COMP0084 Information Retrieval and Data Mining (IRDM)

Content independent ranking: PageRank and HITS

Professor Ingemar J. Cox

Computer Science, UCL

Recap

Goal of information retrieval is to find documents that are relevant to a user’s query.

How is relevance defined?

Recap

Previously we have focused on content-dependent models of relevance.

What are their limitations?

Recap

All documents are not created equal.

Credibility

Trust

Authority

Hypertext and links

Can the link structure of the web be used to help infer the relevance of pages?

Hypertext and links

Applicable to many areas:

Search

Email

Social networks

Intuition

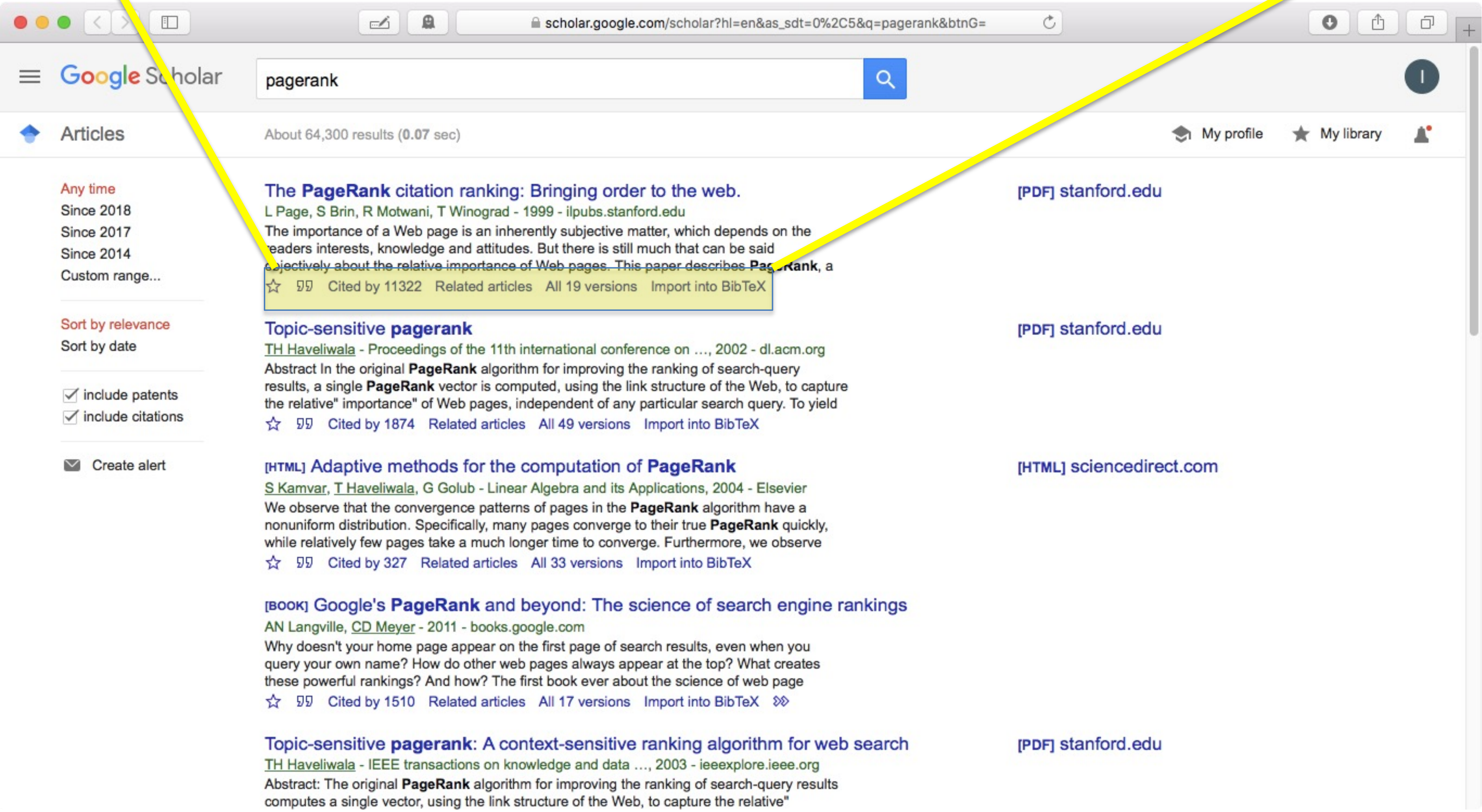
Hypertext links are analogous to citations

Important scientific papers receive a lot of citations

Papers cited by important papers are also likely to be important

Intuition

Cited by 11322 Related articles All 19 versions Import into BibTeX



About 64,300 results (**0.07** sec)

The **PageRank** citation ranking: Bringing order to the web.
L Page, S Brin, R Motwani, T Winograd - 1999 - ilpubs.stanford.edu
The importance of a Web page is an inherently subjective matter, which depends on the readers interests, knowledge and attitudes. But there is still much that can be said objectively about the relative importance of Web pages. This paper describes **PageRank**, a
☆ 99 Cited by 11322 Related articles All 19 versions Import into BibTeX
[PDF] stanford.edu

Topic-sensitive **pagerank**
TH Haveliwala - Proceedings of the 11th international conference on …, 2002 - dl.acm.org
Abstract In the original **PageRank** algorithm for improving the ranking of search-query results, a single **PageRank** vector is computed, using the link structure of the Web, to capture the relative" importance" of Web pages, independent of any particular search query. To yield
☆ 99 Cited by 1874 Related articles All 49 versions Import into BibTeX
[PDF] stanford.edu

[HTML] Adaptive methods for the computation of **PageRank**
S Kamvar, T Haveliwala, G Golub - Linear Algebra and its Applications, 2004 - Elsevier
We observe that the convergence patterns of pages in the **PageRank** algorithm have a nonuniform distribution. Specifically, many pages converge to their true **PageRank** quickly, while relatively few pages take a much longer time to converge. Furthermore, we observe
☆ 99 Cited by 327 Related articles All 33 versions Import into BibTeX
[HTML] sciencedirect.com

