CS345
Data Mining

Link Analysis 2:
Topic-Specific Page Rank
Hubs and Authorities
Spam Detection

Anand Rajaraman, Jeffrey D. Ullman

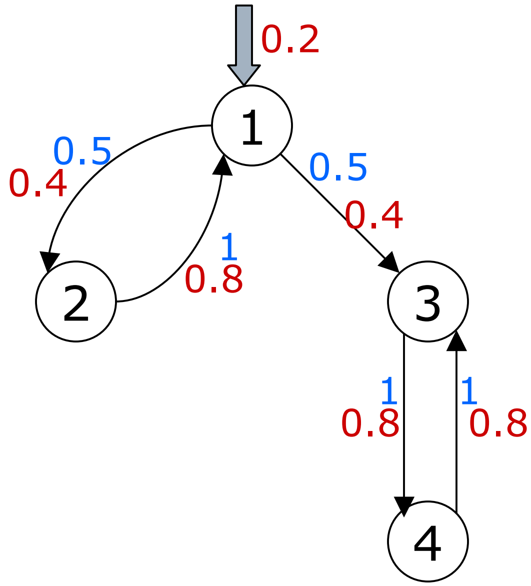
Topic-Specific Page Rank

- Instead of generic popularity, can we measure popularity within a topic?
  - E.g., computer science, health
- Bias the random walk
  - When the random walker teleports, he picks a page from a set S of web pages
  - S contains only pages that are relevant to the topic
  - E.g., Open Directory (DMOZ) pages for a given topic (www.dmoz.org)
- For each teleport set S, we get a different rank vector $\mathbf{r_S}$

Matrix formulation

- $A_{ij} = \beta M_{ij} + (1-\beta)/|S|$ if i 2 S
- $A_{ij} = \beta M_{ij}$ otherwise
- Show that **A** is stochastic
- We have weighted all pages in the teleport set S equally
  - Could also assign different weights to them

Example



Suppose S = {1}, β = 0.8

Node	**Iteration**			
	0	1	2…	stable
1	1.0	0.2	0.52	0.294
2	0	0.4	0.08	0.118
3	0	0.4	0.08	0.327
4	0	0	0.32	0.261

Note how we initialize the page rank vector differently from the unbiased page rank case.

How well does TSPR work?

- Experimental results [Haveliwala 2000]
- Picked 16 topics
  - Teleport sets determined using DMOZ
  - E.g., arts, business, sports,…
- “Blind study” using volunteers
  - 35 test queries
  - Results ranked using Page Rank and TSPR of most closely related topic
  - E.g., bicycling using Sports ranking
  - In most cases volunteers preferred TSPR ranking

Which topic ranking to use?

- User can pick from a menu
- Use Bayesian classification schemes to classify query into a topic
- Can use the context of the query
  - E.g., query is launched from a web page talking about a known topic
  - History of queries e.g., “basketball” followed by “jordan”
- User context e.g., user’s My Yahoo settings, bookmarks, …

Hubs and Authorities

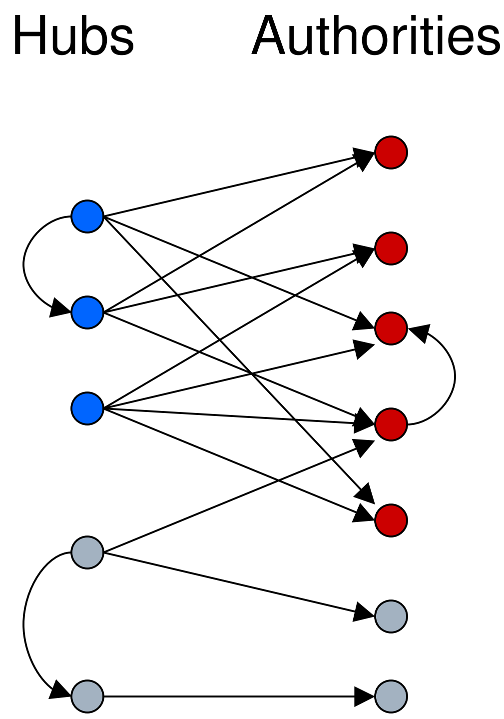
- Suppose we are given a collection of documents on some broad topic
  - e.g., stanford, evolution, iraq
  - perhaps obtained through a text search
- Can we organize these documents in some manner?
  - Page rank offers one solution
  - HITS (Hypertext-Induced Topic Selection) is another
    - proposed at approx the same time (1998)

