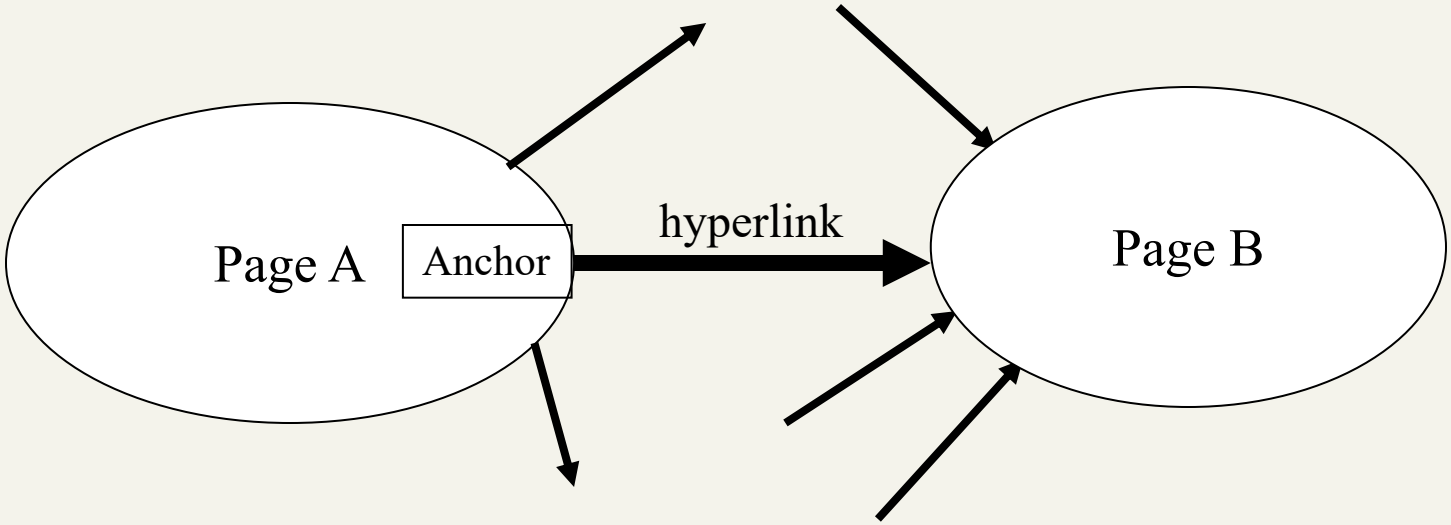
Introduction to Information Retrieval
(Manning, Raghavan, Schutze)

