Information Retrieval

CS 547/DS 547

Worcester Polytechnic Institute

Department of Computer Science

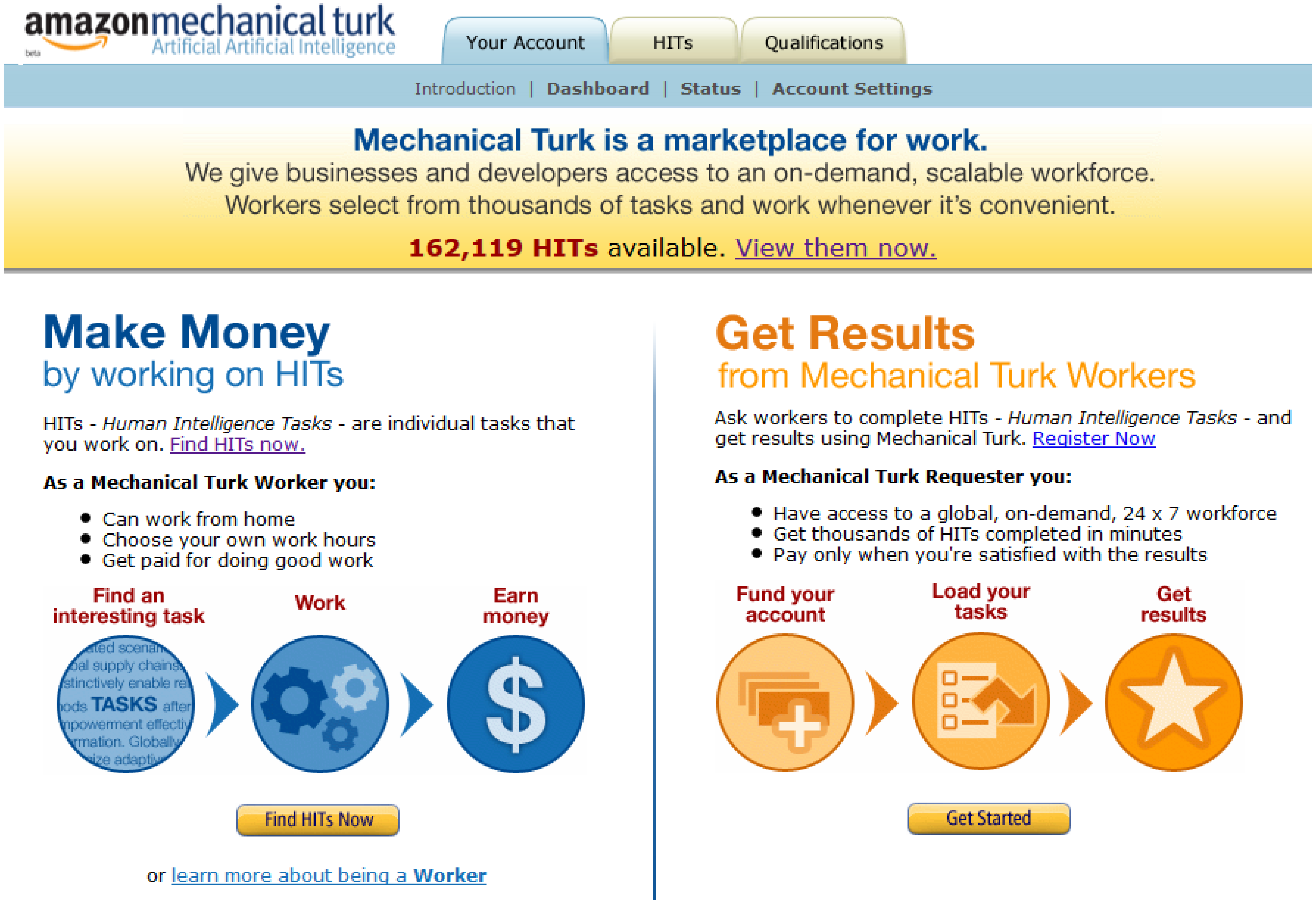
Instructor: Prof. Kyumin Lee

PageRank

Link-based ranking

- Query processing with link-based ranking:
  - First retrieve all pages meeting the query (say **venture capital**)
  - Order these by their link popularity (= citation frequency, first generation)
  - . . . or by Pagerank (second generation)

- Simple link popularity (= number of inlinks of a page) is easy to spam.
- Why?



amazonmechanical turk
Artificial Artificial Intelligence
beta
Your Account
HITs
Qualifications
Introduction | Dashboard | Status | Account Settings

Mechanical Turk is a marketplace for work.

We give businesses and developers access to an on-demand, scalable workforce.
Workers select from thousands of tasks and work whenever it's convenient.

162,119 HITs available. View them now.

Make Money
by working on HITs

HITs - *Human Intelligence Tasks* - are individual tasks that you work on. Find HITs now.

As a Mechanical Turk Worker you:

- Can work from home
- Choose your own work hours
- Get paid for doing good work

or learn more about being a **Worker**

Get Results
from Mechanical Turk Workers

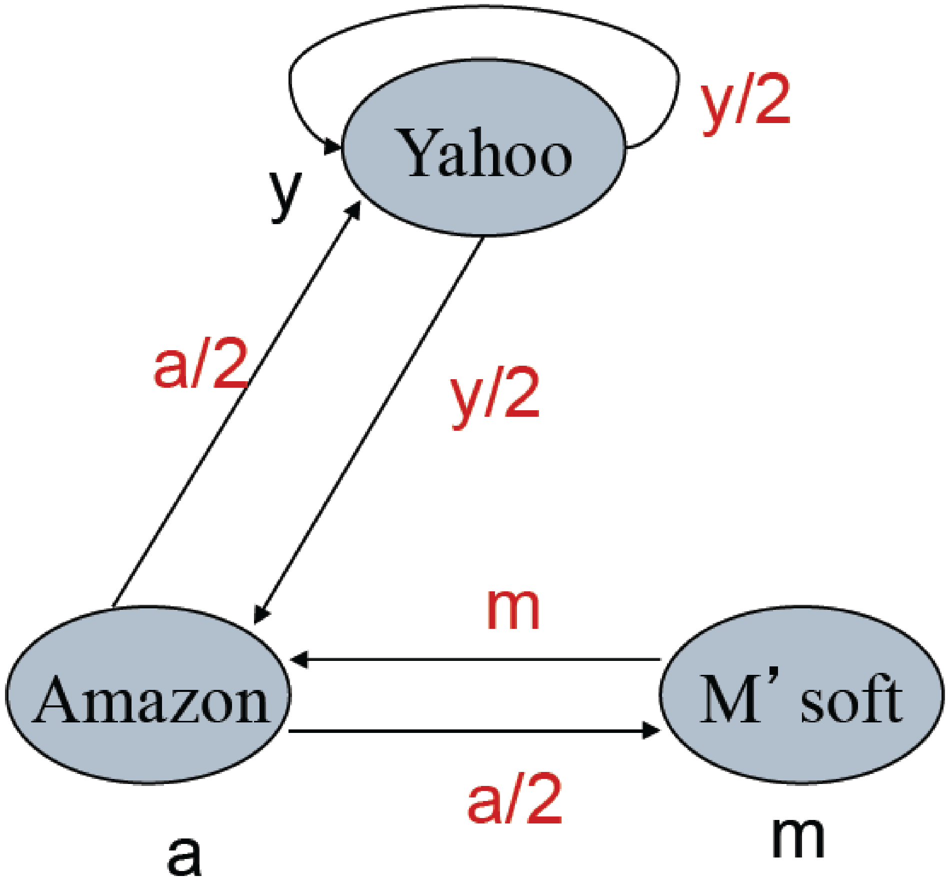
Ask workers to complete HITs - *Human Intelligence Tasks* - and get results using Mechanical Turk. Register Now

As a Mechanical Turk Requester you:

- Have access to a global, on-demand, 24 x 7 workforce
- Get thousands of HITs completed in minutes
- Pay only when you're satisfied with the results

PageRank:
Recursive formulation

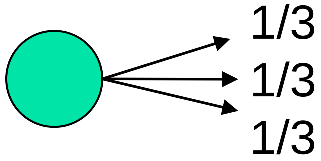
- Each link’s vote is proportional to the **importance of its source page**
- If page P with importance x has n outlines, each link gets x/n votes
- Page P’s own importance is the sum of the vote on its inlinks



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

PageRank basics

- Imagine a web surfer doing a random walk on the web
  - – Start at a random page
  - – At each step, go out of the current page along one of the links on that page, equiprobably

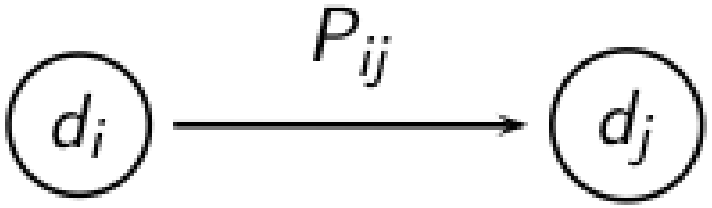


- "In the steady state" each page has a long-term visit rate - use this as the page's score.
- **PageRank = steady state probability**

  = long-term visit rate

Markov chains

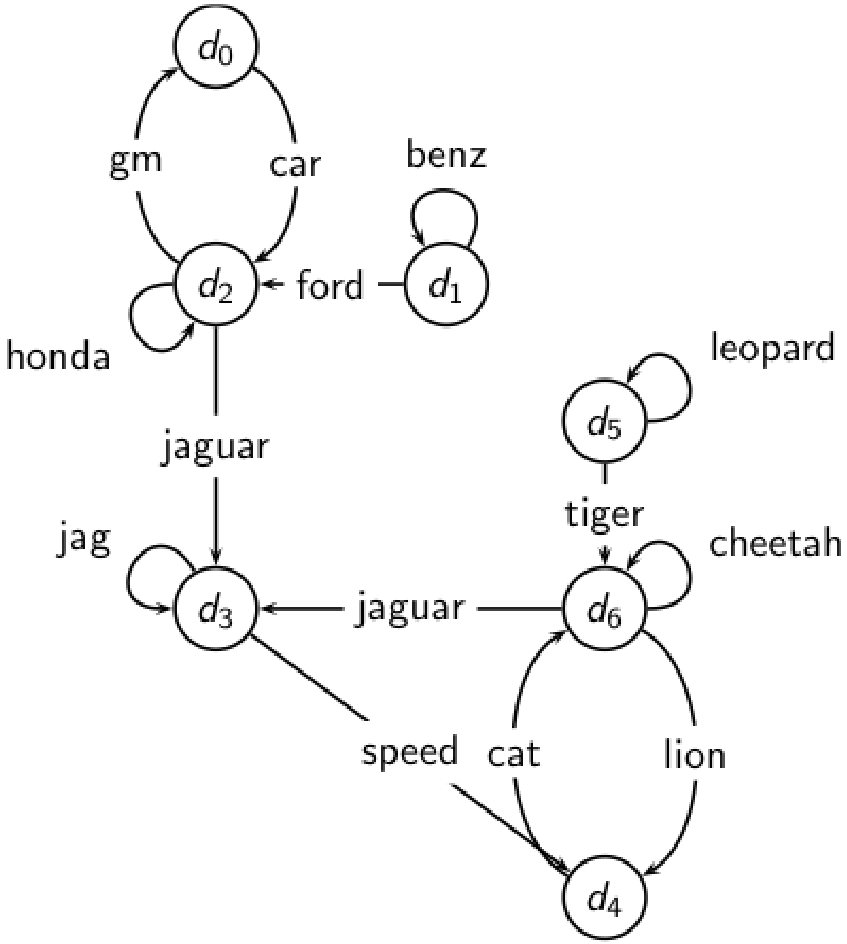
- A Markov chain consists of n states, plus an n×n transition probability matrix **P**.
- **state = page**
- At each step, we are on exactly one of the states.
- For $1 \leq i, j \leq$ n, the matrix entry P_{ij} tells us the probability of *j* being the next state (page), given we are currently on page (state) *i*.



Markov chains

- Clearly, for all i, $\sum_{j=1}^{N} P_{ij} = 1$
- Markov chains are abstractions of random walks.

Example web graph



And the corresponding link matrix

$$\begin{array}{cccccccc}
 & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\
d_0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
d_1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
d_2 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
d_3 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
d_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
d_5 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
d_6 & 0 & 0 & 0 & 1 & 1 & 0 & 1
\end{array}$$

Transition probability matrix P

$$
\begin{array}{cccccccc}
 & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\
d_0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
d_1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
d_2 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
d_3 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
d_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
d_5 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
d_6 & 0 & 0 & 0 & 1 & 1 & 0 & 1
\end{array}
$$

Transition probability matrix

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0.00	0.00	1.00	0.00	0.00	0.00	0.00
d_1	0.00	0.50	0.50	0.00	0.00	0.00	0.00
d_2	0.33	0.00	0.33	0.33	0.00	0.00	0.00
d_3	0.00	0.00	0.00	0.50	0.50	0.00	0.00
d_4	0.00	0.00	0.00	0.00	0.00	0.00	1.00
d_5	0.00	0.00	0.00	0.00	0.00	0.50	0.50
d_6	0.00	0.00	0.00	0.33	0.33	0.00	0.33

Long-term visit rate

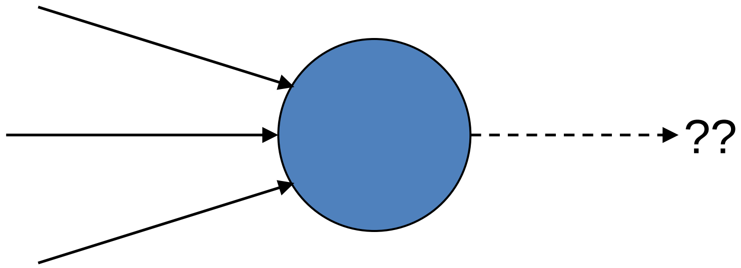
- Recall: PageRank = long-term visit rate

- Long-term visit rate of page ***d*** is the probability that a web surfer is at page ***d*** at a given point in time.

- Next: what properties must hold of the web graph for the long-term visit rate to be well defined?

Not quite enough

- The web is full of dead-ends.
  - Random walk can get stuck in dead-ends.
  - Makes no sense to talk about long-term visit rates.



Teleporting

- At a dead end, jump to a random web page.
- At any non-dead end, with probability 10%, jump to a random web page.
  - With remaining probability (90%), go out on a random link.
  - 10% - a parameter.

Teleporting Matrix

- Recall: At a dead end, jump to a random web page

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	1/7	1/7	1/7	1/7	1/7	1/7	1/7
d_1	1/7	1/7	1/7	1/7	1/7	1/7	1/7
d_2	1/7	1/7	1/7	1/7	1/7	1/7	1/7
d_3	1/7	1/7	1/7	1/7	1/7	1/7	1/7
d_4	1/7	1/7	1/7	1/7	1/7	1/7	1/7
d_5	1/7	1/7	1/7	1/7	1/7	1/7	1/7
d_6	1/7	1/7	1/7	1/7	1/7	1/7	1/7

Result of teleporting

- With teleporting, we cannot get stuck in a dead end

- There is a long-term rate at which any page is visited (not obvious, will show this).

- How do we compute this visit rate?

Formalization of “visit”: Probability vectors

- A probability (row) vector **x** *= (x$_1$, … x$_n$)* tells us where the walk is at any point.
- E.g., $(\underset{1}{0}00\ldots\underset{i}{1}\ldots00\underset{n}{0})$ means we’re in state *i*.

- More generally, the vector **x** *= (x$_1$, … x$_n$)* means the walk is in state *i* with probability x_i.