HITS Model

- Interesting documents fall into two classes

1. Authorities are pages containing useful information
   - course home pages
   - home pages of auto manufacturers
2. Hubs are pages that link to authorities
   - course bulletin
   - list of US auto manufacturers

Idealized view



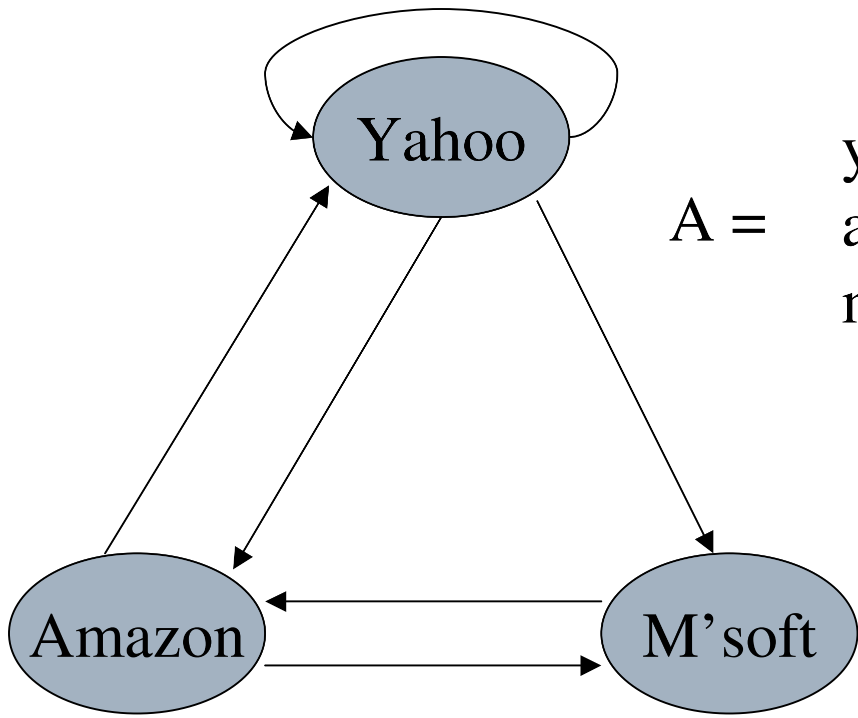
Mutually recursive definition

- A good hub links to many good authorities
- A good authority is linked from many good hubs
- Model using two scores for each node
  - Hub score and Authority score
  - Represented as vectors **h** and **a**

Transition Matrix A

- HITS 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



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

Hub and Authority Equations

- The hub score of page P is proportional to the sum of the authority scores of the pages it links to
  - $\mathbf{h} = \lambda A\mathbf{a}$
  - Constant λ is a scale factor
- The authority score of page P is proportional to the sum of the hub scores of the pages it is linked from
  - $\mathbf{a} = \mu A^T \mathbf{h}$
  - Constant μ is scale factor

Iterative algorithm

- Initialize **h**, **a** to all 1's
- $\mathbf{h} = \mathbf{A}\mathbf{a}$
- Scale **h** so that its max entry is 1.0
- $\mathbf{a} = \mathbf{A}^T\mathbf{h}$
- Scale **a** so that its max entry is 1.0
- Continue until **h**, **a** converge

Example

$$A = \boxed{\begin{matrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{matrix}} \qquad A^T = \boxed{\begin{matrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{matrix}}$$

a(yahoo)	=	1	1	1	1	. . .	1
a(amazon)	=	1	1	4/5	0.75	. . .	0.732
a(m'soft)	=	1	1	1	1	. . .	1

h(yahoo)	=	1	1	1	1	. . .	1.000
h(amazon)	=	1	2/3	0.71	0.73	. . .	0.732
h(m'soft)	=	1	1/3	0.29	0.27	. . .	0.268