[BOOK] Google's **PageRank** and beyond: The science of search engine rankings
AN Langville, CD Meyer - 2011 - books.google.com
Why doesn't your home page appear on the first page of search results, even when you query your own name? How do other web pages always appear at the top? What creates these powerful rankings? And how? The first book ever about the science of web page
☆ 99 Cited by 1510 Related articles All 17 versions Import into BibTeX »

Topic-sensitive **pagerank**: A context-sensitive ranking algorithm for web search
TH Haveliwala - IEEE transactions on knowledge and data …, 2003 - ieeexplore.ieee.org
Abstract: The original **PageRank** algorithm for improving the ranking of search-query results computes a single vector, using the link structure of the Web, to capture the relative"
[PDF] stanford.edu

Citation analysis

Citation frequency

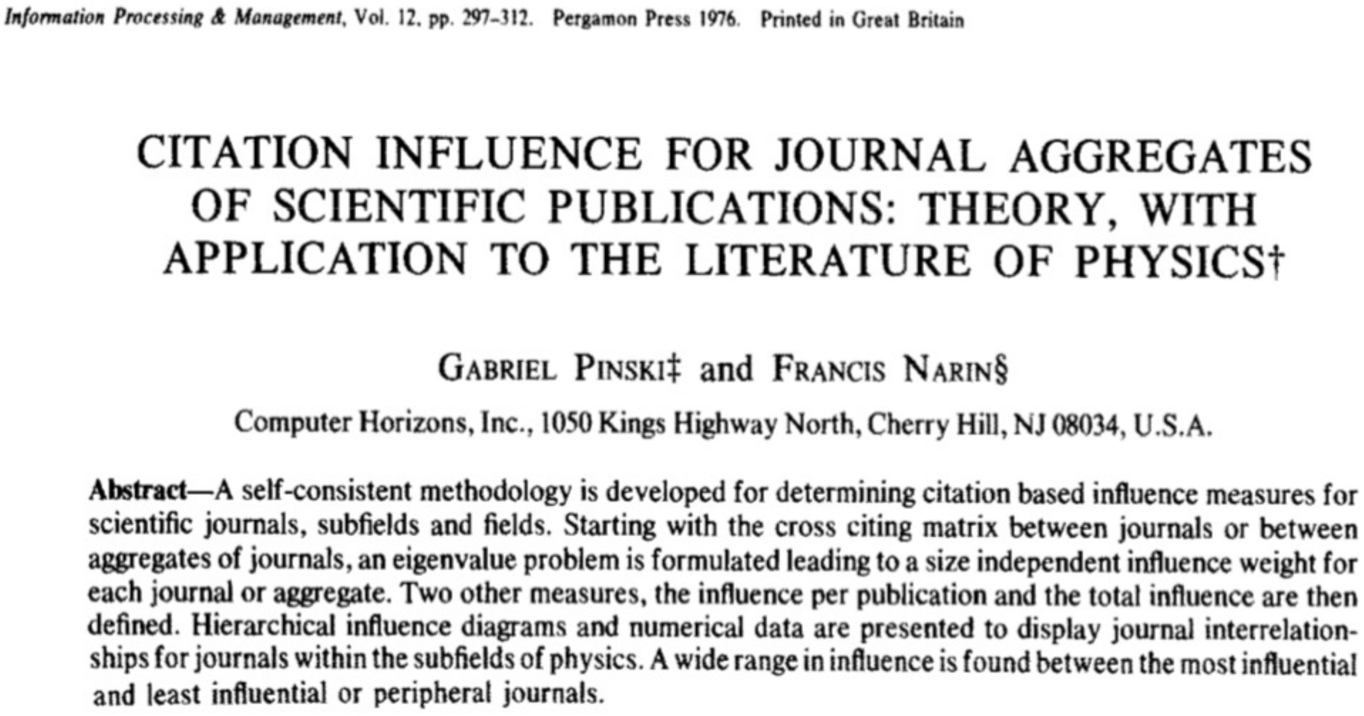
Every article gets one vote

But all articles are not equal

Weighted vote based on impact

Pagerank related to work by Pinski and Narin

Citation analysis



Information Processing & Management, Vol. 12, pp. 297–312. Pergamon Press 1976. Printed in Great Britain

CITATION INFLUENCE FOR JOURNAL AGGREGATES OF SCIENTIFIC PUBLICATIONS: THEORY, WITH APPLICATION TO THE LITERATURE OF PHYSICS†