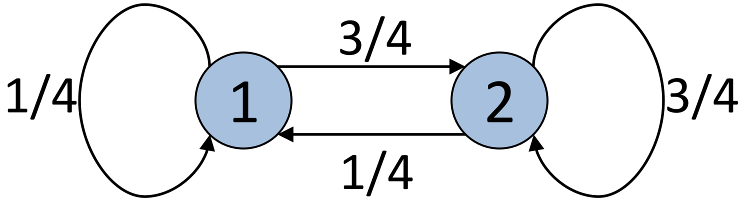
$$\sum_{i=1}^{n} x_i = 1.$$

Change in probability vector

- If the probability vector is **x** $= (x_1, \ldots x_n)$ at this step, what is it at the next step?
- Recall that row *i* of the transition prob. Matrix **P** tells us where we go next from state *i*.
- So from **x**, our next state is distributed as **xP**.

Steady state example

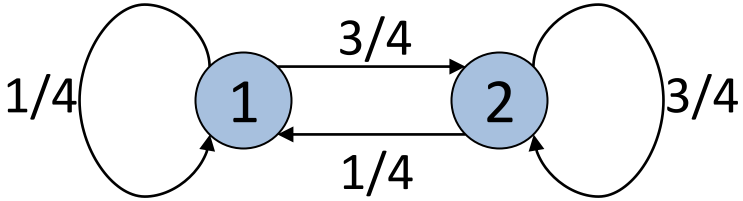
- The steady state looks like a vector of probabilities **a** = $(a_1, \ldots a_n)$:
- a_i is the probability that we are in state *i*.



What is the steady state in this example?

Steady state example

- The steady state looks like a vector of probabilities **a** = $(a_1, \ldots a_n)$:
- a_i is the probability that we are in state *i*.

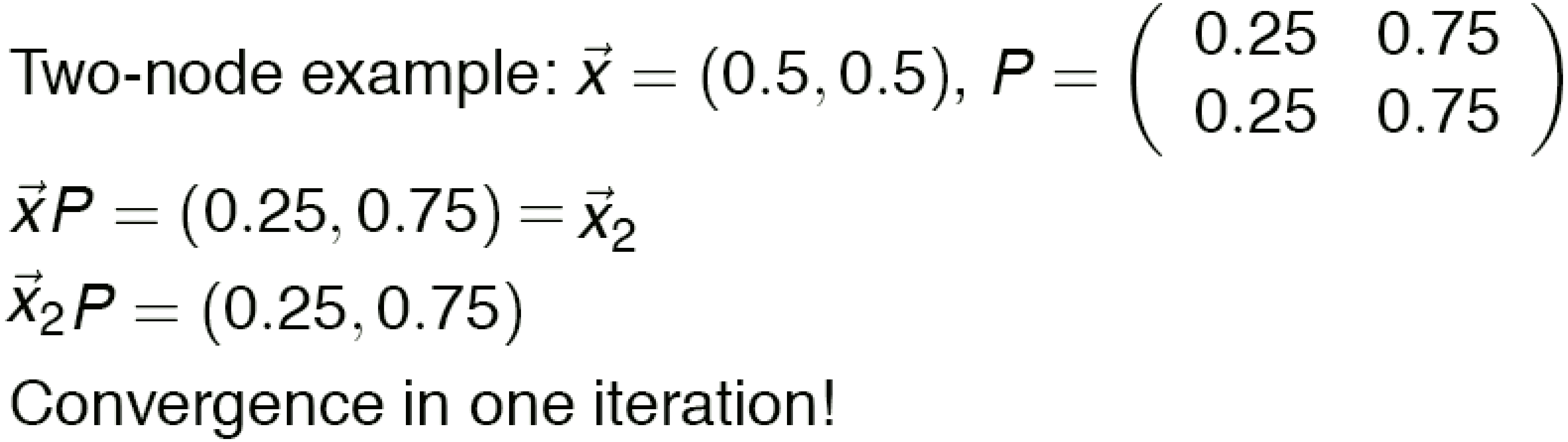


For this example, $a_1=1/4$ and $a_2=3/4$.

How to compute the steady-state?

- Recall, regardless of where we start, we eventually reach the steady state **a**.
- Start with any distribution (say **x**=(*10…0*)).
- After one step, we're at **xP**;
- after two steps at $\mathbf{xP}^2$, then $\mathbf{xP}^3$ and so on.
- "Eventually" means for "large" *k*, $\mathbf{xP}^k$ = **a**.
- Algorithm: multiply **x** by increasing powers of **P** until the product looks stable.
- This is called the power method

Power method: example



Two-node example: $\vec{x} = (0.5, 0.5)$, $P = \begin{pmatrix} 0.25 & 0.75 \\ 0.25 & 0.75 \end{pmatrix}$

$\vec{x}P = (0.25, 0.75) = \vec{x}_2$

$\vec{x}_2 P = (0.25, 0.75)$

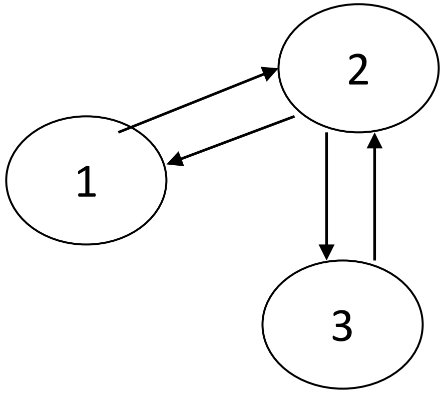
Convergence in one iteration!

Exercise on PageRank

Transition probability matrix of a surfer's walk with teleportation:

P = (1- α) * transition matrix + α * teleporting matrix

- Consider a Web graph with three nodes 1, 2, and 3. The links are as follows: 1->2, 3->2, 2->1, 2->3. Write down the transition probability matrices P and pagerank scores for the surfer's walk with teleporting, with the value of teleport probability α=0.5.



P =

0	1	0
1	0	1
0	1	0

Each 1 divied by the number of ones in this row

(1-α)*

0	1	0
½	0	½
0	1	0

+

α*

1/3	1/3	1/3
1/3	1/3	1/3
1/3	1/3	1/3

=

1/6	2/3	1/6
5/12	1/6	5/12
1/6	2/3	1/6

Exercise on PageRank (Cont'd)

Remember

$\vec{x}_1 = \vec{x}_0 P$

$\vec{x}_2 = \vec{x}_1 P$

$\vec{x}_3 = \vec{x}_2 P$

…

…

…

Until converged

$\vec{x}_0 =$

1	0	0

P=

1/6	2/3	1/6
5/12	1/6	5/12
1/6	2/3	1/6

$\vec{x}_1 =$

1/6	2/3	1/6

$\vec{x}_2 =$

1/3	1/3	1/3

$\vec{x}_3 =$

1/4	1/2	1/4

…

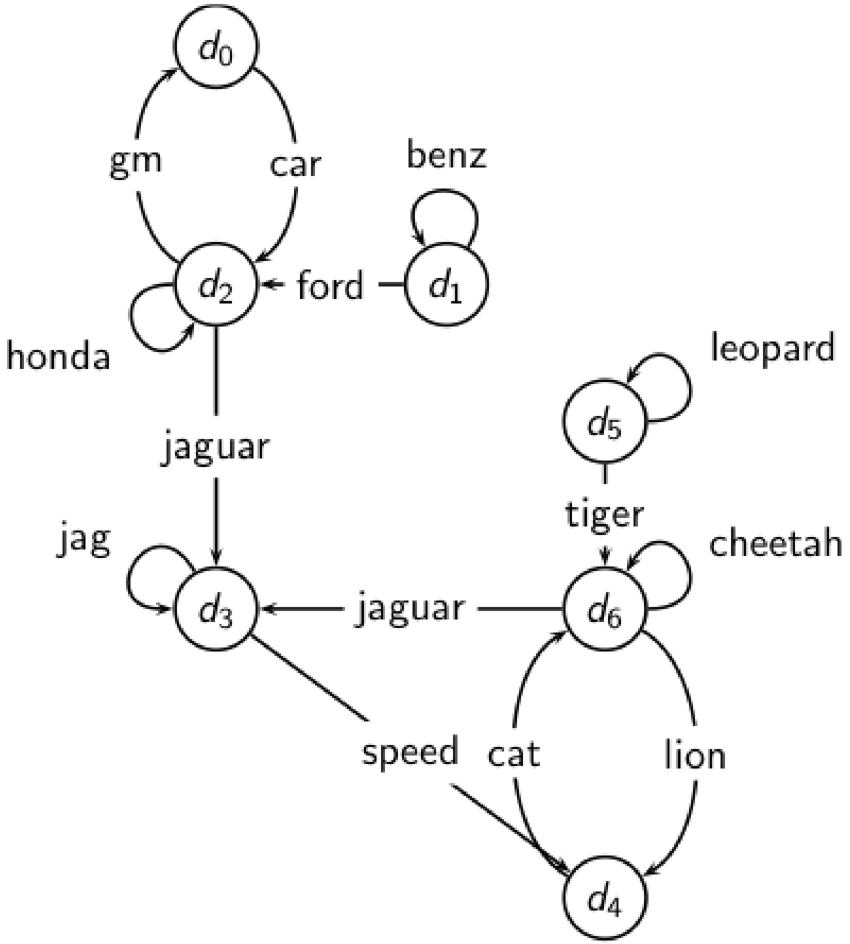
…

$\vec{x}_k =$

5/18	4/9	5/18

← converged

Example web graph



And the corresponding link matrix

$$\begin{array}{c|ccccccc}
 & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\
\hline
d_0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
d_1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
d_2 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\
d_3 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
d_4 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
d_5 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
d_6 & 0 & 0 & 0 & 1 & 1 & 0 & 1
\end{array}$$

Transition matrix with teleporting

$$\begin{array}{cccccccc}
 & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\
d_0 & 0.00 & 0.00 & 1.00 & 0.00 & 0.00 & 0.00 & 0.00 \\
d_1 & 0.00 & 0.50 & 0.50 & 0.00 & 0.00 & 0.00 & 0.00 \\
d_2 & 0.33 & 0.00 & 0.33 & 0.33 & 0.00 & 0.00 & 0.00 \\
d_3 & 0.00 & 0.00 & 0.00 & 0.50 & 0.50 & 0.00 & 0.00 \\
d_4 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 1.00 \\
d_5 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.50 & 0.50 \\
d_6 & 0.00 & 0.00 & 0.00 & 0.33 & 0.33 & 0.00 & 0.33
\end{array}$$

$\alpha = 0.14$

$$P = \begin{array}{cccccccc}
 & d_0 & d_1 & d_2 & d_3 & d_4 & d_5 & d_6 \\
d_0 & 0.02 & 0.02 & 0.88 & 0.02 & 0.02 & 0.02 & 0.02 \\
d_1 & 0.02 & 0.45 & 0.45 & 0.02 & 0.02 & 0.02 & 0.02 \\
d_2 & 0.31 & 0.02 & 0.31 & 0.31 & 0.02 & 0.02 & 0.02 \\
d_3 & 0.02 & 0.02 & 0.02 & 0.45 & 0.45 & 0.02 & 0.02 \\
d_4 & 0.02 & 0.02 & 0.02 & 0.02 & 0.02 & 0.02 & 0.88 \\
d_5 & 0.02 & 0.02 & 0.02 & 0.02 & 0.02 & 0.45 & 0.45 \\
d_6 & 0.02 & 0.02 & 0.02 & 0.31 & 0.31 & 0.02 & 0.31
\end{array}$$

Power method convergence

	x	xP^1	xP^2	xP^3	xP^4	xP^5	xP^6	xP^7	xP^8	xP^9	xP^{10}	xP^{11}	xP^{12}	xP^{13}
d_0	0.14	0.06	0.09	0.07	0.07	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05	0.05
d_1	0.14	0.08	0.06	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
d_2	0.14	0.25	0.18	0.17	0.15	0.14	0.13	0.12	0.12	0.12	0.12	0.11	0.11	0.11
d_3	0.14	0.16	0.23	0.24	0.24	0.24	0.24	0.25	0.25	0.25	0.25	0.25	0.25	0.25
d_4	0.14	0.12	0.16	0.19	0.19	0.20	0.21	0.21	0.21	0.21	0.21	0.21	0.21	0.21
d_5	0.14	0.08	0.06	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04	0.04
d_6	0.14	0.25	0.23	0.25	0.27	0.28	0.29	0.29	0.30	0.30	0.30	0.30	0.31	0.31

Pagerank summary

- Preprocessing:
  - Given graph of links, build matrix **P**.
  - From it compute **a**.
  - The entry a_i is a number between 0 and 1: the pagerank of page *i*.
- Query processing:
  - Retrieve pages meeting query.
  - Rank them by their pagerank.
  - Order is **query-*independent***.

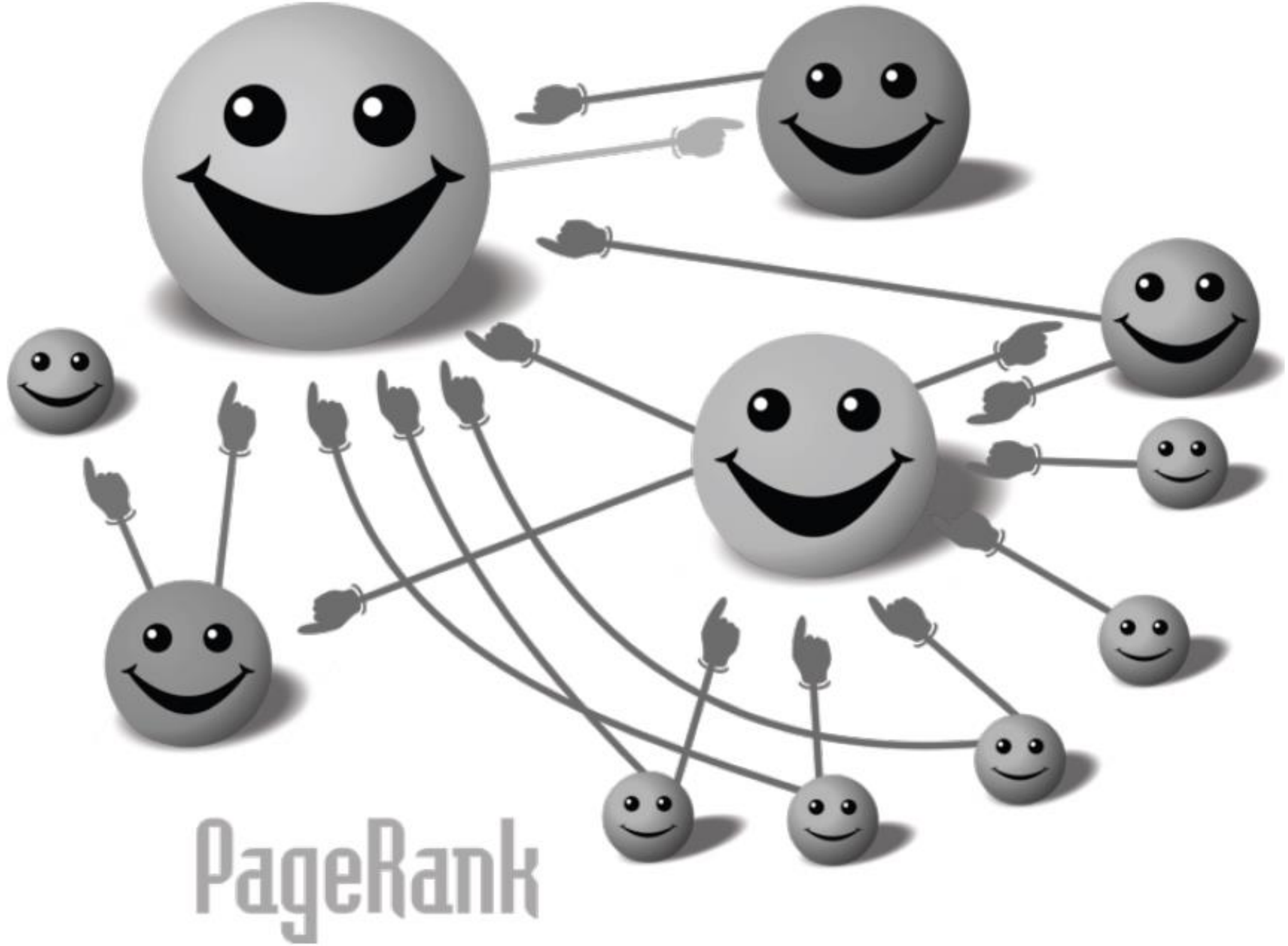
PageRank issues

- Real surfers are not random surfers – Markov model is not a good model of surfing.
  - Issues: back button, short vs. long paths, bookmarks, directories – and search!
- Simple PageRank ranking (as described on previous slide) produces bad results for many pages.
  - Consider the query ***video service***
  - The Yahoo home page (i) has a very high PageRank and (ii) contains both words.
  - If we rank all Boolean hits according to PageRank, then the Yahoo home page would be top-ranked.
  - Clearly not desirable
- In practice: rank according to weighted combination of many factors, including raw text match, anchor text match, PageRank and many other factors

How important is PageRank?

