- But keep the scale of values within bounds.
  - Example: scale to keep the largest component of the vector at 1.

H&A Versus PageRank

- If you talk to someone from IBM, they will tell you "IBM invented PageRank."
  - What they mean is that H&A was invented by Jon Kleinberg when he was at IBM.
- But these are not the same.
- H&A has been used, e.g., to analyze important research papers.