Existence and Uniqueness

$\mathbf{h} = \lambda A\mathbf{a}$

$\mathbf{a} = \mu A^T \mathbf{h}$

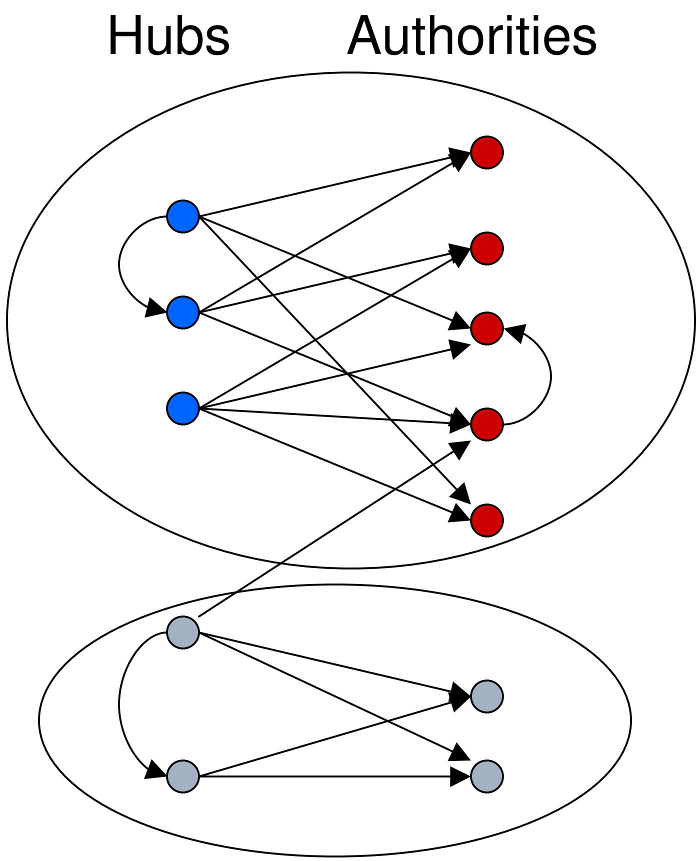
$\mathbf{h} = \lambda\mu AA^T \mathbf{h}$

$\mathbf{a} = \lambda\mu A^TA\, \mathbf{a}$

Under reasonable assumptions about **A**,
the dual iterative algorithm converges to vectors
h* and **a*** such that:

- **h*** is the principal eigenvector of the matrix AA^T
- **a*** is the principal eigenvector of the matrix A^TA

Bipartite cores



Most densely-connected core
(primary core)

Less densely-connected core
(secondary core)

Secondary cores

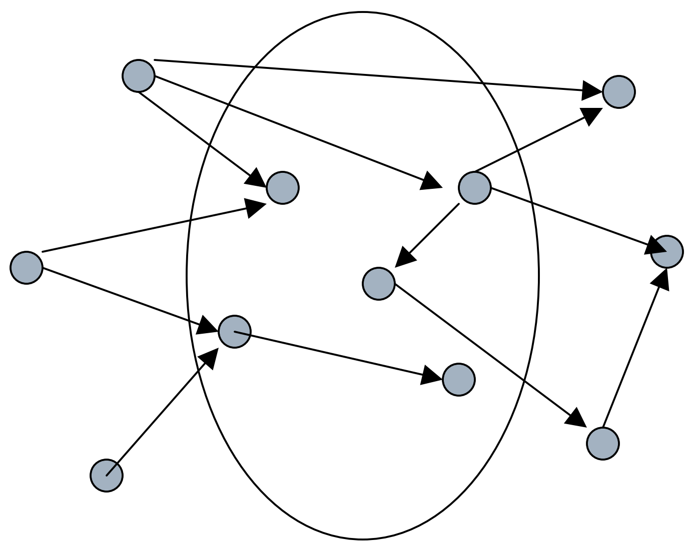
- A single topic can have many bipartite cores
  - corresponding to different meanings, or points of view
  - abortion: pro-choice, pro-life
  - evolution: darwinian, intelligent design
  - jaguar: auto, Mac, NFL team, *panthera onca*
- How to find such secondary cores?

Finding secondary cores

- Once we find the primary core, we can remove its links from the graph
- Repeat HITS algorithm on residual graph to find the next bipartite core
- Roughly, correspond to non-primary eigenvectors of AA^T and A^TA

Creating the graph for HITS

- We need a well-connected graph of pages for HITS to work well



Page Rank and HITS

- Page Rank and HITS are two solutions to the same problem
  - What is the value of an inlink from S to D?
  - In the page rank model, the value of the link depends on the links **into** S
  - In the HITS model, it depends on the value of the other links **out of** S
- The destinies of Page Rank and HITS post-1998 were very different
  - Why?