- Frequent claim: PageRank is the most important component of web ranking.
- The reality:
  - There are several components that are at least as important: e.g., anchor text, indexing , zone weighting, phrases ...
- Rumor has it that PageRank in his original form (as presented here) now has a negligible impact on ranking!
- However, variants of a page’s PageRank are still an essential part of ranking.
- Addressing link spam is difficult and crucial.

What is PageRank?



HITS: Hubs & Authorities

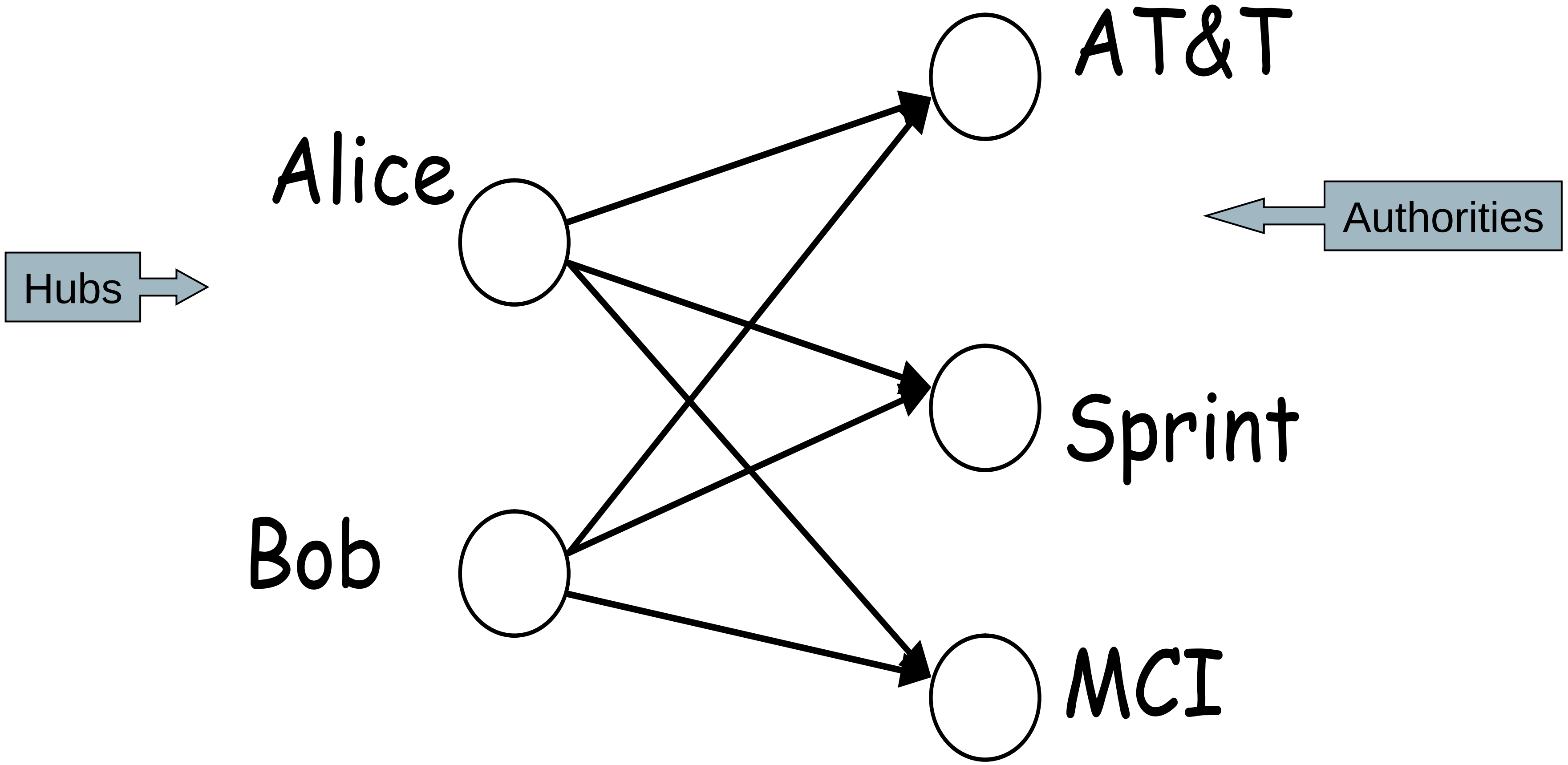
HITS – Hyperlink-Induced Topic Search

- Premise: there are two different types of relevance on the web.
- Relevance type 1: Hubs. A hub page is a good list of links to pages answering the information need.
  - E.g, for query [chicago bulls]: Bob’s list of recommended resources on the Chicago Bulls sports team
- Relevance type 2: Authorities. An authority page is a direct answer to the information need.
  - The home page of the Chicago Bulls sports team
  - By definition: Links to authority pages occur repeatedly on hub pages.
- Most approaches to search (including PageRank ranking) don’t make the distinction between these two very different types of relevance.

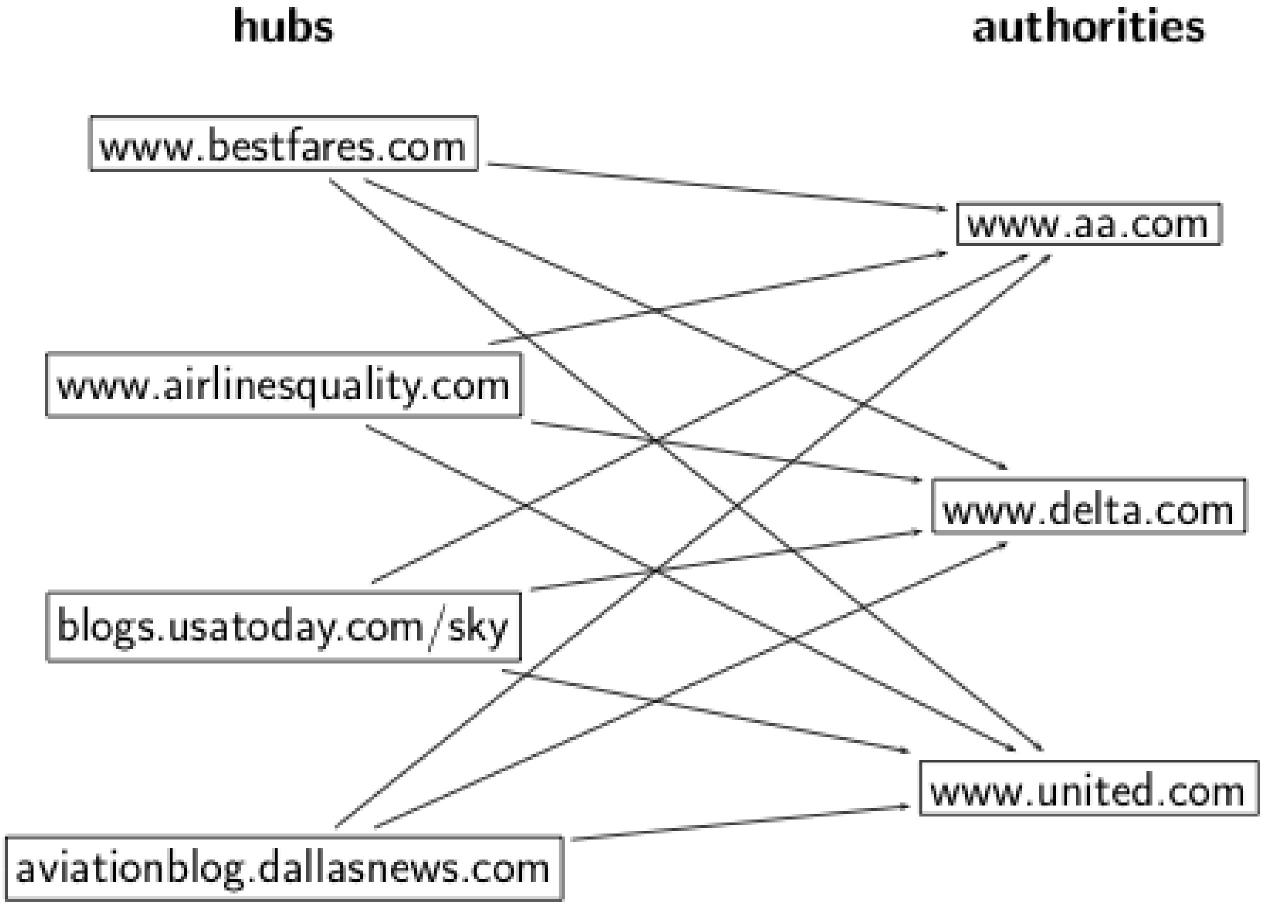
Hubs and authorities : Definition

- Thus, a good hub page for a topic *points* to many authority pages for that topic.
- A good authority page for a topic is *pointed* to by many hub pages for that topic.
- Circular definition – we will turn this into an iterative computation.

The hope



Long distance telephone companies

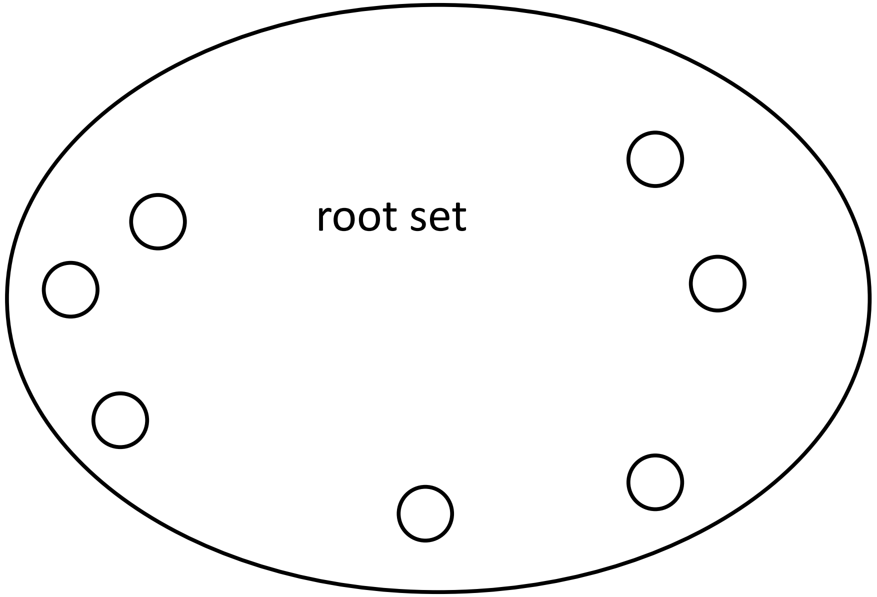

hubs
authorities
www.bestfares.com
www.airlinesquality.com
blogs.usatoday.com/sky
aviationblog.dallasnews.com
www.aa.com
www.delta.com
www.united.com

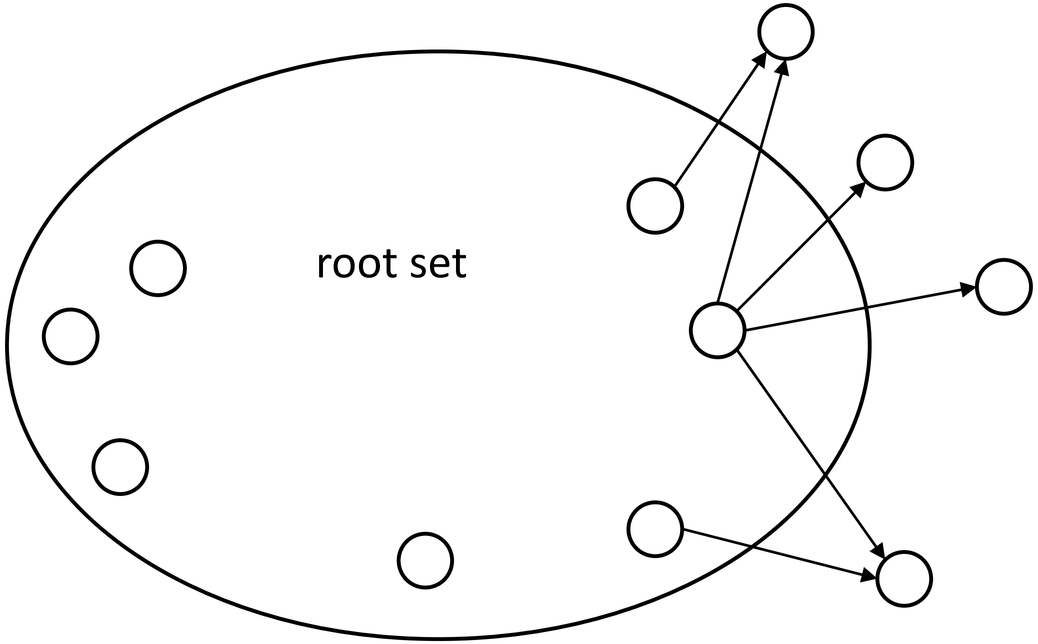
High-level scheme

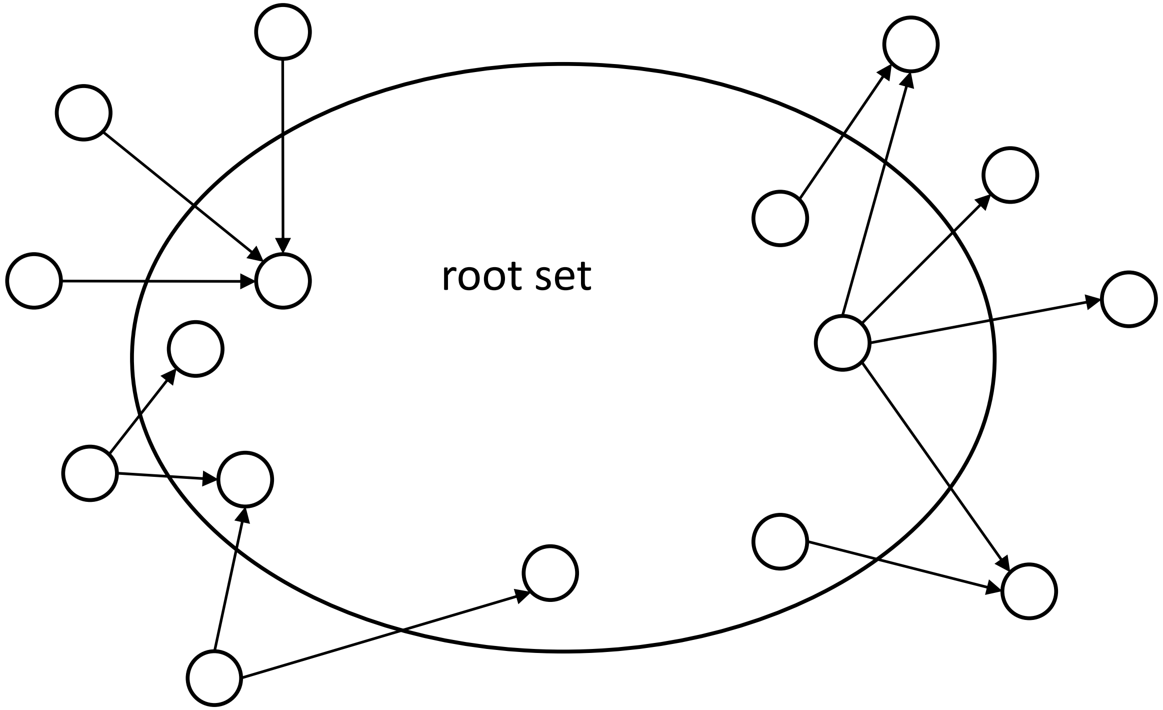
- Extract from the web a base set of pages that *could* be good hubs or authorities.
- From these, identify a small set of top hub and authority pages;
  - →iterative algorithm.