GABRIEL PINSKI‡ and FRANCIS NARIN§

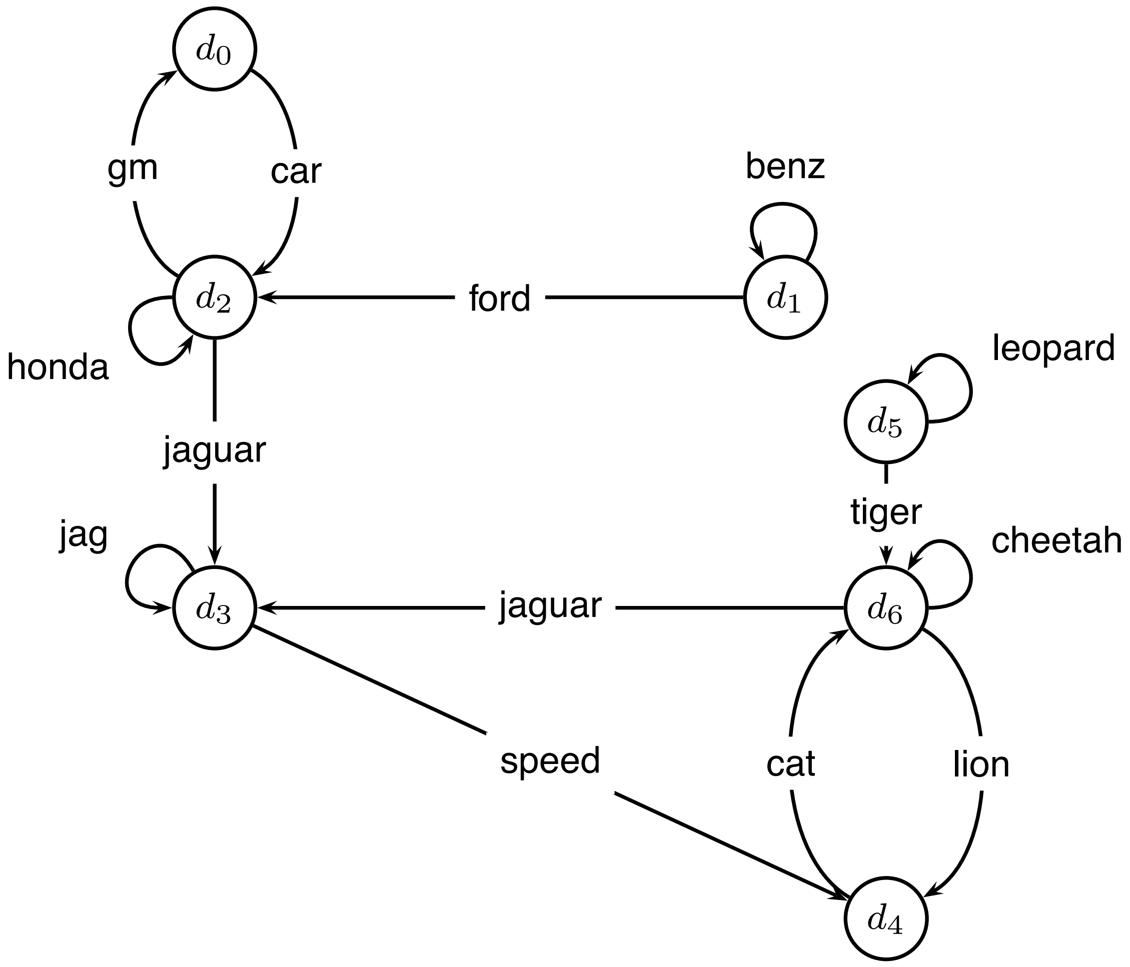
Computer Horizons, Inc., 1050 Kings Highway North, Cherry Hill, NJ 08034, U.S.A.

Abstract—A self-consistent methodology is developed for determining citation based influence measures for scientific journals, subfields and fields. Starting with the cross citing matrix between journals or between aggregates of journals, an eigenvalue problem is formulated leading to a size independent influence weight for each journal or aggregate. Two other measures, the influence per publication and the total influence are then defined. Hierarchical influence diagrams and numerical data are presented to display journal interrelationships for journals within the subfields of physics. A wide range in influence is found between the most influential and least influential or peripheral journals.

Link analysis

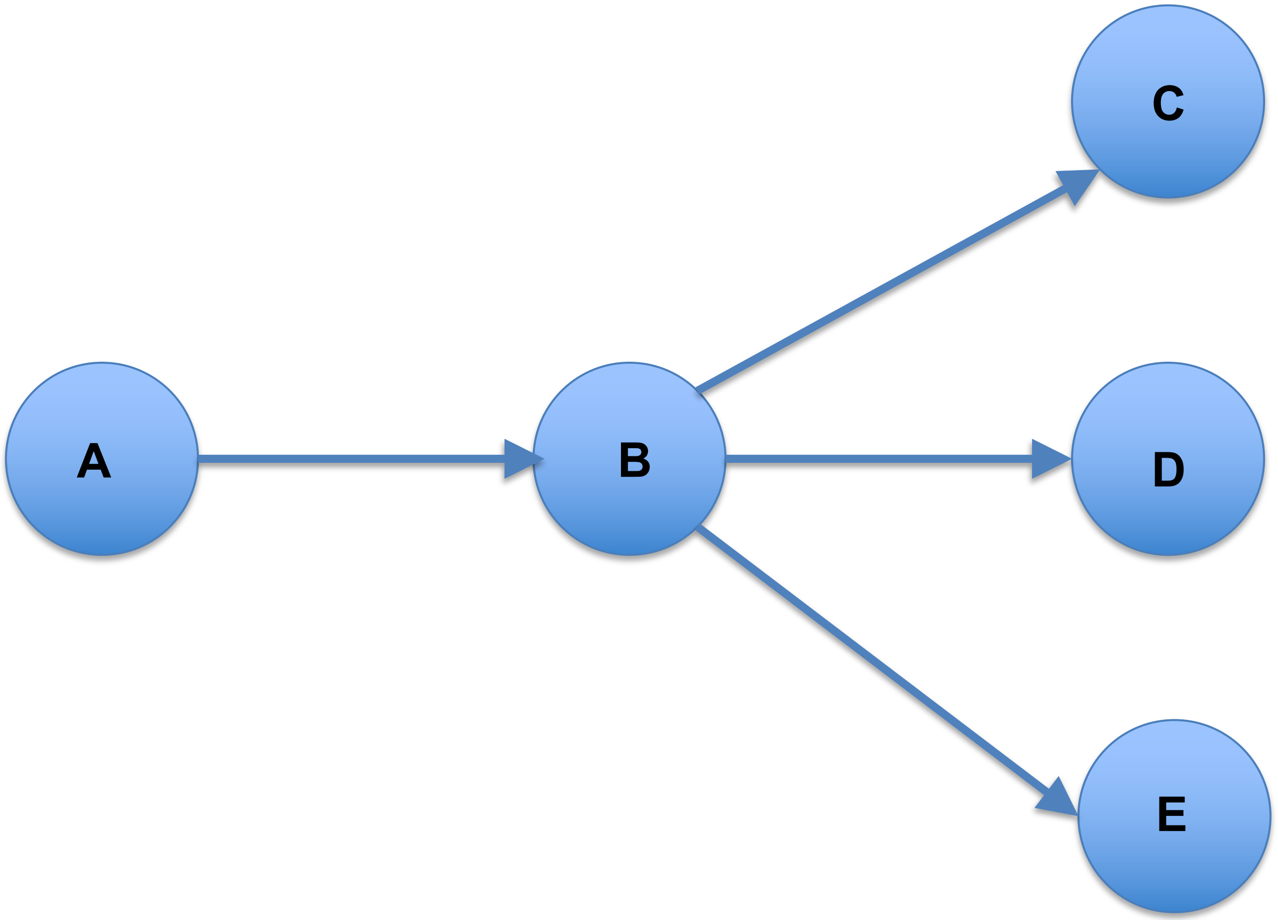
How can we score hypertext pages based on links?