Web Spam

- Search has become the default gateway to the web
- Very high premium to appear on the first page of search results
  - e.g., e-commerce sites
  - advertising-driven sites

What is web spam?

- Spamming = any deliberate action solely in order to boost a web page’s position in search engine results, incommensurate with page’s real value
- Spam = web pages that are the result of spamming
- This is a very broad defintion
  - SEO industry might disagree!
  - SEO = search engine optimization
- Approximately 10-15% of web pages are spam

Web Spam Taxonomy

- We follow the treatment by Gyongyi and Garcia-Molina [2004]
- Boosting techniques
  - Techniques for achieving high relevance/importance for a web page
- Hiding techniques
  - Techniques to hide the use of boosting
    - From humans and web crawlers

Boosting techniques

- Term spamming
  - Manipulating the text of web pages in order to appear relevant to queries
- Link spamming
  - Creating link structures that boost page rank or hubs and authorities scores

Term Spamming

- Repetition
  - of one or a few specific terms e.g., free, cheap, viagra
  - Goal is to subvert TF.IDF ranking schemes
- Dumping
  - of a large number of unrelated terms
  - e.g., copy entire dictionaries
- Weaving
  - Copy legitimate pages and insert spam terms at random positions
- Phrase Stitching
  - Glue together sentences and phrases from different sources

