Evaluating the Web

PageRank
Hubs and Authorities

1

Page Rank

- 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.

2

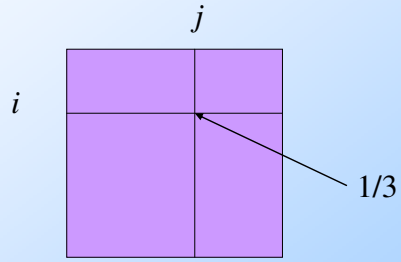
Stochastic Matrix of the Web

- 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*.
  - Seems backwards, but allows multiplication by *M* on the left to represent "follow a link."

3

Example

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



4

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 Mv.

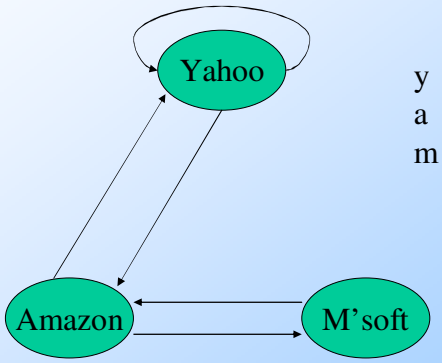
5

Random Walks --- (2)

- Starting from any vector v, the limit $M(M(\ldots M(Mv)\ \ldots))$ 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.

6

Example: The Web in 1839



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

7

Simulating a Random Walk

- Start with the vector $v = [1,1,\ldots,1]$ 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.

8

Example

- Equations $v = Mv$:
  - $y = y/2 + a/2$
  - $a = y/2 + m$
  - $m = a/2$

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

9

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).

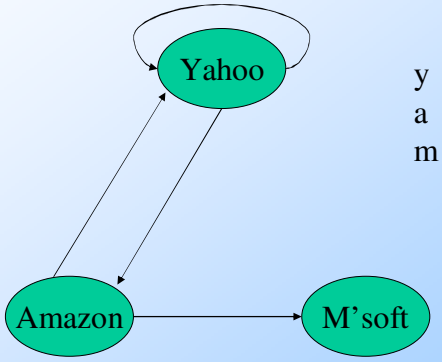
10

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.

11

Microsoft Becomes Dead End



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

12

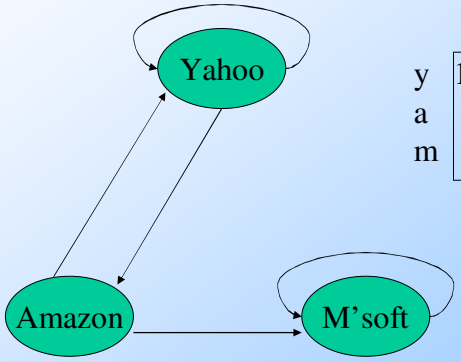
Example

- Equations $v = Mv$:
  - $y = y/2 + a/2$
  - $a = y/2$
  - $m = a/2$

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

13

M'soft Becomes Spider Trap



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

14

Example

- Equations $v = Mv$:
  - $y = y/2 + a/2$
  - $a = y/2$
  - $m = a/2 + m$

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

15

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 in which surfer has a fixed probability of abandoning search and going to a random page next.

16

Ex: Previous with 20% Tax

- Equations $v = 0.8(Mv) + 0.2$:
  - $y = 0.8(y/2 + a/2) + 0.2$
  - $a = 0.8(y/2) + 0.2$
  - $m = 0.8(a/2 + m) + 0.2$

y		1	1.00	0.84	0.776	7/11
a	=	1	0.60	0.60	0.536 . . .	5/11
m		1	1.40	1.56	1.688	21/11

17

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.

18

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.

19

Search-Engine Architecture

- All search engines, including Google, select pages that have the words of your query.
- Give more weight to the word appearing in the title, header, etc.
- Inverted indexes speed the discovery of pages with given words.

20

Google Anti-Spam Devices

- Early search engines relied on the words on a page to tell what it is about.
  - Led to "tricks" in which pages attracted attention by placing false words in the background color on their page.
- Google trusts the words in anchor text
  - Relies on others telling the truth about your page, rather than relying on you.

21

Use of Page Rank

- Pages are ordered by many criteria, including the PageRank and the appearance of query words.
  - "Important" pages more likely to be what you want.
- PageRank is also an antispam device.
  - Creating bogus links to yourself doesn't help if you are not an important page.

22

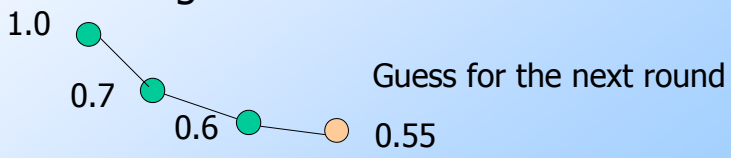
Speeding Convergence

- 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 can take days.

23

Predicting Component Values

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



24

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.

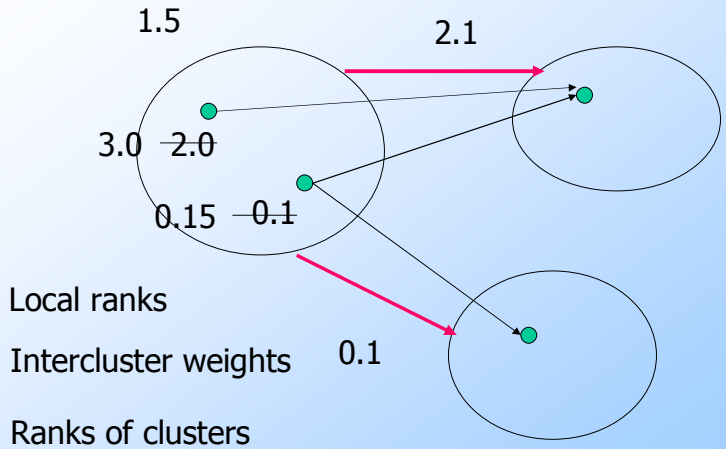
25

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.

26

In Pictures



Local ranks

Intercluster weights

Ranks of clusters

Initial eigenvector

27

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.

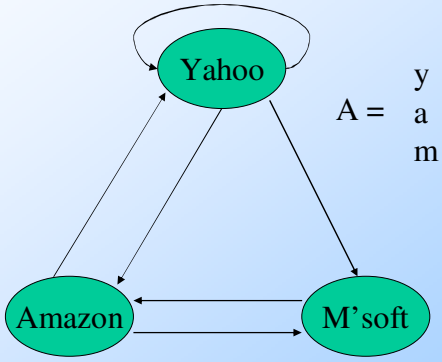
28

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.

29

Example



A =

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

30