The Web as a directed graph



–

The Web as a directed graph

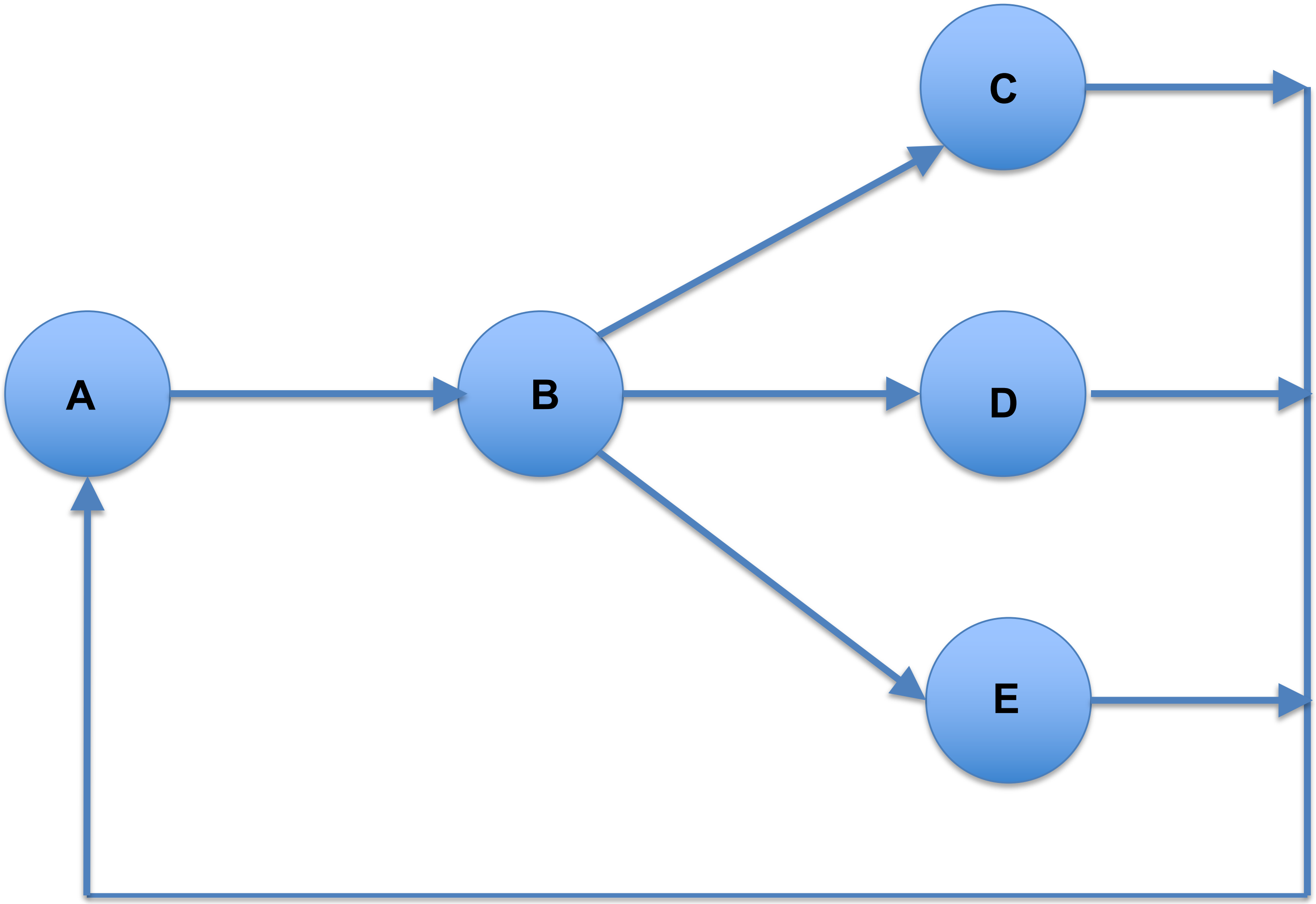


The web as a directed graph

What is the probability of moving from B to C?

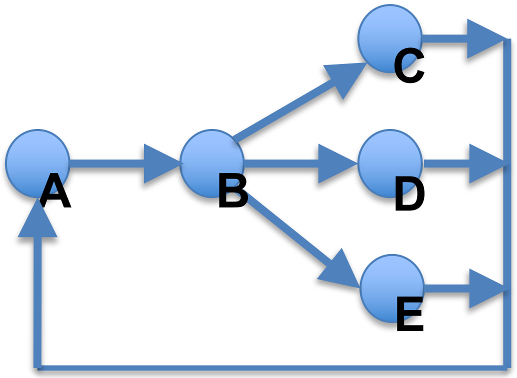
What is the probability of moving from A to B?

The Web as a directed graph



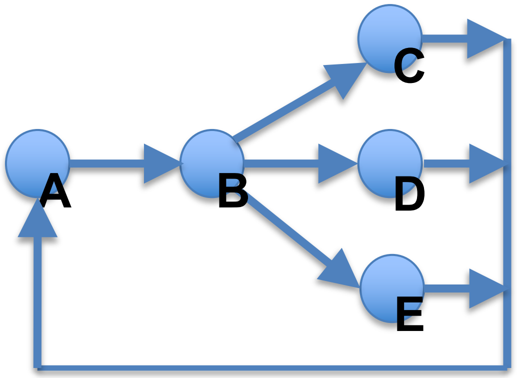
The Web as a directed graph

What is the probability that you will be at Node A?