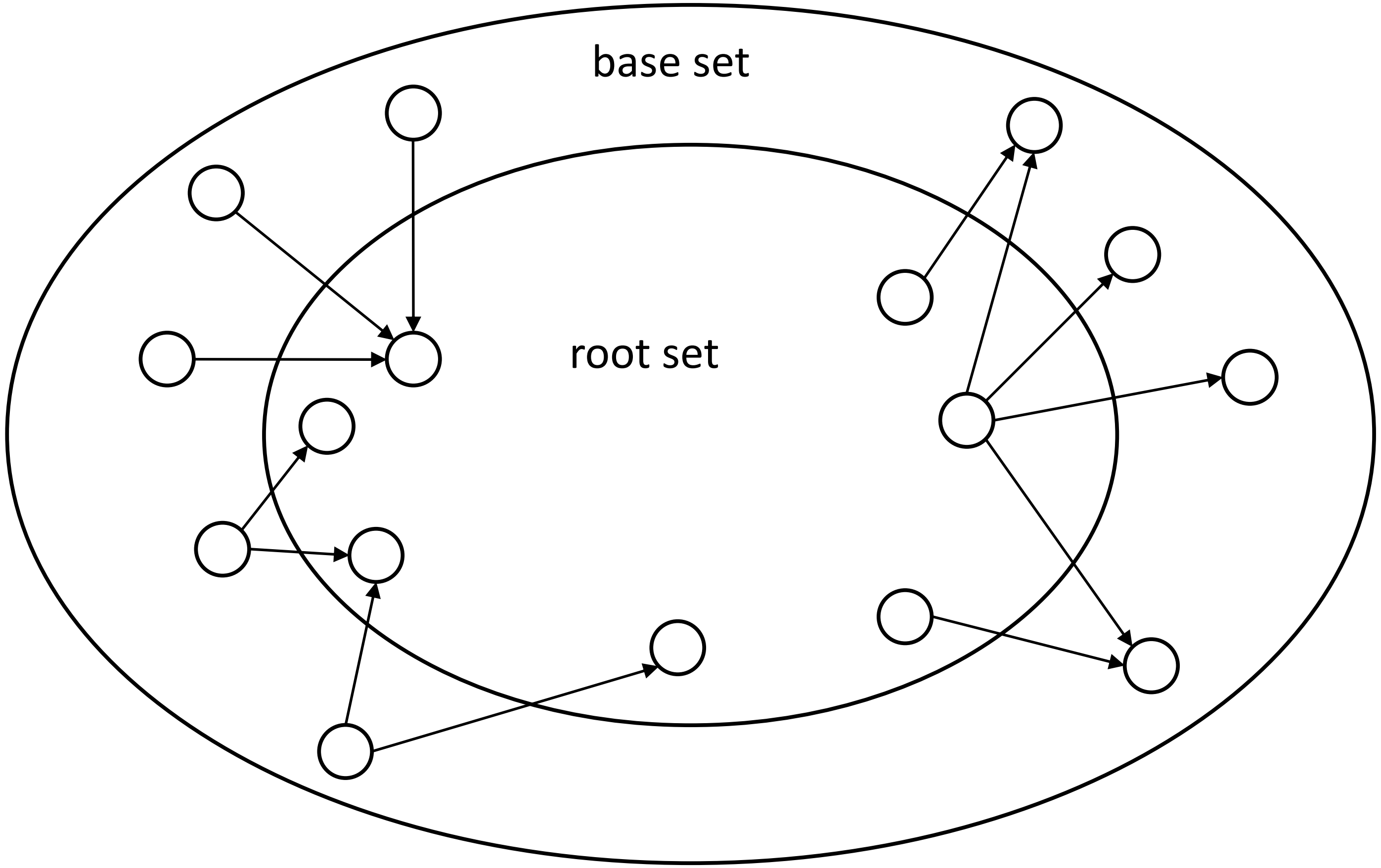
Root set and base set

- Do a regular web search first
- Call the search result the root set
- Find all pages that are linked to or link to pages in the root set
- Call first larger set the base set
- Finally, compute hubs and authorities for the base set (which we'll view as a small web graph)


root set


root set


root set


base set
root set

Root set and base set

- Root set typically has 200-1000 nodes.
- Base set may have up to 5000 nodes.
- Computation of base set, as shown on previous slide:
  - Follow outlinks by parsing the pages in the root set
  - Find *x*'s inlinks by searching for all pages containing a link to *x*
  - This assumes our inverted index supports search for links (in addition to terms)

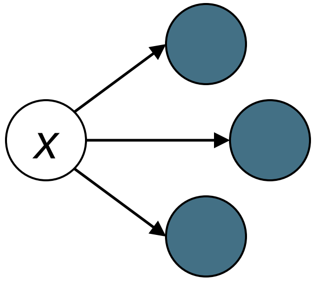
Hub and authority scores

- Compute for each page x in the base set a hub score *h*(x) and an authority score *a*(x)
- Initialization: for all x: *h*(x) ← 1, *a*(x) ← 1;
- Iteratively update all *h*(x), *a*(x)
- After convergence:
  - Output pages with highest *h()* scores as top hubs
  - highest *a()* scores as top authorities

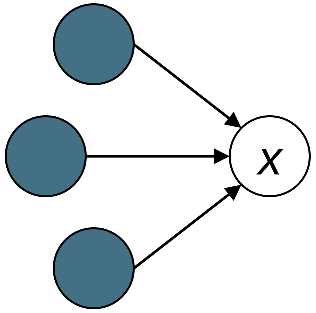
Iterative update

- Repeat the following updates, for all *x*:

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$



$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$



Scaling

- To prevent the *a()* and *h()* values from getting too big, can scale down after each iteration.
- Scaling factor doesn’t really matter:
- we only care about the ***relative*** values of the scores.

How many iterations?

- Claim: relative values of scores will converge after a few iterations:
  - in fact, suitably scaled, *h()* and *a()* scores settle into a steady state!
- We only require the relative orders of the *h()* and *a()* scores - not their absolute values.
- In practice, ~5 iterations get you close to stability.

Japan Elementary Schools

Hubs

- schools
- LINK Page-13
- “ú–{,ÌŠw•Z
- a‰„¬ŠwZƒz[ƒƒy[ƒW
- 100 Schools Home Pages (English)
- K-12 from Japan 10/...rnet and Education)
- http://www...iglobe.ne.jp/~IKESAN
- ,I,f,j¬ŠwZ,U”N,P‘g•¨Œê
- ÒŠ—’¬—§ÒŠ—“Œ¬ŠwZ
- Koulutus ja oppilaitokset
- TOYODA HOMEPAGE
- Education
- Cay's Homepage(Japanese)
- –y“ì¬ŠwZ,Ìƒz[ƒƒy[ƒW
- UNIVERSITY
- ‰J—³¬ŠwZ DRAGON97-TOP
- Â‰ª¬ŠwZ,T”N,P‘gƒz[ƒƒy[ƒW
- ¶µ°é¼ÂÁ© ¥á¥Ë¥å¡¼ ¥á¥Ë¥å¡¼

Authorities

- The American School in Japan
- The Link Page
- ‰ªè•ž—§ˆä“c¬ŠwZƒz[ƒƒy[ƒW
- Kids' Space
- ˆÀé•ž—§ˆÀé¼•”¬ŠwZ
- ‹{é‹³ˆç‘åŠw••‘®¬ŠwZ
- •“ÞìŒ§E‰¡•ls—§’†ì¼¬ŠwZ,Ìƒy
- KEIMEI GAKUEN Home Page (Japanese)
- Shiranuma Home Page
- fuzoku-es.fukui-u.ac.jp
- welcome to Miasa E&J school
- http://www...p/~m_maru/index.html
- fukui haruyama-es HomePage
- Torisu primary school
- goo
- Yakumo Elementary,Hokkaido,Japan
- FUZOKU Home Page
- Kamishibun Elementary School...

Things to note

- Pulled together good pages regardless of language of page content.
- Use *only* link analysis after base set assembled
  - iterative scoring is query-independent.
- Downside: Iterative computation after text index retrieval - significant overhead.

Hub/authority vectors

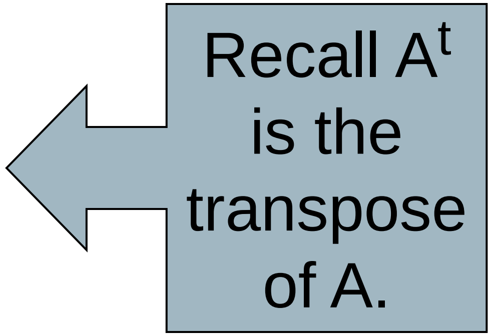
- View the hub scores *h()* and the authority scores *a()* as vectors with *n* components.
- Recall the iterative updates

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$

$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$

Rewrite in matrix form

- $\mathbf{h}=\mathbf{A}\mathbf{a}$.
- $\mathbf{a}=\mathbf{A}^{t}\mathbf{h}$.

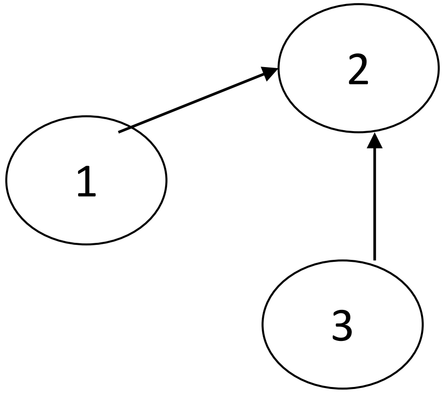


Recall A^t is the transpose of A.

- *A is a square* matrix with one row and one column for each page in the subset
  - *Aij is 1 if there is a hyperlink from page i to page j, and 0 otherwise*

Exercise on HITS

- Consider a Web graph with three nodes 1, 2, and 3. The links are as follows: 1->2, 3->2.



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

$$\vec{h}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \qquad \vec{a}_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

Remember

$$\vec{h}_1 = A\vec{a}_0 \qquad \vec{a}_1 = A^T\vec{h}_0$$

$$\vec{h}_2 = A\vec{a}_1 \qquad \vec{a}_2 = A^T\vec{h}_1$$

$$\vec{h}_3 = A\vec{a}_2 \qquad \vec{a}_3 = A^T\vec{h}_2$$

…

Until converged

$$\vec{h}_1 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \qquad \vec{a}_1 = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$$

$$\vec{h}_2 = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \qquad \vec{a}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Normalization

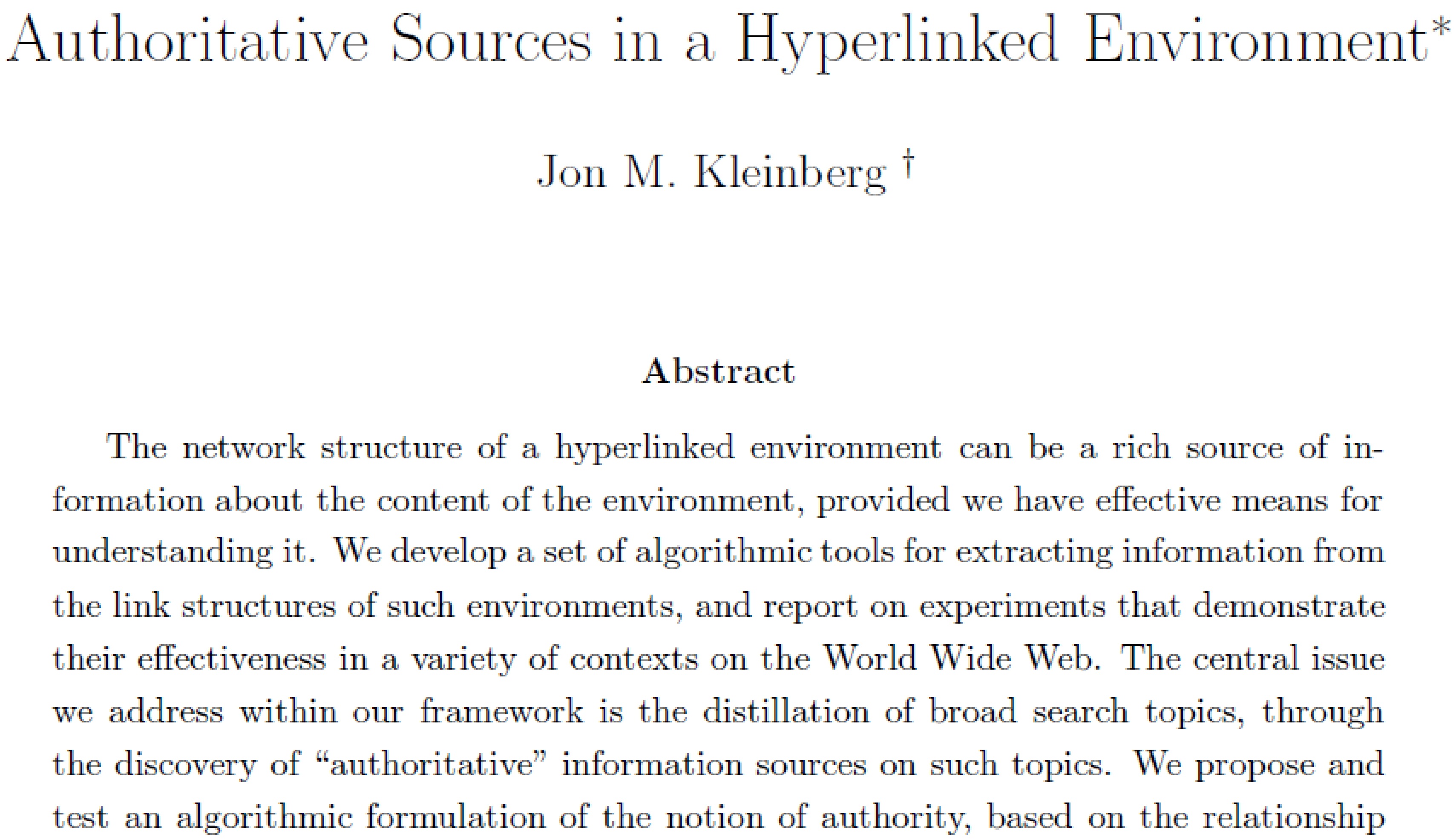
$$\vec{h}_1 = \begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix} \qquad \vec{a}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\vec{h}_2 = \begin{bmatrix} 1/2 \\ 0 \\ 1/2 \end{bmatrix} \qquad \vec{a}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

converged

PageRank vs. HITS: Discussion

- PageRank can be precomputed, HITS has to be computed at query time.
    - HITS is too expensive in most application scenarios.
- We could also apply HITS to the entire web and PageRank to a small base set.
- On the web, a good hub almost always is also a good authority.
- The actual difference between PageRank ranking and HITS ranking is therefore not as large as one might expect.