Chapter 21
Link analysis

Content

- Anchor text
- Link analysis for ranking
  - Pagerank and variants
  - HITS

The Web as a Directed Graph



Assumption 1: a hyperlink is a quality signal

- A hyperlink between pages denotes author perceived relevance

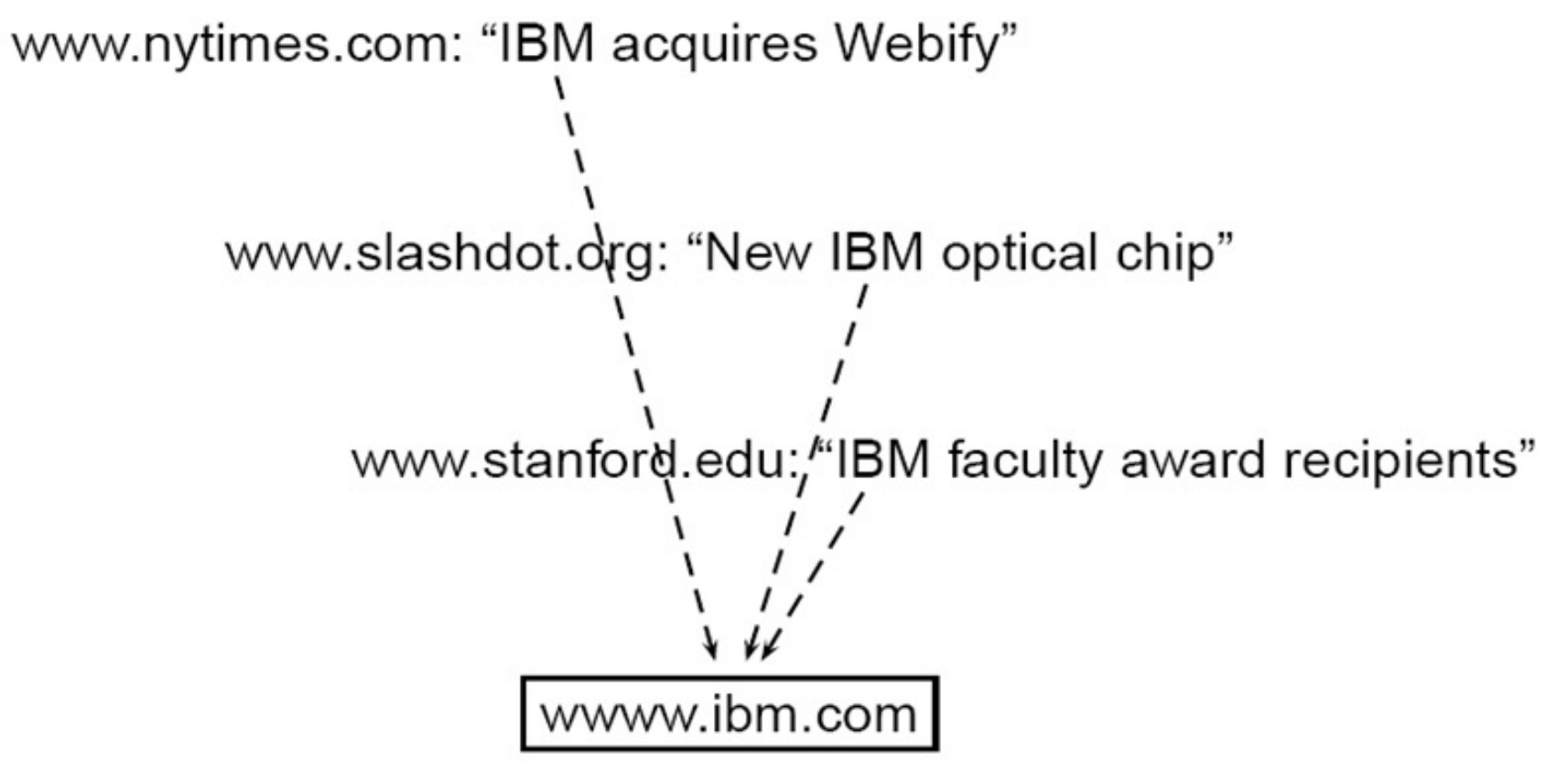
Assumption 2: The anchor text describes the target page

- we use anchor text somewhat loosely here
- extended anchor text, window of text surrounding anchor text
- You can find cheap cars here

[document text only] vs. [document text + anchor text]

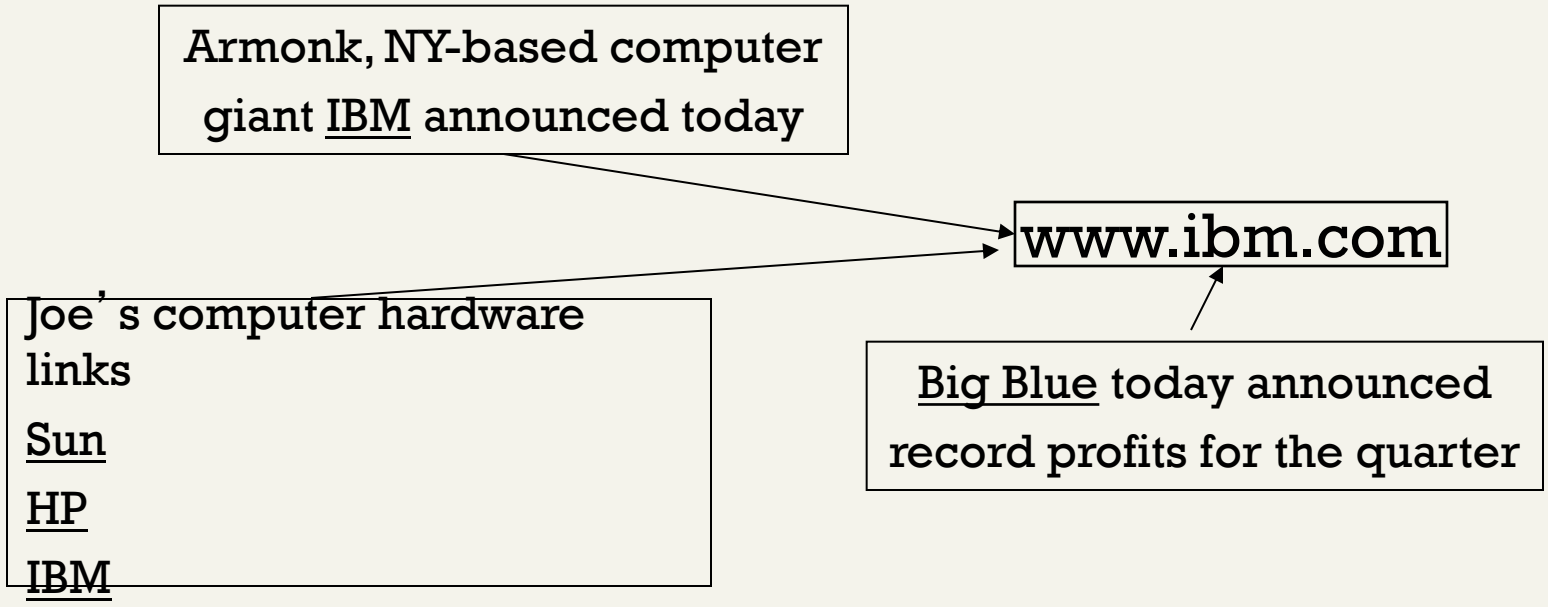
- Searching on [document text + anchor text] is often more effective than searching on [document text only].
- Example: Query *IBM*
  - Matches IBM's copyright page
  - Matches many spam pages
  - Matches IBM wikipedia article
  - May not match IBM home page! (if IBM home page is mostly graphical)
- Searching on anchor text is better for the query *IBM*.
- Represent each page by all the anchor text pointing to it.
- In this representation, the page with the most occurrences of *IBM* is www.ibm.com.