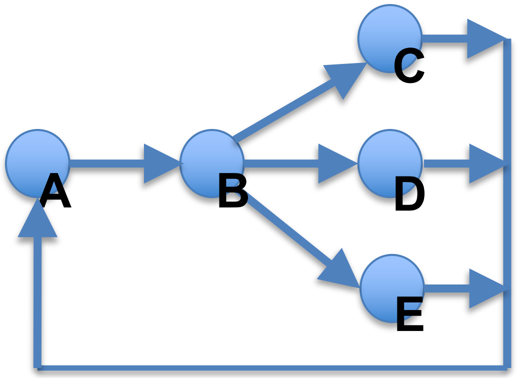
The Web as a directed graph

What is the probability that you will be at Node B?



The Web as a directed graph

What is the probability that you will be at Node C, D, or E?



Long-term visit rate

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.

Long-term visit rate

A graph may not have well-defined long-term visit rates.

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

Long-term visit rate

- Exists if there is a positive integer T_0 such that for all pairs of states i and j, then if started at time 0 in state i, then for all $t>T_0$, there is a probability > 0 of being in state j.
- Irreducibility and aperiodicity

Markov chains

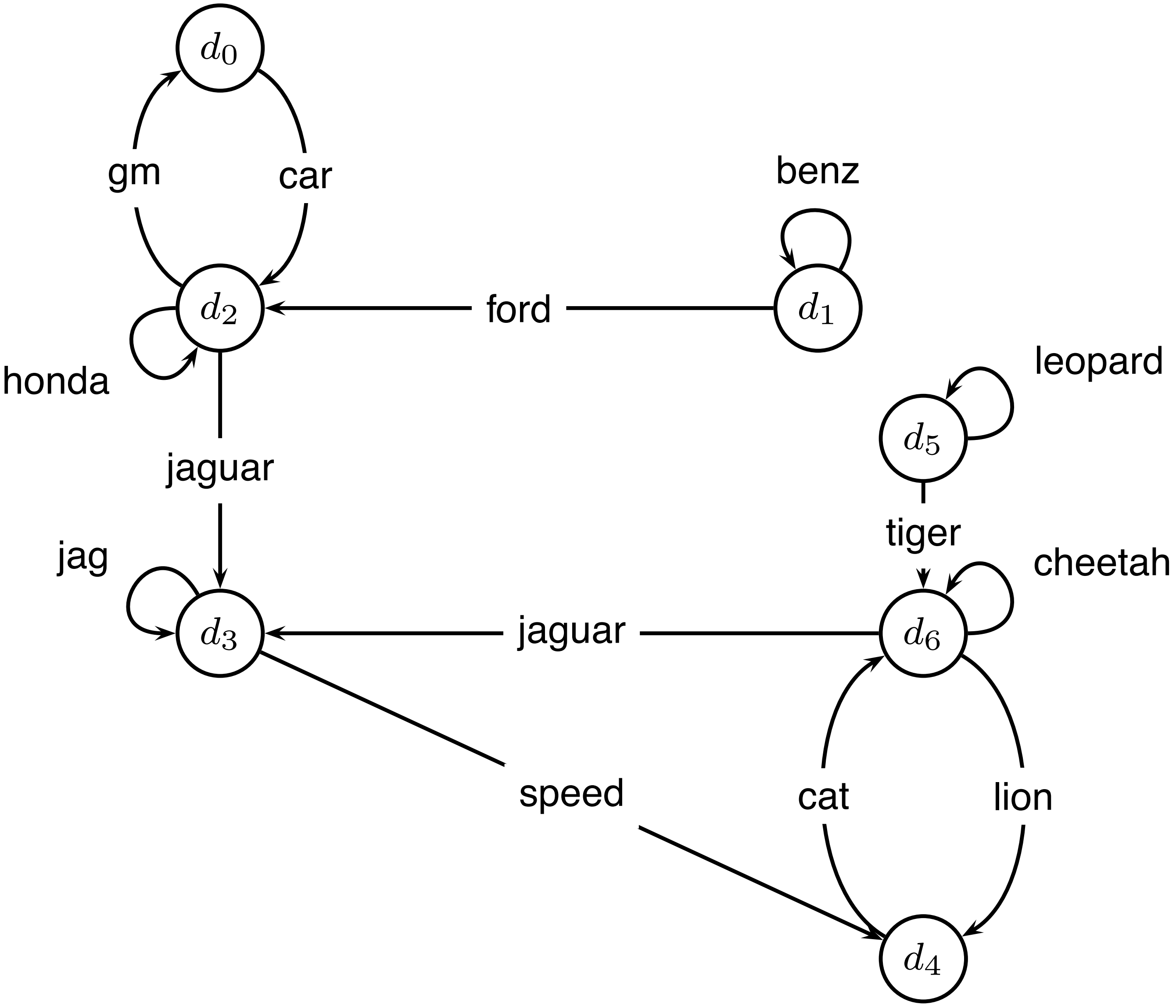
- A Markov chain consists of *n* states, plus an *nxn* transition probability matrix **P**.
  - state = page
- At each step, we are in exactly one of the states
- For *$1 \le i,j \le n$,* the matrix entry P_{ij} tells us the probability of *j* being the next state, given we are currently in state *i*.

Markov chains

Clearly, for all *i*, $\sum_{j=1}^{n} P_{ij} = 1.$

Markov chains are abstractions of random walks.