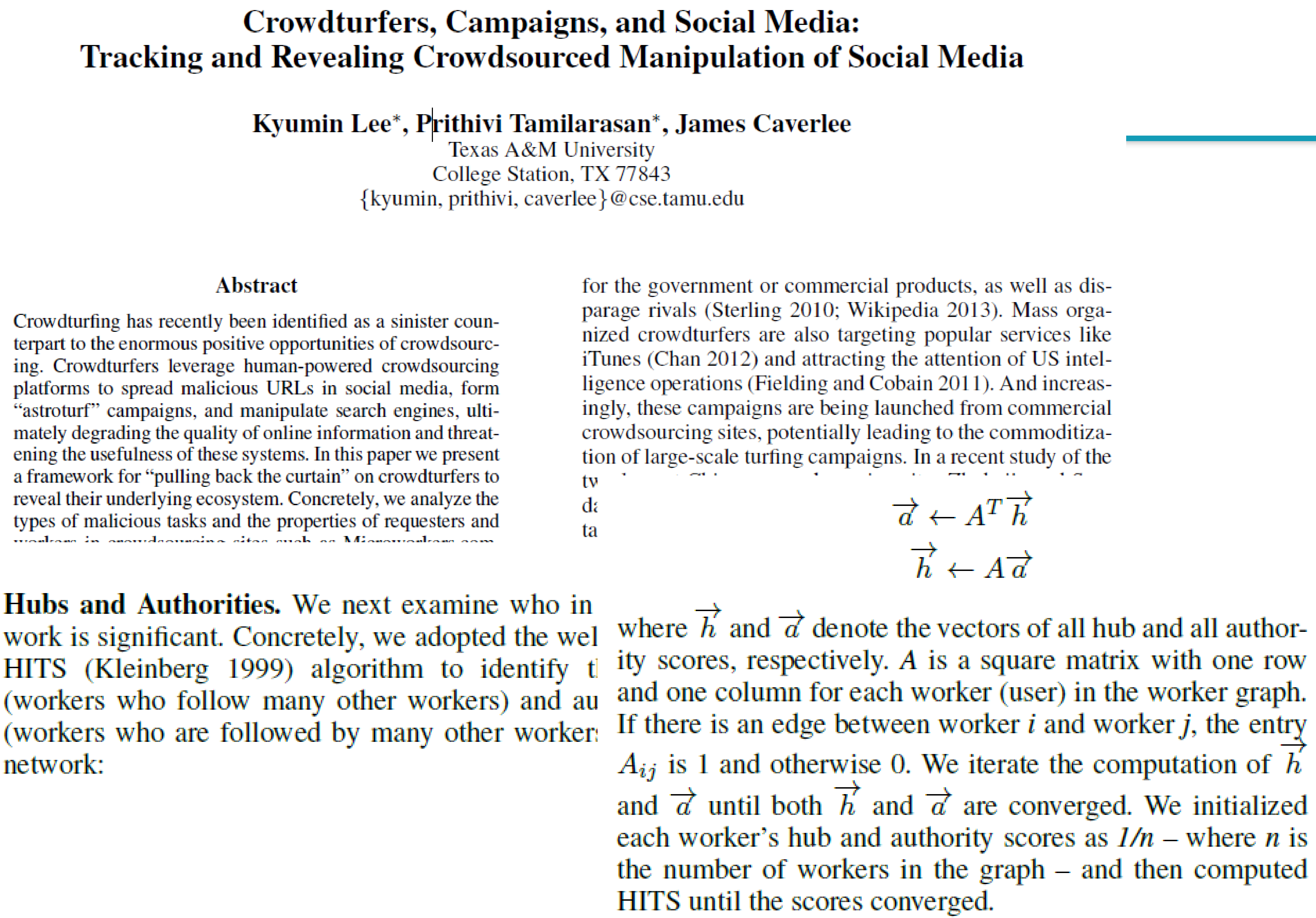


Authoritative Sources in a Hyperlinked Environment[*]

Jon M. Kleinberg [†]

Abstract

The network structure of a hyperlinked environment can be a rich source of information about the content of the environment, provided we have effective means for understanding it. We develop a set of algorithmic tools for extracting information from the link structures of such environments, and report on experiments that demonstrate their effectiveness in a variety of contexts on the World Wide Web. The central issue we address within our framework is the distillation of broad search topics, through the discovery of "authoritative" information sources on such topics. We propose and test an algorithmic formulation of the notion of authority, based on the relationship



Crowdturfers, Campaigns, and Social Media: Tracking and Revealing Crowdsourced Manipulation of Social Media

Kyumin Lee*, Prithivi Tamilarasan*, James Caverlee
Texas A&M University
College Station, TX 77843
{kyumin, prithivi, caverlee}@cse.tamu.edu

Abstract

Crowdturfing has recently been identified as a sinister counterpart to the enormous positive opportunities of crowdsourcing. Crowdturfers leverage human-powered crowdsourcing platforms to spread malicious URLs in social media, form "astroturf" campaigns, and manipulate search engines, ultimately degrading the quality of online information and threatening the usefulness of these systems. In this paper we present a framework for "pulling back the curtain" on crowdturfers to reveal their underlying ecosystem. Concretely, we analyze the types of malicious tasks and the properties of requesters and workers in crowdsourcing sites such as Microworkers.com

for the government or commercial products, as well as disparage rivals (Sterling 2010; Wikipedia 2013). Mass organized crowdturfers are also targeting popular services like iTunes (Chan 2012) and attracting the attention of US intelligence operations (Fielding and Cobain 2011). And increasingly, these campaigns are being launched from commercial crowdsourcing sites, potentially leading to the commoditization of large-scale turfing campaigns. In a recent study of the

tw

da

ta

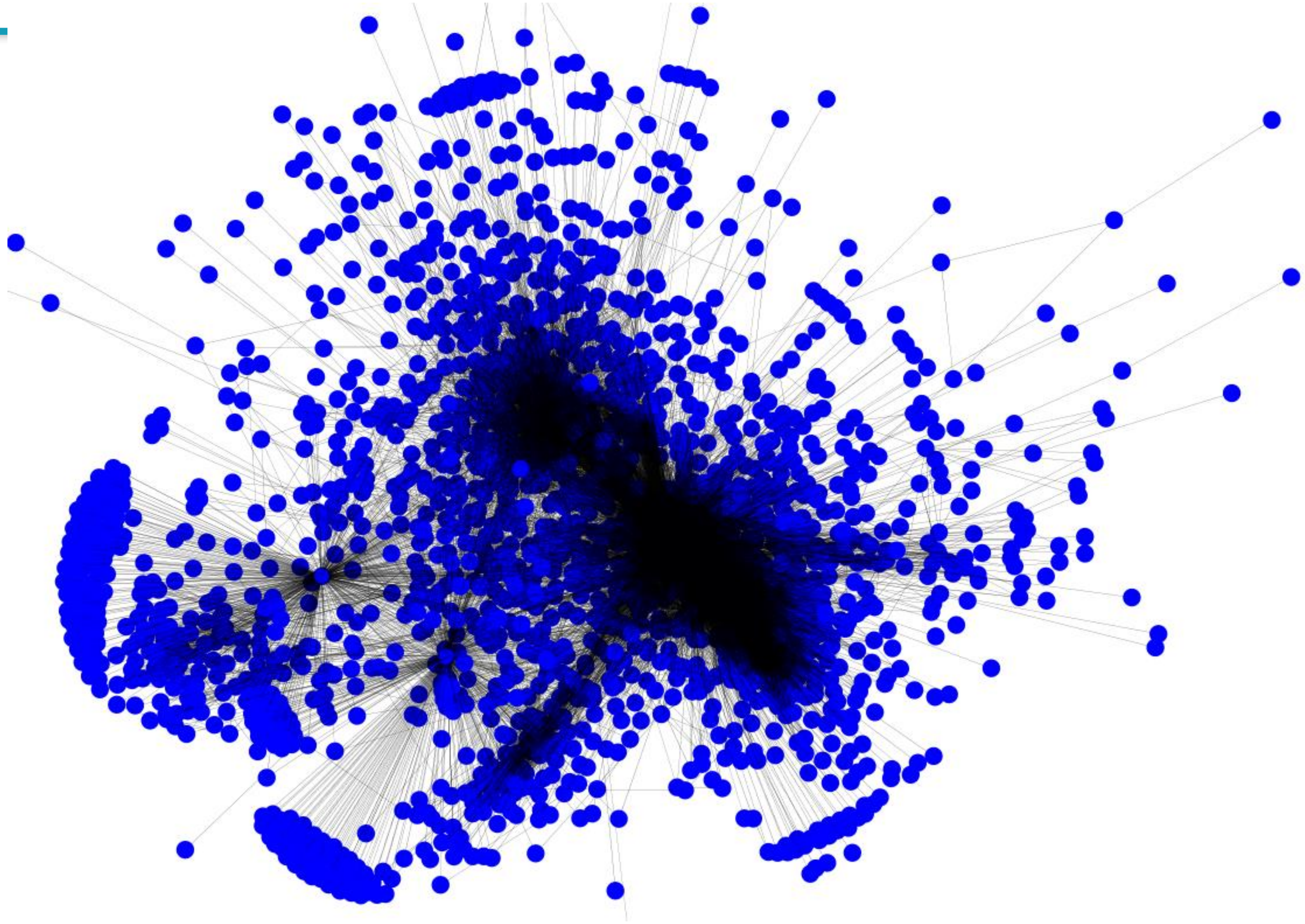
Hubs and Authorities. We next examine who in work is significant. Concretely, we adopted the wel HITS (Kleinberg 1999) algorithm to identify t (workers who follow many other workers) and au (workers who are followed by many other worker network:

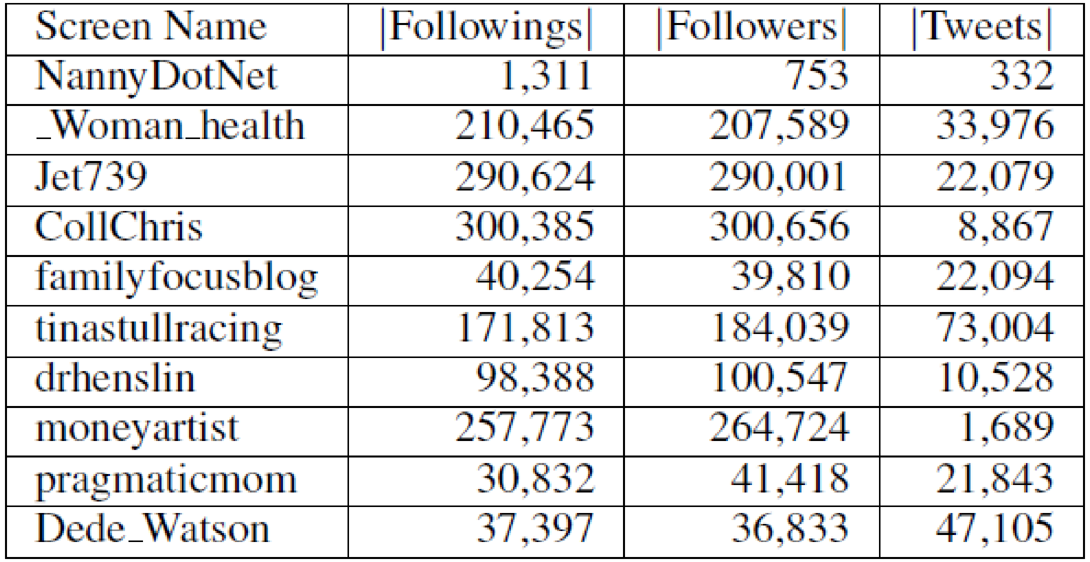
$$\vec{a} \leftarrow A^T \vec{h}$$

$$\vec{h} \leftarrow A \vec{a}$$

where $\vec{h}$ and $\vec{a}$ denote the vectors of all hub and all authority scores, respectively. A is a square matrix with one row and one column for each worker (user) in the worker graph. If there is an edge between worker i and worker j, the entry A_{ij} is 1 and otherwise 0. We iterate the computation of $\vec{h}$ and $\vec{a}$ until both $\vec{h}$ and $\vec{a}$ are converged. We initialized each worker's hub and authority scores as *1/n* – where *n* is the number of workers in the graph – and then computed HITS until the scores converged.

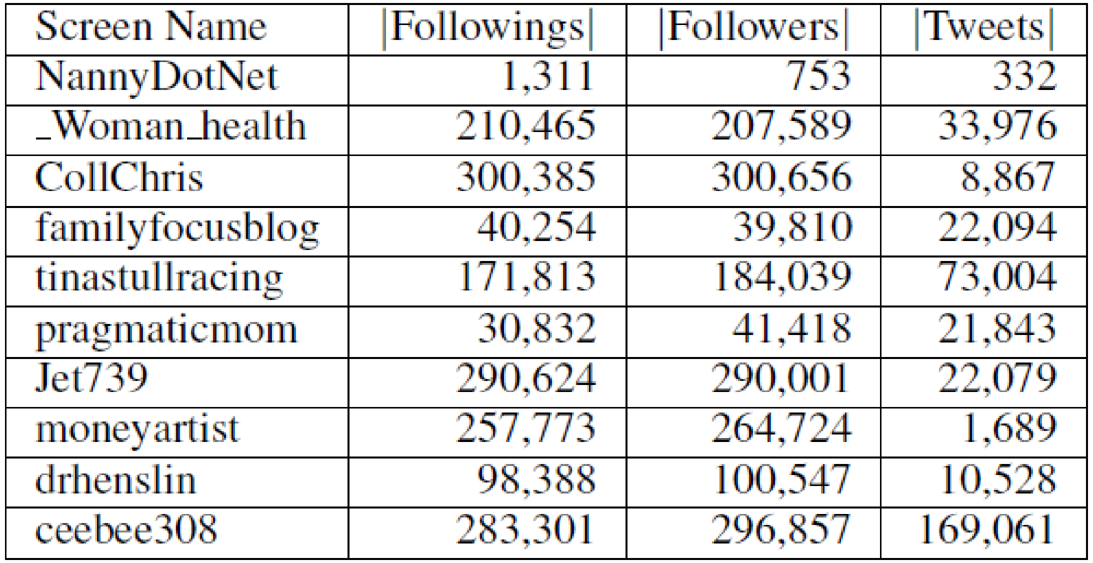
Twitter workers' following-follower relationship





Screen Name	|Followings|	|Followers|	|Tweets|
NannyDotNet	1,311	753	332
_Woman_health	210,465	207,589	33,976
Jet739	290,624	290,001	22,079
CollChris	300,385	300,656	8,867
familyfocusblog	40,254	39,810	22,094
tinastullracing	171,813	184,039	73,004
drhenslin	98,388	100,547	10,528
moneyartist	257,773	264,724	1,689
pragmaticmom	30,832	41,418	21,843
Dede_Watson	37,397	36,833	47,105

Table 6: Top-10 hubs of the workers.



Screen Name	|Followings|	|Followers|	|Tweets|
NannyDotNet	1,311	753	332
_Woman_health	210,465	207,589	33,976
CollChris	300,385	300,656	8,867
familyfocusblog	40,254	39,810	22,094
tinastullracing	171,813	184,039	73,004
pragmaticmom	30,832	41,418	21,843
Jet739	290,624	290,001	22,079
moneyartist	257,773	264,724	1,689
drhenslin	98,388	100,547	10,528
ceebee308	283,301	296,857	169,061

Table 7: Top-10 authorities of the workers.