Anchor text containing IBM pointing to www.ibm.com



Indexing anchor text

- Thus: anchor text is often a better description of a page's content than the page itself
- Anchor text can be weighted more highly than document text (based on Assumptions 1 & 2)
- When indexing a document *D*, include anchor text from links pointing to *D*.



Google bombs

- Indexing anchor text can have unexpected side effects: Google bombs.
  - whatelse does not have side effects?
- A Google bomb is a search with "bad" results due to maliciously manipulated anchor text
- Google introduced a new weighting function in January 2007 that fixed many Google bombs

Google bomb example

Origins of PageRank: Citation analysis (1)

- Citation analysis: analysis of citations in the scientific literature
- Example citation: "Miller (2001) has shown that physical activity alters the metabolism of estrogens."
- "Miller (2001)" is a hyperlink linking two scientific articles.
- One application of these "hyperlinks" in the scientific literature:
  - Measure the similarity of two articles by the overlap of other articles citing them.
  - This is called cocitation similarity.
- Cocitation similarity on the web?

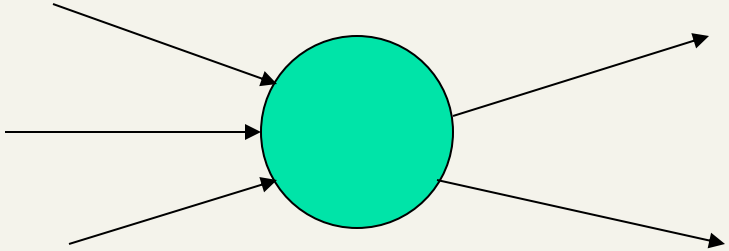
Cocitation similarity on Google: similar pages

Origins of PageRank: Citation analysis (2)

- Citation frequency can be used to measure the impact of an article.
  - Each article gets one vote.
  - Not a very accurate measure
- Better measure: weighted citation frequency / citation rank
  - An article's vote is weighted according to its citation impact.
  - Circular? No: can be formalized in a well-defined way.
  - This is basically PageRank.
  - PageRank was invented in the context of citation analysis by Pinsker and Narin in the 1960s.

Query-independent ordering

- First generation link-based ranking for web search
  - using link counts as simple measures of popularity.
  - simple link popularity: number of in-links

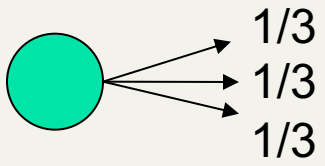


  - First, retrieve all pages meeting the text query (say ***venture capital***).
  - Then, order these by the simple link popularity

- Easy to spam. Why?

Basics for PageRank: random walk

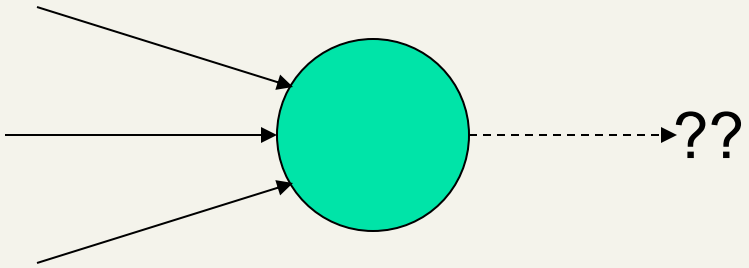
- Imagine a web surfer doing a random walk on the web page:
  - start at a random page
  - at each step, go out of the current page along one of the links on that page, equiprobably



- In the steady state each page has a long-term visit rate - use this as the page's score
- So, pagerank = steady state probability
  = long-term visit rate

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

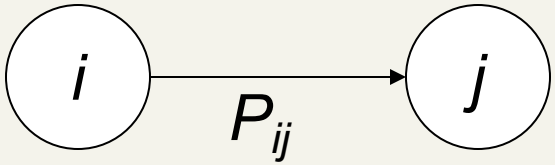


Teleporting

- Teleport operation: surfer jumps from a node to any other node in the web graph, chosen uniformly at random from all web pages
- Used in two ways:
  - At a dead end, jump to a random web page
  - At any non-dead end, with **teleportation probability** $0 < \alpha < 1$ (say, $\alpha = 0.1$), jump to a random web page; with remaining probability $1 - \alpha$ (0.9), go out on a random link
- Now cannot get stuck locally
- There is a long-term rate at which any page is visited
  - Not obvious, explain later
  - How do we compute this visit rate?