Markov chains: Example



Adjacency matrix

	DO	D1	D2	D3	D4	D5	D6
D0	0	0	1	0	0	0	0
D1	0	1	1	0	0	0	0
D2	1	0	1	1	0	0	0
D3	0	0	0	1	1	0	0
D4	0	0	0	0	0	0	1
D5	0	0	0	0	0	1	1
D6	0	0	0	1	1	0	1

Transition probability matrix, P

	DO	D1	D2	D3	D4	D5	D6
D0	0	0	1.0	0	0	0	0
D1	0	0.5	0.5	0	0	0	0
D2	0.33	0	0.33	0.33	0	0	0
D3	0	0	0	0.5	0.5	0	0
D4	0	0	0	0	0	0	1.0
D5	0	0	0	0	0	0.5	0.5
D6	0	0	0	0.33	0.33	0	0.33

Teleporting

- Example does not meet requirements.
- Solution is "teleporting"
- At a dead end, jump to a random web page.
- At any non-dead end, with certain probability, say 10%, jump to a random web page.
  - With remaining probability (90%), go out on a random out-link.
  - 10% is a parameter

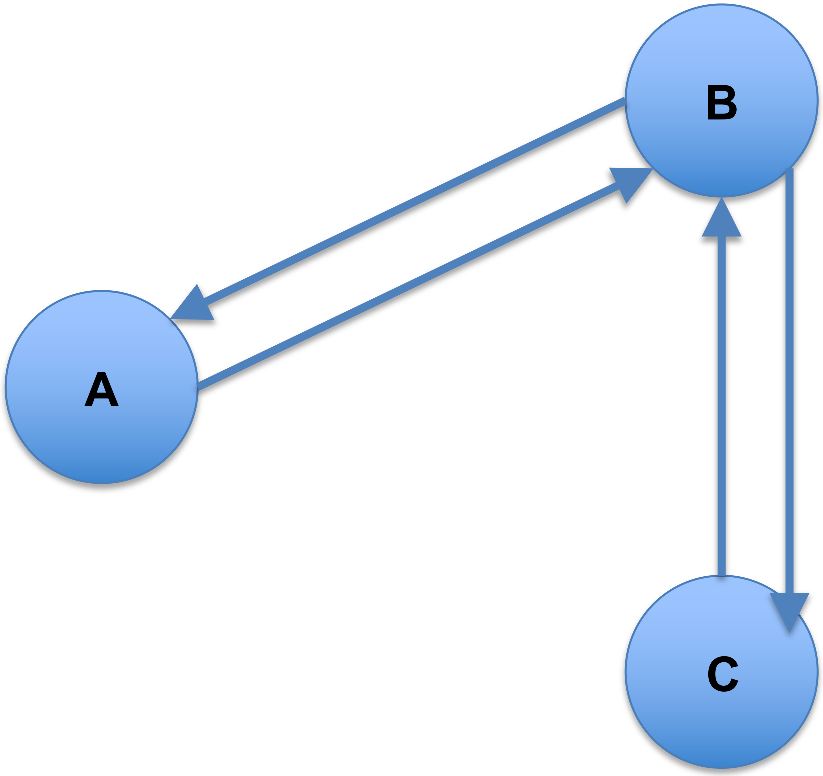
Transition probability matrix, P, with teleporting (0.14)

	DO	D1	D2	D3	D4	D5	D6
D0	0.02	0.02	0.88	0.02	0.02	0.02	0.02
D1	0.02	0.45	0.45	0.02	0.02	0.02	0.02
D2	0.31	0.02	0.31	0.31	0.02	0.02	0.02
D3	0.02	0.02	0.02	0.45	0.45	0.02	0.02
D4	0.02	0.02	0.02	0.02	0.02	0.02	0.88
D5	0.02	0.02	0.02	0.02	0.02	0.45	0.45
D6	0.02	0.02	0.02	0.31	0.31	0.02	0.31

Transition probability matrix

- Start with the N × N adjacency matrix A
- If row of A is all zeros, then add 1/N to each entry in the row
- For all other rows:
  - Divide each 1 in A by number of 1's in row
- Multiply resulting matrix by 1-α, then
- Add α/N to all elements of resulting matrix

Example



Transition probability matrix

- Start with the N × N adjacency matrix A

Adjacency matrix

	A	B	C
A	0	1	0
B	1	0	1
C	0	1	0

Transition probability matrix

- Start with the N × N adjacency matrix A
- If row of A is all zeros, then add 1/N to each entry in the row

Adjacency matrix

	A	B	C
A	0	1	0
B	1	0	1
C	0	1	0

Transition probability matrix

- Start with the N × N adjacency matrix A
- If row of A is all zeros, then add 1/N to each entry in the row
- For all other rows:
  - Divide each 1 in A by number of 1’s in row

Adjacency matrix

	A	B	C
A	0	1	0
B	0.5	0	0.5
C	0	1	0

Transition probability matrix

- Start with the N × N adjacency matrix A
- If row of A is all zeros, then add 1/N to each entry in the row
- For all other rows:
  - Divide each 1 in A by number of 1's in row
- Multiply resulting matrix by 1-α, (α=0.5)

Adjacency matrix

	A	B	C
A	0	0.5	0
B	0.25	0	0.25
C	0	0.5	0

Transition probability matrix

- Start with the N × N adjacency matrix A
- If row of A is all zeros, then add 1/N to each entry in the row
- For all other rows:
  - Divide each 1 in A by number of 1’s in row
- Multiply resulting matrix by 1-α, then
- Add α/N to all elements of resulting matrix (0.5/3=0.17)

Transition probability matrix

	A	B	C
A	0.17	0.67	0.17
B	0.42	0.17	0.42
C	0.17	0.67	0.17

Long-term visit rate

Let probability (row) vector **x** = (x_1, … x_n) be the probability of a "surfer" being at state (page) x_i

(0, 0, 0.5, 0.5, 0, 0)

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

Long-term visit rate

If the probability vector is $\mathbf{x} = (x_1, \ldots\ x_n)$ at time t, what is it at the next step, t+1?

$$x(t+1) = x(t).P$$

$$x(t+2)=x(t+1).P=x(t).P^2$$

Long-term visit rate

- Let **a** = $(a_1, \ldots a_n)$ denote the row vector of steady-state probabilities
  - a_i is the long-term visit rate (or PageRank) of page i.
- So we can think of PageRank as a very long vector – one entry per page.