Evaluating a Search Engine

Measuring relevance

- Three elements:
  - A benchmark document collection
  - A benchmark suite of queries
  - An assessment of either Relevant or Nonrelevant for each query and each document

Some public test Collections

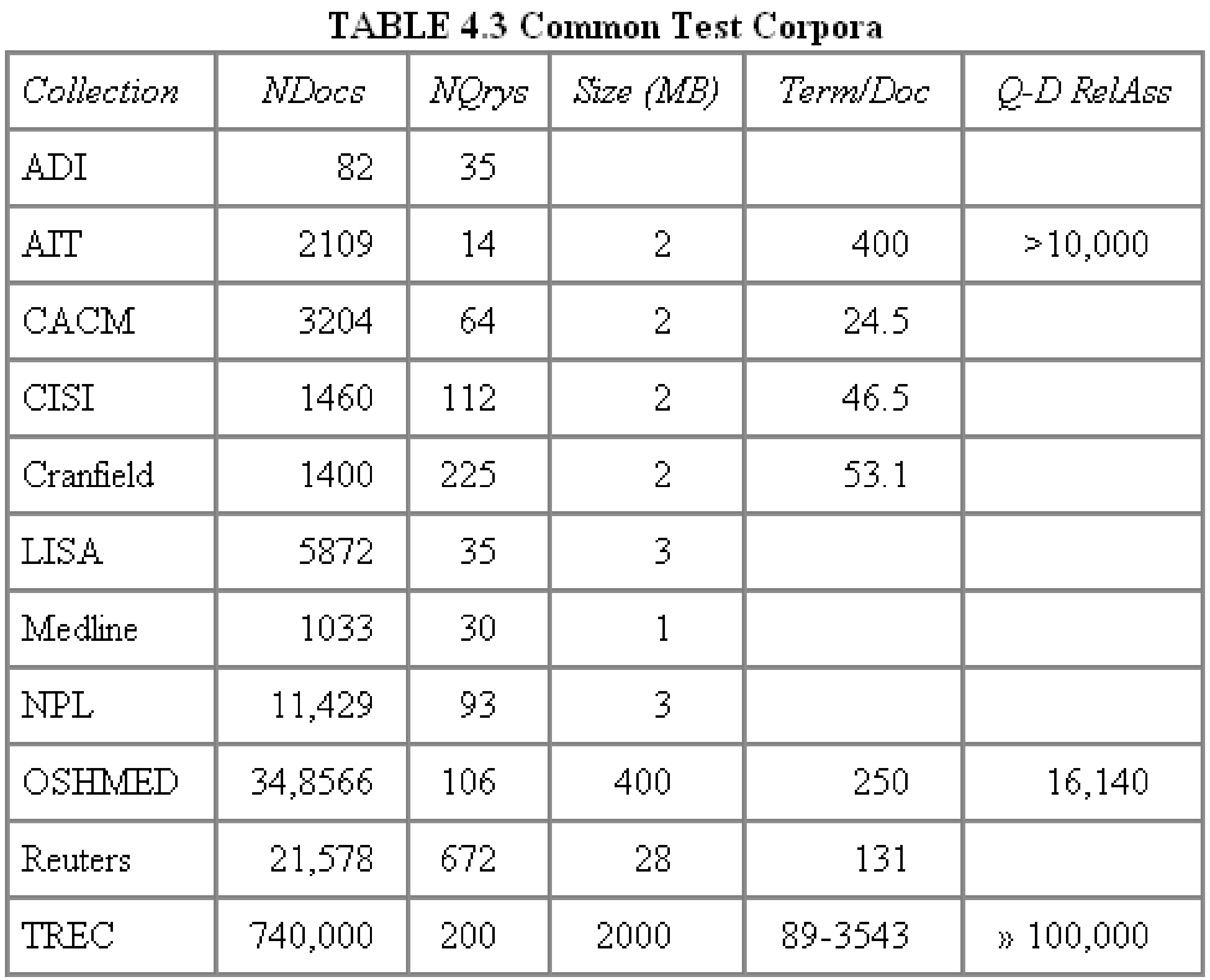


TABLE 4.3 Common Test Corpora

Collection	*NDocs*	*NQrys*	*Size (MB)*	*Term/Doc*	*Q-D RelAss*
ADI	82	35			
AIT	2109	14	2	400	>10,000
CACM	3204	64	2	24.5	
CISI	1460	112	2	46.5	
Cranfield	1400	225	2	53.1	
LISA	5872	35	3		
Medline	1033	30	1		
NPL	11,429	93	3		
OSHMED	34,8566	106	400	250	16,140
Reuters	21,578	672	28	131	
TREC	740,000	200	2000	89-3543	» 100,000

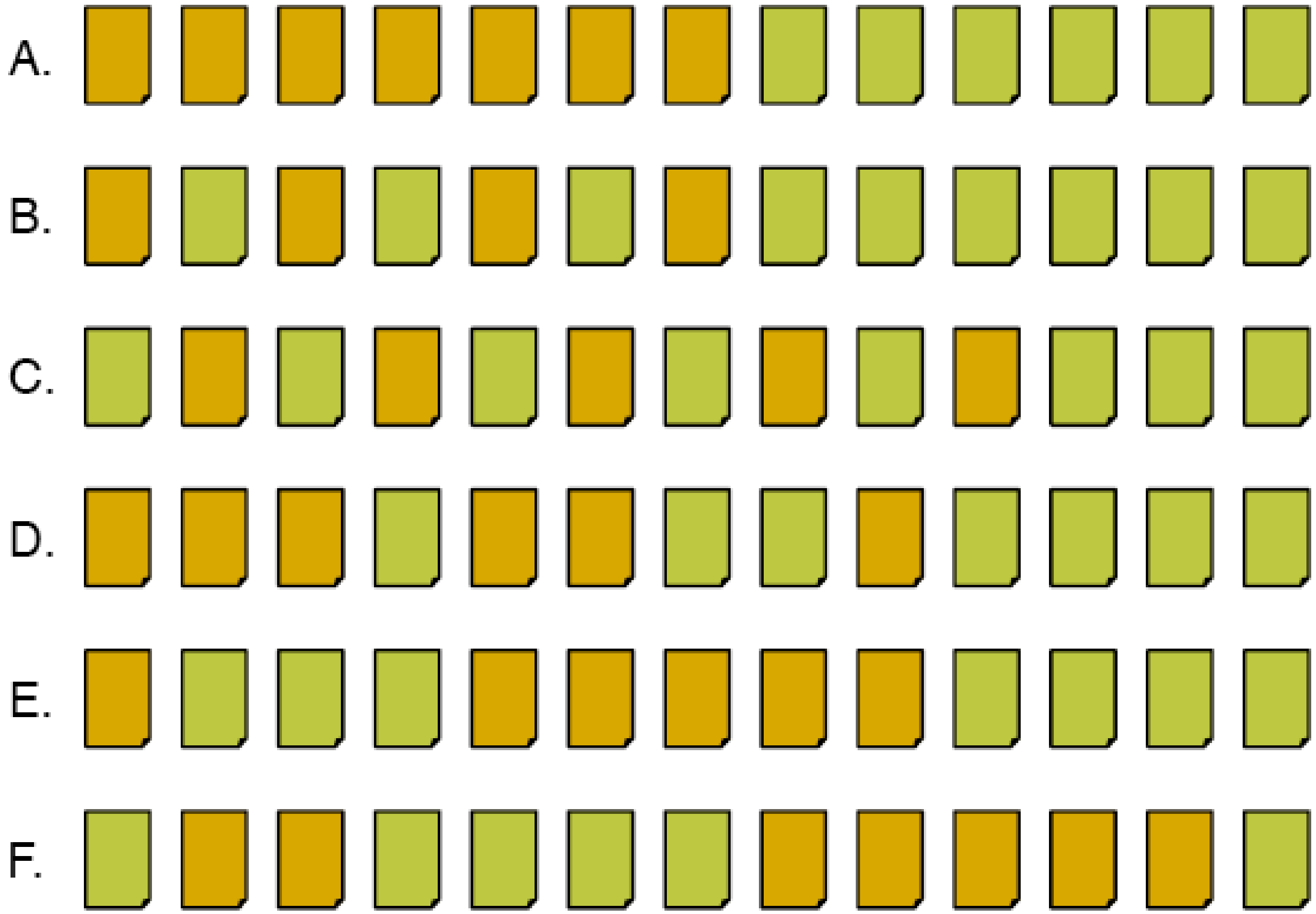
Now we have the basics of a benchmark

- Let’s review some evaluation measures
  - Precision
  - Recall
  - F measure
  - NDCG

Evaluating an IR system

- Note: the **information need** is translated into a **query**
- Relevance is assessed relative to the **information need*,*** *not* the **query**
- E.g., Information need: *My swimming pool bottom is becoming black and needs to be cleaned.*
- Query: ***pool cleaner***
- You evaluate whether the doc addresses the underlying need, not whether it has these words

Which is the best rank order?



Unranked Evaluation

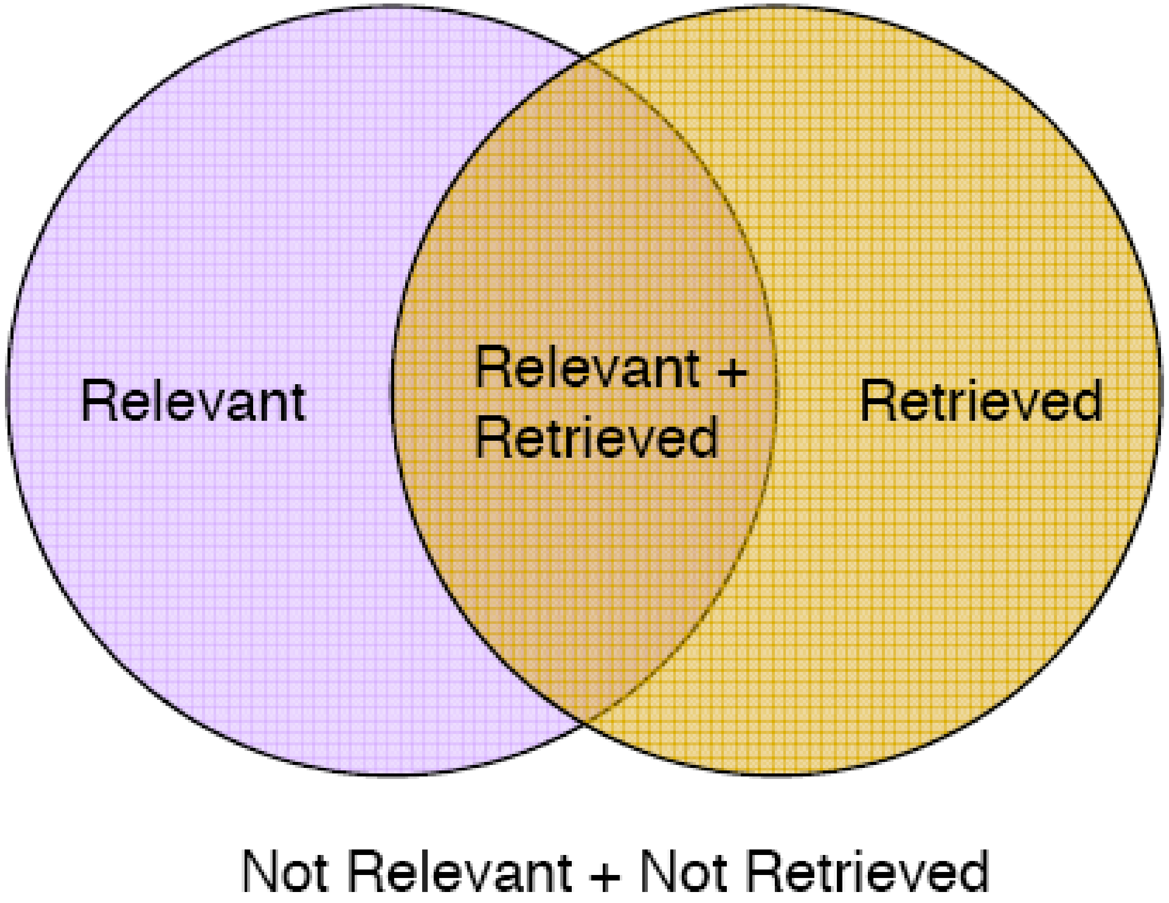
Unranked retrieval evaluation: Precision and Recall

- **Precision**: fraction of retrieved docs that are relevant = P(relevant|retrieved)
- **Recall**: fraction of relevant docs that are retrieved

  = P(retrieved|relevant)

	Relevant	Not Relevant
Retrieved	tp	fp
Not Retrieved	fn	tn

- Precision P = tp/(tp + fp)
- Recall R = tp/(tp + fn)

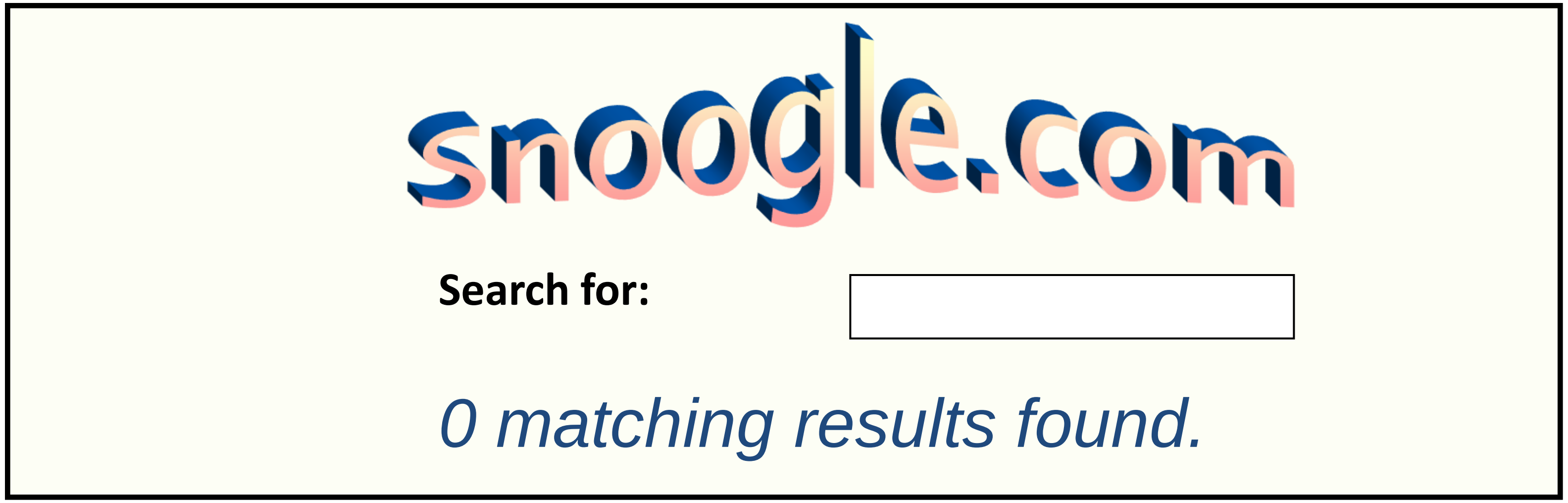

Relevant
Relevant +
Retrieved
Retrieved
Not Relevant + Not Retrieved

Accuracy

- Given a query an engine classifies each doc as “Relevant” or “Irrelevant”.
- Accuracy of an engine: the fraction of these classifications that is correct.
  - Accuracy = (tp + tn)/(tp + fp + fn + tn)
- Why is this not a very useful evaluation measure in IR?

Why not just use accuracy?

- How to build a 99.9999% accurate search engine on a low budget….



0 matching results found.

- People doing information retrieval *want to find something* and have a certain tolerance for junk.

Precision/Recall: Things to watch out for

- Should average over large number of queries
  - 100s to 1000s
- Assessments have to be binary
  - more on this later
- Heavily skewed by corpus/authorship
  - Results may not translate from one domain to another

Precision/Recall tradeoff

- You can increase recall by returning more docs.
- Recall is a non-decreasing function of the number of docs retrieved.

- A system that returns all docs has 100% recall!
- The converse is also true (usually): It’s easy to get high precision for very low recall.

- We have to make balance between precision and recall.