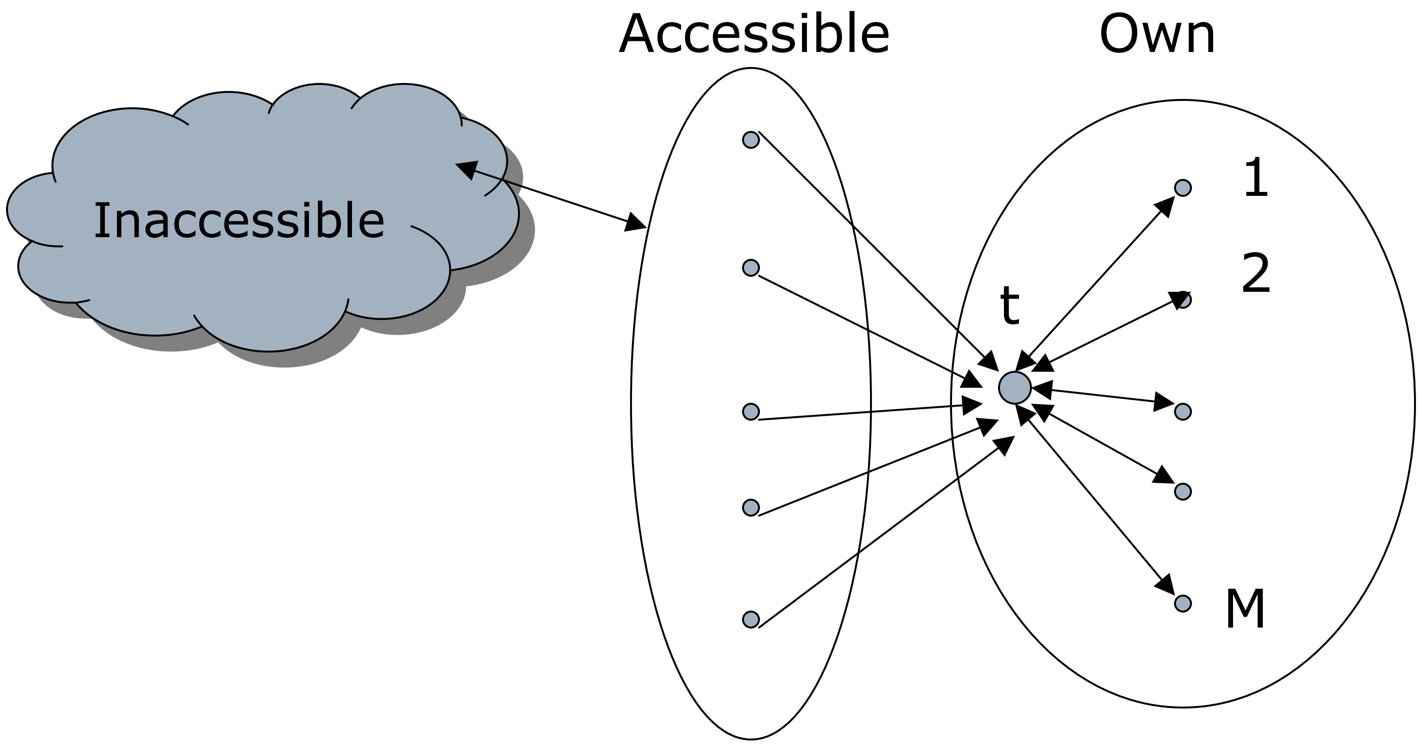
Link spam

- Three kinds of web pages from a spammer’s point of view
  - Inaccessible pages
  - Accessible pages
    - e.g., blog comments pages
    - spammer can post links to his pages
  - Own pages
    - Completely controlled by spammer
    - May span multiple domain names

Link Farms

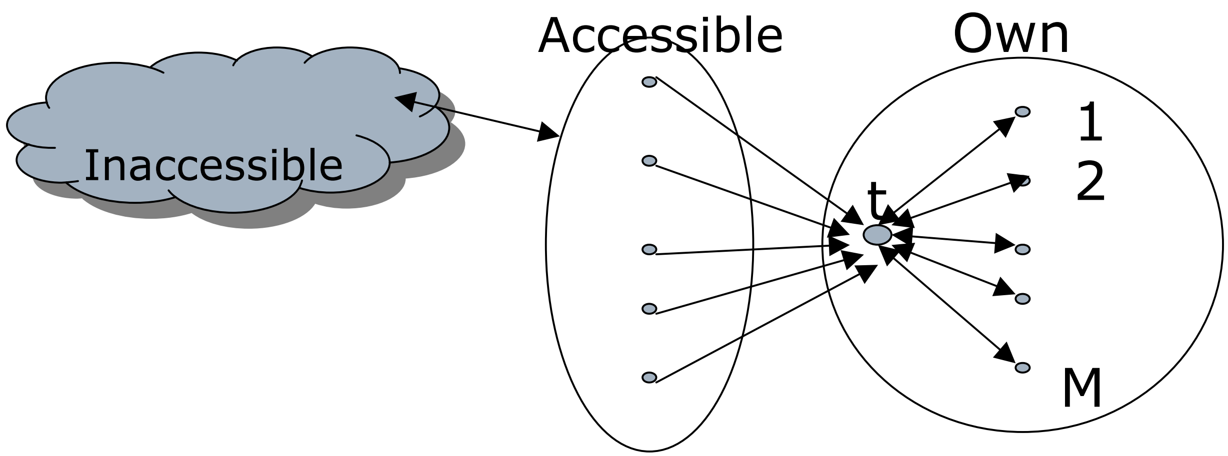
- Spammer’s goal
  - Maximize the page rank of target page t
- Technique
  - Get as many links from accessible pages as possible to target page t
  - Construct “link farm” to get page rank multiplier effect

Link Farms



One of the most common and effective organizations for a link farm

Analysis



Suppose rank contributed by accessible pages = x

Let page rank of target page = y

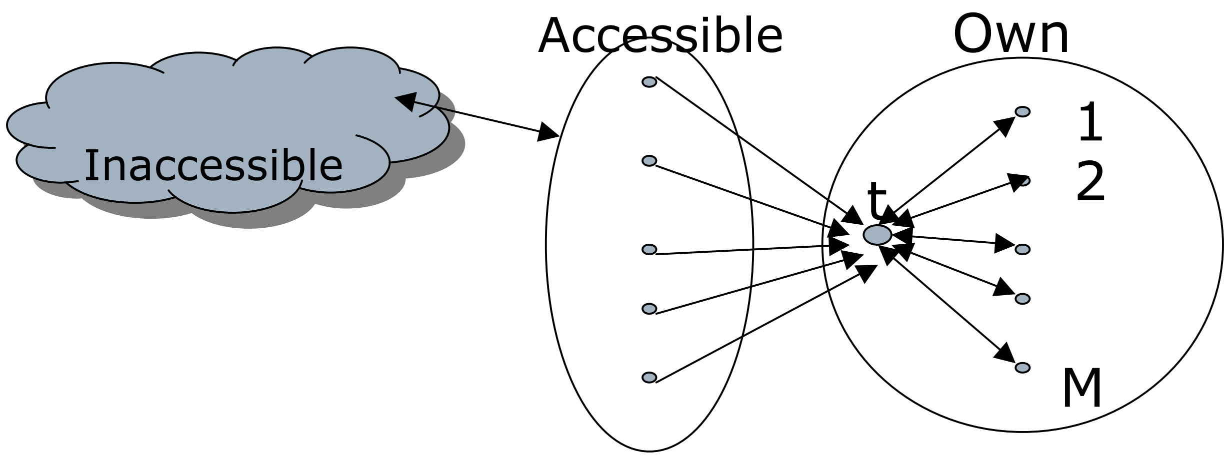
Rank of each "farm" page = $\beta y/M + (1-\beta)/N$