PageRank

If our current position is described by **a**, then the next step is distributed as **aP.** But **a** is the steady state, so **a=aP**.

We can measure **P**. We need to solve for **a**

a is the principal left eigenvector of **P**

PageRank computation

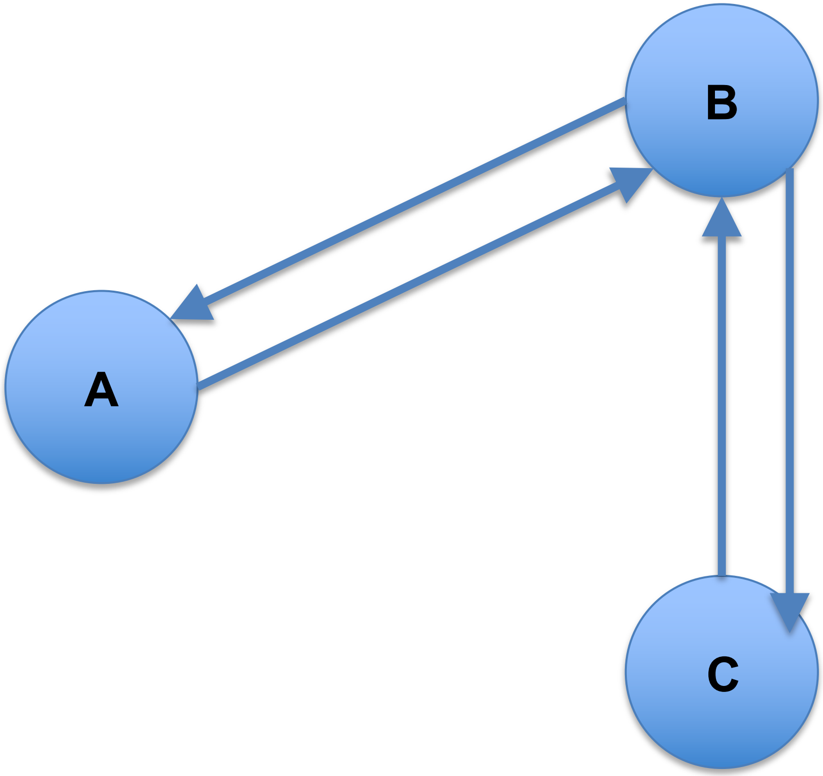
Many ways to compute

Conceptually simplest is *power iteration*

$x(t+1) = x(t).P$

$x(t=2)=x(t+1).P=x(t).P^2$

Example



Transition probability matrix

- Start with the N × N adjacency matrix A

Adjacency matrix

	A	B	C
A	0	1	0
B	1	0	1
C	0	1	0

Transition probability matrix

- Start with the N × N adjacency matrix A
- If row of A is all zeros, then add 1/N to each entry in the row

Adjacency matrix

	A	B	C
A	0	1	0
B	1	0	1
C	0	1	0

Transition probability matrix

- Start with the N × N adjacency matrix A
- If row of A is all zeros, then add 1/N to each entry in the row
- For all other rows:
  - Divide each 1 in A by number of 1's in row

Adjacency matrix

	A	B	C
A	0	1	0
B	0.5	0	0.5
C	0	1	0

Transition probability matrix

- Start with the N × N adjacency matrix A
- If row of A is all zeros, then add 1/N to each entry in the row
- For all other rows:
  - Divide each 1 in A by number of 1’s in row
- Multiply resulting matrix by 1-α, (α=0.5)

Adjacency matrix

	A	B	C
A	0	0.5	0
B	0.25	0	0.25
C	0	0.5	0

Transition probability matrix

- Start with the N × N adjacency matrix A
- If row of A is all zeros, then add 1/N to each entry in the row
- For all other rows:
  - Divide each 1 in A by number of 1’s in row
- Multiply resulting matrix by 1-α, then
- Add α/N to all elements of resulting matrix (0.5/3=0.17)

Transition probability matrix

	A	B	C
A	0.17	0.67	0.17
B	0.42	0.17	0.42
C	0.17	0.67	0.17

Power iteration

Start with x=(1 0 0)

Power iteration

$$\begin{array}{|c|c|c|}\hline 1 & 0 & 0 \\ \hline \end{array} \times \begin{array}{|c|c|c|}\hline 0.17 & 0.67 & 0.17 \\ \hline 0.42 & 0.17 & 0.42 \\ \hline 0.17 & 0.67 & 0.17 \\ \hline \end{array} = \begin{array}{|c|c|c|}\hline 0.17 & 0.67 & 0.17 \\ \hline \end{array}$$

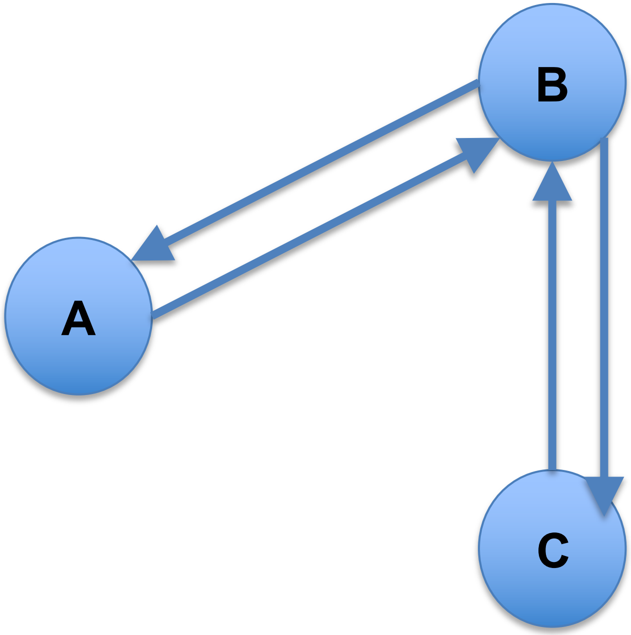
Power iteration

$$\begin{bmatrix} 0.17 & 0.67 & 0.17 \end{bmatrix} \times \begin{bmatrix} 0.17 & 0.67 & 0.17 \\ 0.42 & 0.17 & 0.42 \\ 0.17 & 0.67 & 0.17 \end{bmatrix} = \begin{bmatrix} 0.34 & 0.34 & 0.34 \end{bmatrix}$$