A combined measure: *F measure*

- Combined measure that assesses this tradeoff is F measure (weighted harmonic mean):

$$F = \frac{1}{\alpha \frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

- People usually use balanced F_1 measure
  - i.e., with $\beta = 1$ or $\alpha = ½$
- Harmonic mean is conservative average

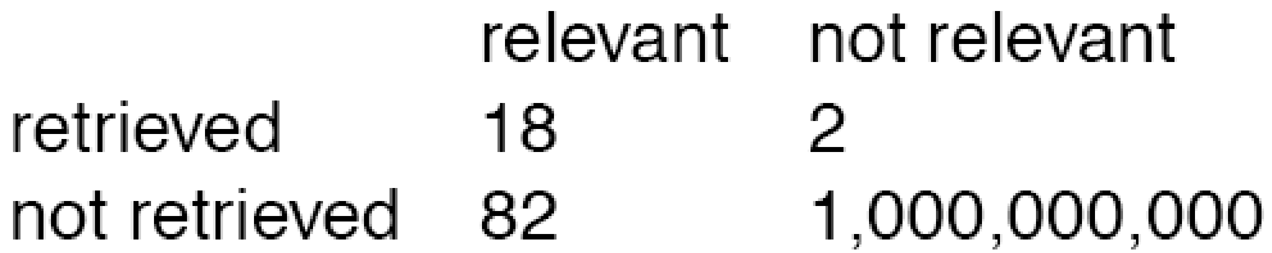
F-measure details

$\beta^2 = \frac{1-\alpha}{\alpha}$

Harmonic mean: $\frac{1}{F} = \frac{1}{2}(\frac{1}{P} + \frac{1}{R})$

$F_1 = \frac{2PR}{P+R}$

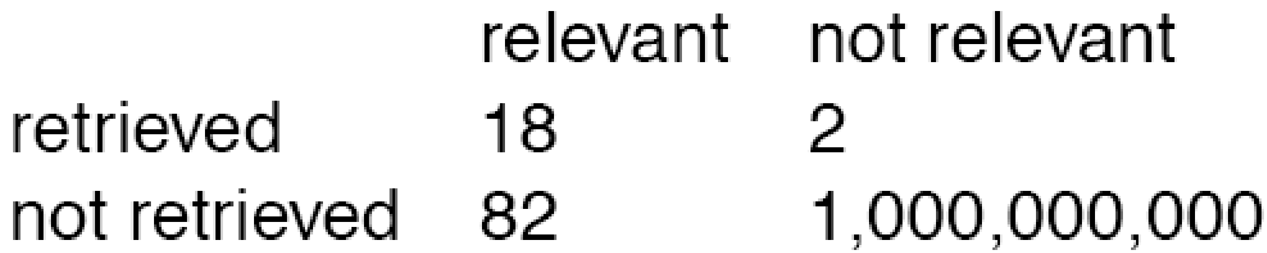
F-measure example



	relevant	not relevant
retrieved	18	2
not retrieved	82	1,000,000,000

- Precision?
- Recall?
- F?

F-measure example



	relevant	not relevant
retrieved	18	2
not retrieved	82	1,000,000,000

- precision = 18/(18+2) = 0.9
- recall = 18/(18+82) = 0.18
- F = 2PR/(P+R) = 2 * 0.9 * 0.18 / (0.9+0.18) = 0.3
- Note: F is a lot lower than AVG(P,R) = 0.54
- Number of true negatives is not factored in

Ranked Evaluation

Mean Average Precision

- Average of precision at each retrieved relevant document
- Provides a single-figure measure of quality across recall levels.

$$MAP = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{Q_j}\sum_{i=1}^{Q_j} P(doc_i)$$

with:

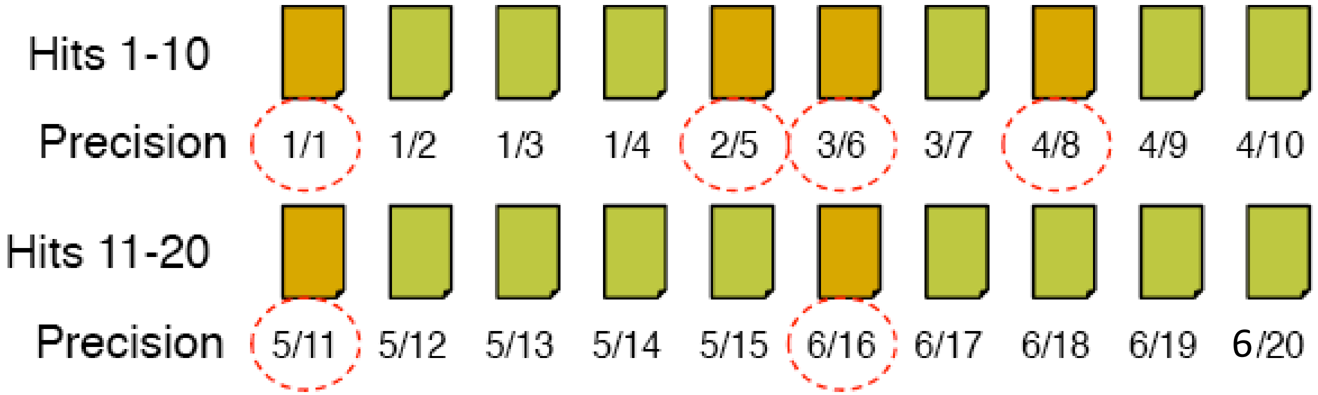
Q_j	number of relevant documents for query j
N	number of queries
$P(doc_i)$	precision at ith relevant document

Mean Average Precision

- Average of precision at each retrieved relevant document

$$MAP = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{Q_j}\sum_{i=1}^{Q_j}P(doc_i)$$

- Relevant documents not retrieved contribute 0 to score



Hits 1-10										
Precision	1/1	1/2	1/3	1/4	2/5	3/6	3/7	4/8	4/9	4/10
Hits 11-20										
Precision	5/11	5/12	5/13	5/14	5/15	6/16	6/17	6/18	6/19	6/20

Assume total of 14 relevant documents: 8 relevant documents not retrieved contribute eight zeros

MAP = .2307

Variance

- For a test collection, it is usual that a system does crummily on some information needs (e.g., MAP = 0.1) and excellently on others (e.g., MAP = 0.7)
- Indeed, it is usually the case that the variance in performance of the same system across queries is much greater than the variance of different systems on the same query.
- That is, there are easy information needs and hard ones!

Discounted Cumulative Gain (DCG)

- Popular measure for evaluating web search and related tasks

- Two assumptions:
  - Highly relevant documents are more useful than marginally relevant documents
  - the lower the ranked position of a relevant document, the less useful it is for the user, since it is less likely to be examined

DCG

- Uses *graded relevance* as a measure of usefulness, or *gain,* from examining a document
- Gain is accumulated starting at the top of the ranking and may be reduced, or *discounted*, at lower ranks
- Typical discount is 1/log *(rank)*
  - With base 2, the discount at rank 4 is 1/2, and at rank 8 it is 1/3

DCG

- What if relevance judgments are in a scale of [0,r]? r>2
- Cumulative Gain (CG) at rank n
  - Let the ratings of the n documents be r_1, r_2, …r_n (in ranked order)
  - CG = $r_1+r_2+\ldots r_n$
- Discounted Cumulative Gain (DCG) at rank n
  - DCG = $r_1 + r_2/\log_2 2 + r_3/\log_2 3 + \ldots\ r_n/\log_2 n$
  - We may use any base for the logarithm

DCG

- *DCG* is the total gain accumulated at a particular rank *p*:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2 i}$$

- Alternative formulation:

$$DCG_p = \sum_{i=1}^{p} \frac{2^{rel_i}-1}{log(1+i)}$$

- used by some web search companies
- emphasis on retrieving highly relevant documents

DCG Example

- 10 ranked documents judged on 0-3 relevance scale:

  3, 2, 3, 0, 0, 1, 2, 2, 3, 0

- discounted gain:

  3, 2/1, 3/1.59, 0, 0, 1/2.59, 2/2.81, 2/3, 3/3.17, 0

  = 3, 2, 1.89, 0, 0, 0.39, 0.71, 0.67, 0.95, 0

- DCG:

  3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61

Summarize a Ranking: NDCG

- Normalized Discounted Cumulative Gain (NDCG) at rank *n*
  - Normalize DCG at rank *n* by the DCG value at rank *n* of the ideal ranking
  - The ideal ranking would first return the documents with the highest relevance level, then the next highest relevance level, etc
- Normalization useful for contrasting queries with varying numbers of relevant results

- NDCG is now quite popular in evaluating Web search

NDCG Example

- 10 ranked documents judged on 0-3 relevance scale:

  3, 2, 3, 0, 0, 1, 2, 2, 3, 0

- The Best order

  3, 3, 3, 2, 2, 2, 1, 0, 0, 0

- 3, 3/1, 3/1.59, 2/2, 2/2.32, 2/2.59, 1/2.81, 0, 0, 0

  = 3, 3, 1.89, 1, 0.86, 0.77, 0.36, 0, 0, 0

- DCG of Ground Truth (MaxDCG):

  3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10.88

NDCG Example

- DCG of Ground Truth (MaxDCG):

  3, 6, 7.89, 8.89, 9.75, 10.52, 10.88, 10.88, 10.88, 10.88

- DCG of a search engine:

  3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61

- NDCG@5:
  - 6.89/9.75 = 0.71
- NDCG@10:
  - 9.61/10.88 = 0.88

NDCG - Example

4 documents: d_1, d_2, d_3, d_4

i	Ground Truth		Ranking Function$_1$		Ranking Function$_2$	
	Document Order	r_i	Document Order	r_i	Document Order	r_i
1	d4	2	d3	2	d3	2
2	d3	2	d4	2	d2	1
3	d2	1	d2	1	d4	2
4	d1	0	d1	0	d1	0
	$NDCG_{GT}$=1.00		$NDCG_{RF1}$=1.00		$NDCG_{RF2}$=0.9203	

$$DCG_{GT} = 2 + \left(\frac{2}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.6309$$

$$DCG_{RF1} = 2 + \left(\frac{2}{\log_2 2} + \frac{1}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.6309$$

$$DCG_{RF2} = 2 + \left(\frac{1}{\log_2 2} + \frac{2}{\log_2 3} + \frac{0}{\log_2 4} \right) = 4.2619$$

$$MaxDCG = DCG_{GT} = 4.6309$$

So far…

- Unranked Evaluation
  - Precision
  - Recall
  - F-measure

- Ranked Evaluation
  - Mean Average Precision
  - NDCG

Human judgments are

- Expensive
- Inconsistent
  - Between raters
  - Over time
- Not always representative of “real users”
- So – what alternatives do we have?

Using User Clicks

Comparing two rankings via clicks

Query: [support vector machines]

Ranking A	Ranking B
Kernel machines	Kernel machines
SVM-light	SVMs
Lucent SVM demo	Intro to SVMs
Royal Holl. SVM	Archives of SVM
SVM software	SVM-light
SVM tutorial	SVM software

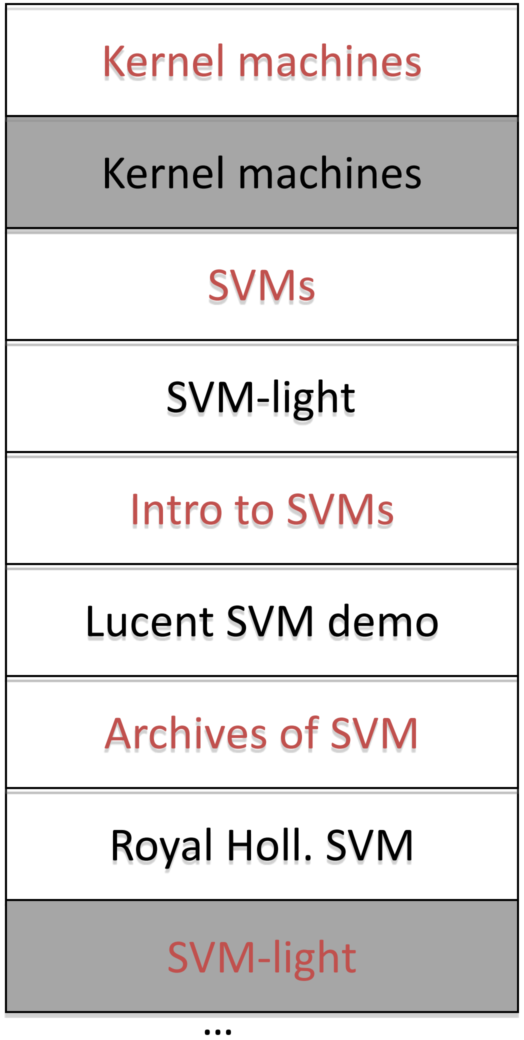
Interleave the two rankings

This interleaving starts with B

Kernel machines
Kernel machines
SVMs
SVM-light
Intro to SVMs
Lucent SVM demo
Archives of SVM
Royal Holl. SVM
SVM-light

…

Remove duplicate results



Kernel machines

Kernel machines

SVMs

SVM-light

Intro to SVMs

Lucent SVM demo

Archives of SVM

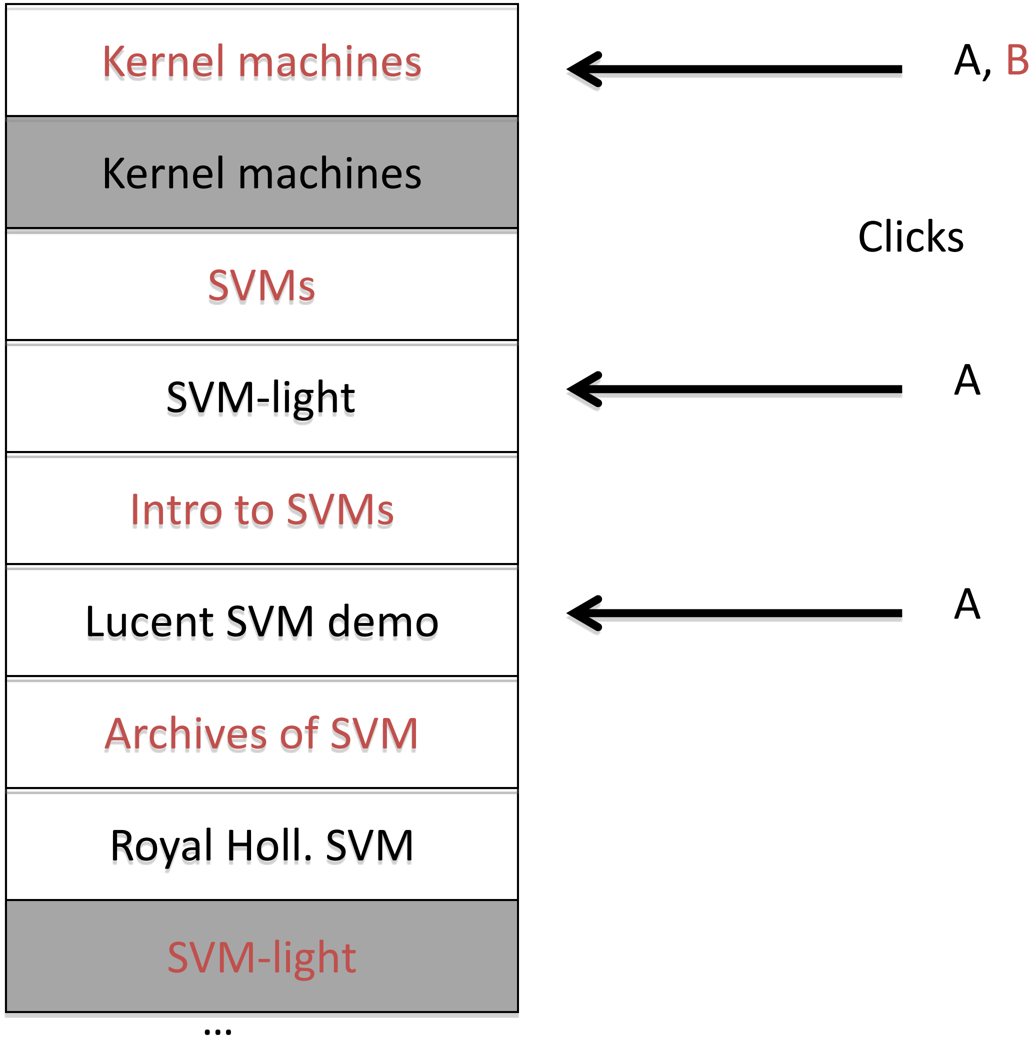
Royal Holl. SVM

SVM-light

…

Count user clicks

Ranking A: 3
Ranking B: 1



Kernel machines

Kernel machines

SVMs

SVM-light

Intro to SVMs

Lucent SVM demo

Archives of SVM

Royal Holl. SVM

SVM-light

...