$y = x + \beta M[\beta y/M + (1-\beta)/N] + (1-\beta)/N$

$\quad = x + \beta^2 y + \beta(1-\beta)M/N + \boxed{(1-\beta)/N}$ Very small; ignore

$y = x/(1-\beta^2) + cM/N$ where $c = \beta/(1+\beta)$

Analysis



- $y = x/(1-\beta^2) + cM/N$ where $c = \beta/(1+\beta)$
- For $\beta = 0.85$, $1/(1-\beta^2) = 3.6$
  - Multiplier effect for "acquired" page rank
  - By making M large, we can make y as large as we want

Detecting Spam

- Term spamming
  - Analyze text using statistical methods e.g., Naïve Bayes classifiers
  - Similar to email spam filtering
  - Also useful: detecting approximate duplicate pages
- Link spamming
  - Open research area
  - One approach: TrustRank

TrustRank idea

- Basic principle: approximate isolation
  - It is rare for a “good” page to point to a “bad” (spam) page
- Sample a set of “seed pages” from the web
- Have an oracle (human) identify the good pages and the spam pages in the seed set
  - Expensive task, so must make seed set as small as possible

Trust Propagation

- Call the subset of seed pages that are identified as “good” the “trusted pages”
- Set trust of each trusted page to 1
- Propagate trust through links
  - Each page gets a trust value between 0 and 1
  - Use a threshold value and mark all pages below the trust threshold as spam

Rules for trust propagation

- Trust attenuation
  - The degree of trust conferred by a trusted page decreases with distance
- Trust splitting
  - The larger the number of outlinks from a page, the less scrutiny the page author gives each outlink
  - Trust is "split" across outlinks

Simple model

- Suppose trust of page p is t(p)
  - Set of outlinks O(p)
- For each q2O(p), p confers the trust
  - $\beta t(p)/|O(p)|$ for $0<\beta<1$
- Trust is additive
  - Trust of p is the sum of the trust conferred on p by all its inlinked pages
- Note similarity to Topic-Specific Page Rank
  - Within a scaling factor, trust rank = biased page rank with trusted pages as teleport set

Picking the seed set

- Two conflicting considerations
  - Human has to inspect each seed page, so seed set must be as small as possible
  - Must ensure every “good page” gets adequate trust rank, so need make all good pages reachable from seed set by short paths

Approaches to picking seed set

- Suppose we want to pick a seed set of k pages
- PageRank
  - Pick the top k pages by page rank
  - Assume high page rank pages are close to other highly ranked pages
  - We care more about high page rank “good” pages

Inverse page rank

- Pick the pages with the maximum number of outlinks
- Can make it recursive
  - Pick pages that link to pages with many outlinks
- Formalize as “inverse page rank”
  - Construct graph G′ by reversing each edge in web graph G
  - Page Rank in G′ is inverse page rank in G
- Pick top k pages by inverse page rank

Spam Mass

- In the TrustRank model, we start with good pages and propagate trust
- Complementary view: what fraction of a page’s page rank comes from “spam” pages?
- In practice, we don’t know all the spam pages, so we need to estimate

Spam mass estimation

$r(p)$ = page rank of page p

$r^+(p)$ = page rank of p with teleport into "good" pages only

$r^-(p) = r(p) - r^+(p)$

Spam mass of p = $r^-(p)/r(p)$

Good pages

- For spam mass, we need a large set of "good" pages
  - Need not be as careful about quality of individual pages as with TrustRank
- One reasonable approach
  - .edu sites
  - .gov sites
  - .mil sites

Another approach

- Backflow from known spam pages
  - Course project from last year's edition of this course
- Still an open area of research…