Power iteration

$$\begin{bmatrix} 0.34 & 0.34 & 0.34 \end{bmatrix} \times \begin{bmatrix} 0.17 & 0.67 & 0.17 \\ 0.42 & 0.17 & 0.42 \\ 0.17 & 0.67 & 0.17 \end{bmatrix} = \begin{bmatrix} 0.26 & 0.51 & 0.26 \end{bmatrix}$$

Power iteration



A	B	C
0.28	0.44	0.28

HITS: HUBS AND AUTHORITIES

Hyperlink-Induced Topic Search (HITS) Model

“Authoritative sources in a hyperlinked environment”, JM Kleinberg, Journal of the ACM (JACM), 1999

Two classes of documents

- Authorities are pages containing useful information
  - home pages of auto manufacturers
- Hubs are pages that link to authorities
  - list of US auto manufacturers

Definitions

- A good hub links to many good authorities
- A good authority has links from many good hubs
- Circular definition - will turn this into an iterative computation

- We now have two scores for each node
  - A Hub score and Authority score
  - Represented as vectors **h** and **a**

The problem being addressed

- *Broad-topic queries*. For example, “Find information about the Java programming language.”
- For *broad-topic queries*, we expect to find many thousands of relevant pages
- *Abundance Problem: The number of pages that could reasonably be returned as relevant is far too large for a human user to digest*

The problem being addressed

Query “Harvard”, but www.Harvard.edu is just one of millions of pages containing the word.

Content-dependent (text) search cannot fix this.

The problem being addressed

Query “search engines”, but Yahoo! and Google may not use these term on their websites.

Query “automobile manufacturers”, but Toyota may not use this term on its website

The problem being addressed

"Another issue is the difficulty in finding an appropriate balance between the criteria of *relevance* and *popularity*"

The problem being addressed

“Of all pages containing the query string, return those with the greatest number of in-links.”

“…, this heuristic would consider a universally popular page such as www.yahoo.com or www.netscape.com to be highly authoritative with respect to any query string that it contained.”

Solution: Base set

- Given text query (say ***browser***), use a text index to get all (top 100) pages containing ***browser.***
  - Call this the *root set* of pages.
- Add in any page that either
  - points to a page in the root set, or
  - is pointed to by a page in the root set.
- Call this the base set.

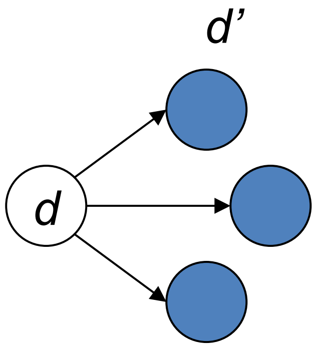
Hubs and Authorities

- Compute, for each page *d* in the base set, a hub score *h(d)* and an authority score *a(d).*
- Initialize: for all *d, h(d)=1; a(d) =1*;
- Iteratively update all *h(), a()*;
- After iterations
  - output pages with highest *h()* scores as top hubs
  - highest *a()* scores as top authorities.

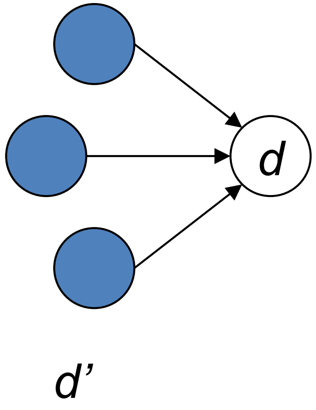
Iterative update

Repeat the following updates, for all *d*

$$h(d) \leftarrow \sum_{\forall d' : d \mapsto d'} a(d')$$



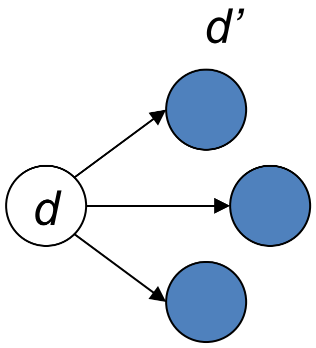
$$a(d) \leftarrow \sum_{\forall d' : d' \mapsto d} h(d')$$



Iterative update

Repeat the following updates, for all *d*

$$h(d) \leftarrow \sum_{\forall d' : d \mapsto d'} a(d')$$



Hubs point to authorities, so the hub score is the sum of the authority of each page pointed to

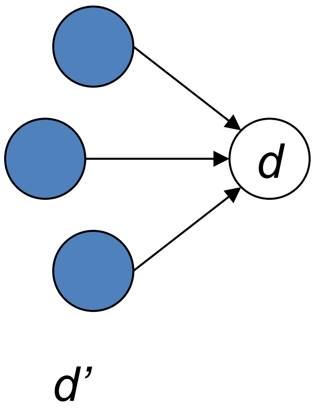
a´

Iterative update

Repeat the following updates, for all *d*

Authorities are pointed to by hubs, so the authority score is the sum of the hub score of each page pointing to it

$$a(d) \leftarrow \sum_{\forall d' : d' \mapsto d} h(d')$$



Theory

From

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

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

We can obtain

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

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

where A is the adjacency matrix

Theory

- 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
- Similar to PageRank, the algorithm is a particular known algorithm for computing eigenvectors: the *power iteration* method.

PageRank and HITS

PageRank and HITS are two solutions to the same problem

The destinies of PageRank and HITS post-1998 were very different

PageRank, HITS, and in-degree

See “HITS on the Web: How does it Compare?”, M. Najork, Hugo Zaragoza and Michael Taylor, SIGIR 2007