A, B

Clicks

A

A

Interleaved ranking

- Present interleaved ranking to users
  - Start randomly with ranking A or ranking B to evens out presentation bias

- Count clicks on results from A versus results from B

- Better ranking will (on average) get more clicks

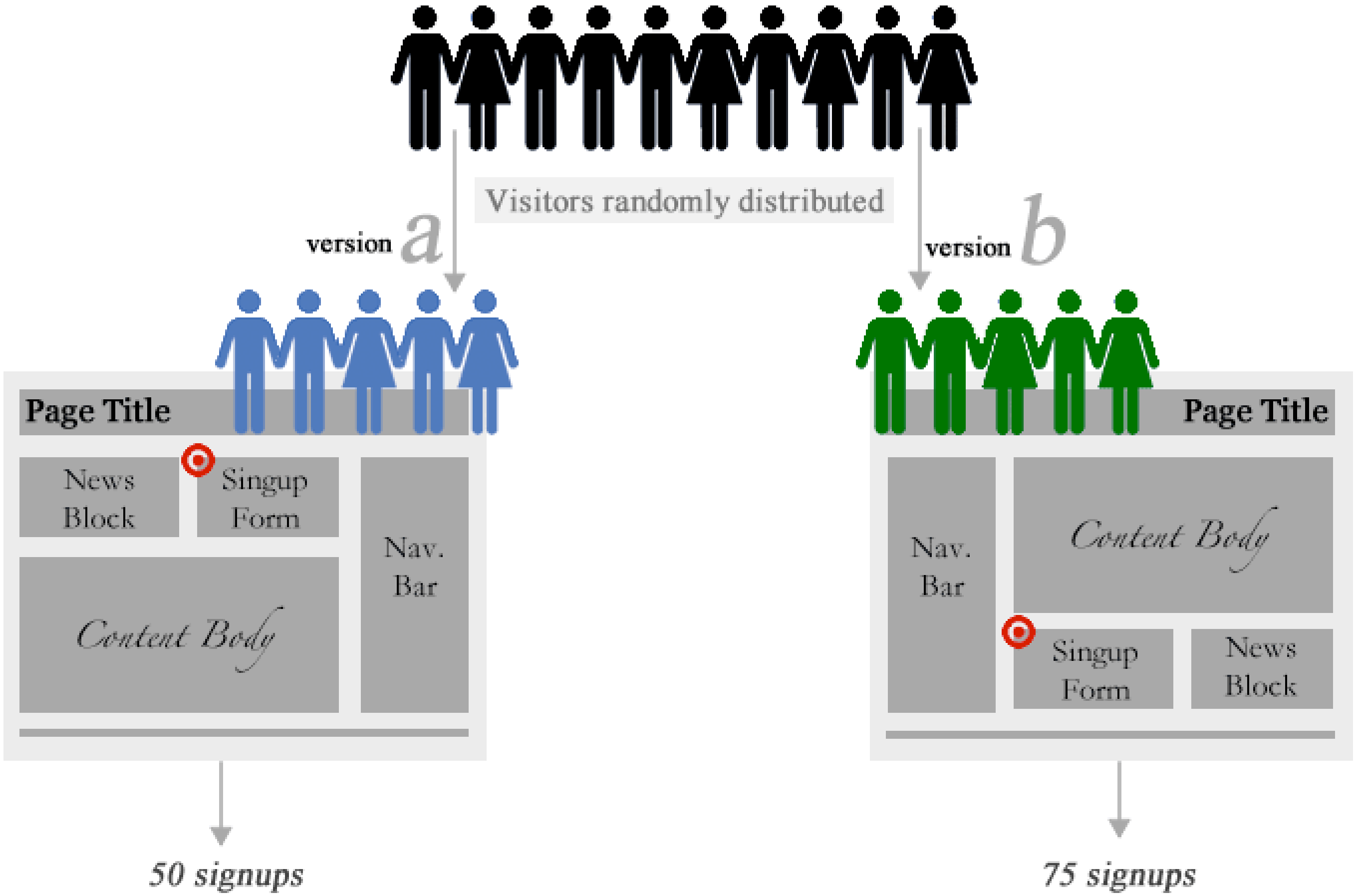
A/B Testing: Randomized Controlled Experiments

http://www.wired.com/2012/04/ff_abtesting/all/

A/B testing at web search engines

- Purpose: Test a single innovation
- Prerequisite: You have a large search engine up and running.
- Have most users use old system
- Divert a small proportion of traffic (e.g., 1%) to an experiment to evaluate an innovation

Another example of A/B testing



Visitors randomly distributed

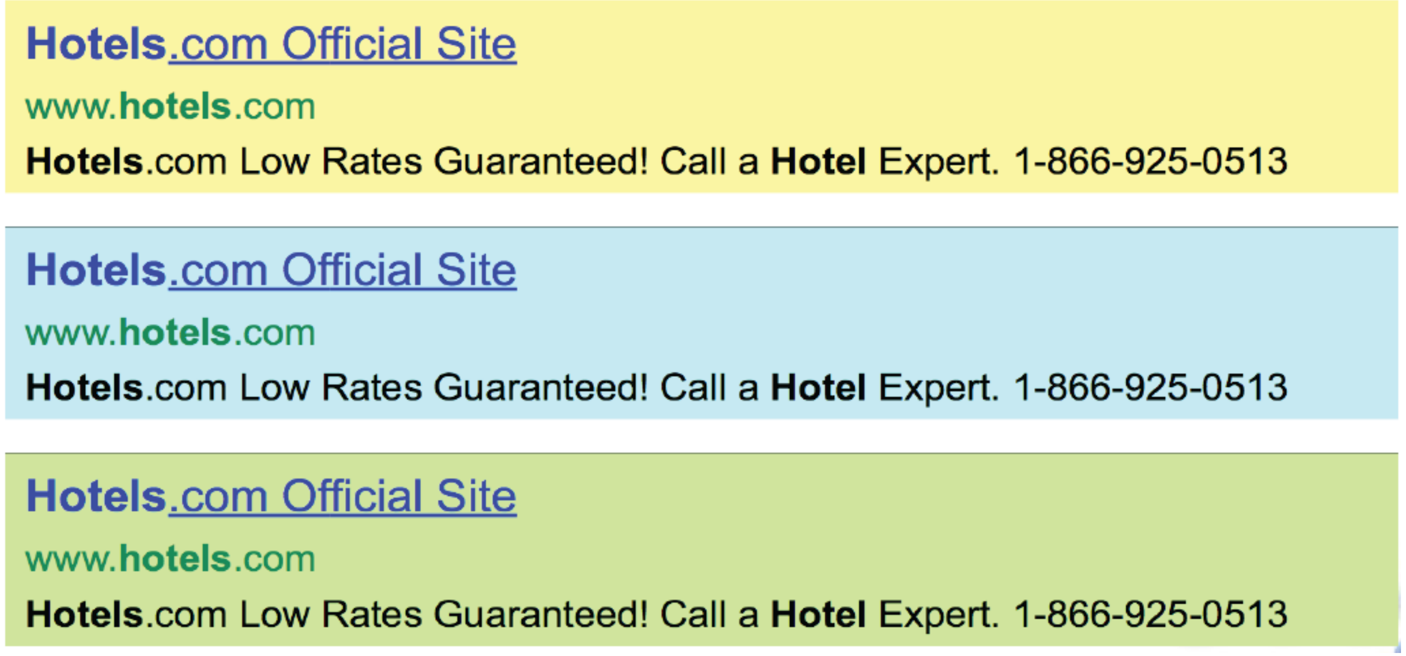
50 signups

75 signups

Version B is better than version A

Why run experiments?

- Gathers data on impact of changes
  - How do users behave differently, if at all?
- Data-driven decisions:
  - UI



Hotels.com Official Site

www.**hotels**.com

Hotels.com Low Rates Guaranteed! Call a **Hotel** Expert. 1-866-925-0513

Hotels.com Official Site

www.**hotels**.com

Hotels.com Low Rates Guaranteed! Call a **Hotel** Expert. 1-866-925-0513

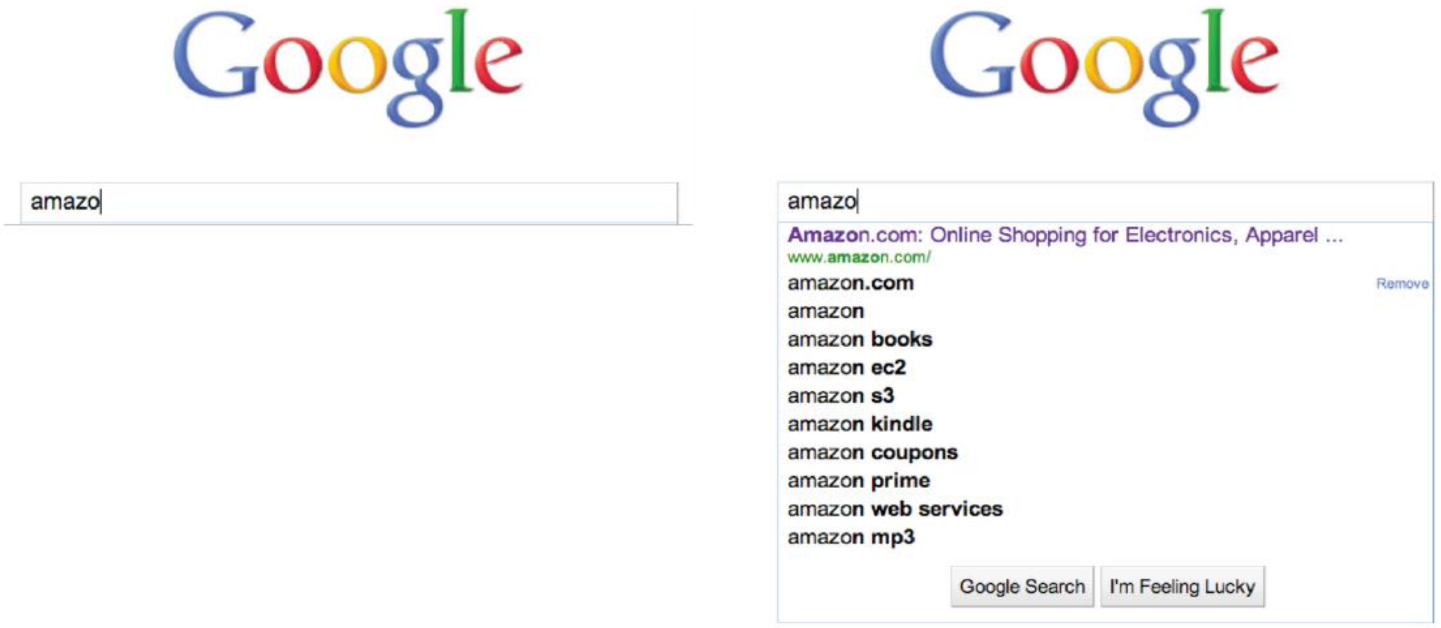
Hotels.com Official Site

www.**hotels**.com

Hotels.com Low Rates Guaranteed! Call a **Hotel** Expert. 1-866-925-0513

Why run experiments?

- Gathers data on impact of changes
  - How do users behave differently, if at all?
- Data-driven decisions:
  - UI



amazo

amazo

Amazon.com: Online Shopping for Electronics, Apparel ...
www.**amazo**n.com/

amazon**.com**
amazon
amazon **books**
amazon **ec2**
amazon **s3**
amazon **kindle**
amazon **coupons**
amazon **prime**
amazon **web services**
amazon **mp3**

Why run experiments?

- Gathers data on impact of changes
  - How do users behave differently, if at all?
- Data-driven decisions:
  - UI
  - Algorithms, e.g., CTR prediction (Click-Through Rate)
    - How many passes over the data
    - Data range
    - Different machine learning algorithms

Search Engine Optimization

SEO

- *Search Engine Optimization:*
    - "Tuning" your web page to rank highly in the algorithmic search results for select keywords
    - Alternative to paying for placement
    - Thus, intrinsically a marketing function
- Performed by companies, webmasters and consultants ("Search engine optimizers") for their clients
- Some perfectly legitimate, some very shady

Search engine optimization (Spam)

- Motives
  - Commercial, political, religious, lobbies
  - Promotion funded by advertising budget
- Operators
  - Contractors (Search Engine Optimizers) for lobbies, companies
  - Web masters
  - Hosting services
- Forums
  - E.g., Web master world (www.webmasterworld.com)
    - Search engine specific tricks
    - Discussions about academic papers ☺

Simplest forms

- First generation engines relied heavily on *tf/idf*
  - The top-ranked pages for the query **maui resort** were the ones containing the most **maui**’s and **resort**’s
- SEOs responded with dense repetitions of chosen terms
  - e.g., **maui resort maui resort maui resort**
  - Often, the repetitions would be in the same color as the background of the web page
    - Repeated terms got indexed by crawlers
    - But not visible to humans on browsers

Pure word density cannot be trusted as an IR signal

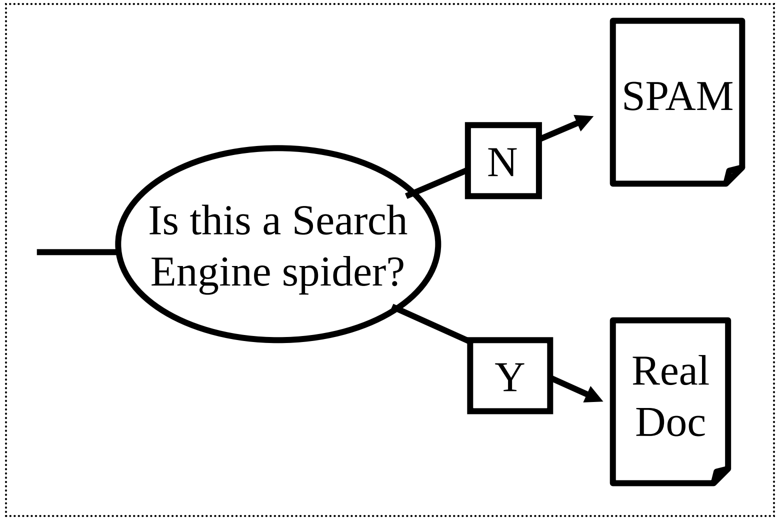
Variants of keyword stuffing

- Misleading meta-tags, excessive repetition
- Hidden text with colors, style sheet tricks, etc.

Meta-Tags =
"… London hotels, hotel, holiday inn, hilton, discount,
booking, reservation, sex, mp3, britney spears, viagra, …"

Cloaking

- Serve fake content to search engine spider
- DNS cloaking: Switch IP address. Impersonate



More spam techniques

- **Doorway pages**
  - Pages optimized for a single keyword that re-direct to the real target page
- **Link spamming**
  - Mutual admiration societies, hidden links, awards
  - *Domain flooding:* numerous domains that point or re-direct to a target page
- **Robots**
  - Fake query stream – rank checking programs
    - “Curve-fit” ranking programs of search engines
  - Millions of submissions via Add-Url

The war against spam

- Quality signals - Prefer authoritative pages based on:
  - Votes from authors (linkage signals)
  - Votes from users (usage signals)
- Limits on meta-keywords
- Robust link analysis
  - Ignore statistically implausible linkage (or text)
  - Use link analysis to detect spammers (guilt by association)
- Spam recognition by machine learning
  - Training set based on known spam
- Editorial intervention
  - Blacklists
  - Top queries audited
  - Complaints addressed
  - Suspect pattern detection