Markov chains

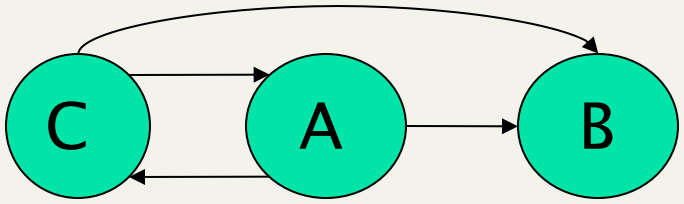
- A Markov chain consists of n states, plus an $n \times n$ transition probability matrix **P**.
- At each step, we are in exactly one of the states.
- For $1 \leq i, j \leq n$, the matrix entry P_{ij} tells us the probability of j being the next state, given we are currently in state i.



- Clearly, for each i, $\sum_{j=1}^{n} P_{ij} = 1.$
- Markov chains are abstractions of random walk
  - State = page

Exercise

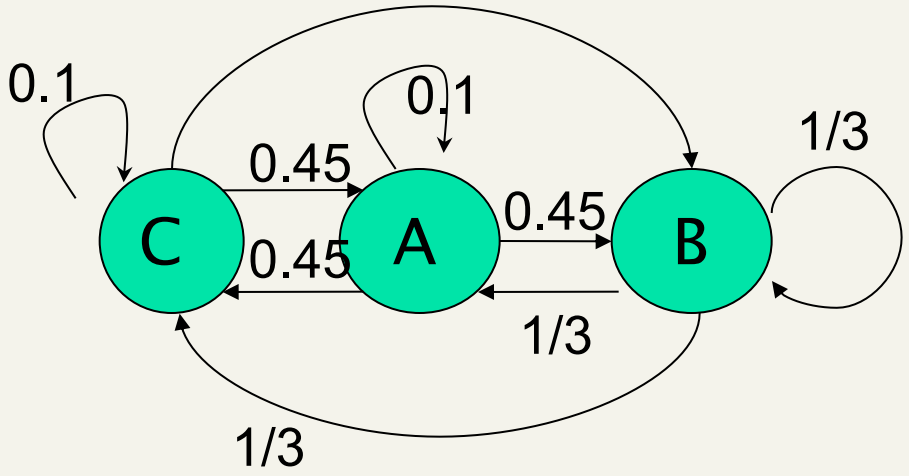
Represent the teleporting random walk as a Markov chain, for the following case, using transition probability matrix



Link structure

$\alpha = 0.3$:

State diagram

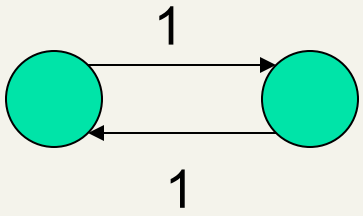


Transition matrix

0.1	0.45	0.45
1/3	1/3	1/3
0.45	0.45	0.1

Ergodic Markov chains

- A Markov chain is ergodic iff it's irreducible and aperiodic
  - Irreducibility: roughly, there is a path from any state to any other
  - Aperiodicity: roughly, the nodes cannot be partitioned such that the random walker visits the partitions sequentially
- A non-ergodic Markov chain



Ergodic Markov chains

- Theorem: For any ergodic Markov chain, there is a unique long-term visit rate for each state.
  - *Steady-state probability distribution*.
- Over a long time-period, we visit each state in proportion to this rate.
- It doesn't matter where we start.

Formalization of visit: probability vector

- A probability (row) vector $\mathbf{x} = (x_1, \ldots x_n)$ tells us where the walker is at any point.
- e.g., (000…1…000) means we're in state i
  (1 under the first 0, i under the 1, n under the last 0)

More generally, the vector $\mathbf{x} = (x_1, \ldots x_n)$ 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 $\mathbf{x} = (x_1, \ldots x_n)$ 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 is simply a vector of probabilities $\mathbf{a} = (a_1, \ldots a_n)$:
  - a_i is the probability that we are in state i
  - a_i is the long-term visit rate (or pagerank) of state (page) I
  - so we can think of pagerank as a long vector, one entry for each page

How do we compute this vector?

- Let **a** = $(a_1, \ldots a_n)$ denote the row vector of steady-state probabilities.
- 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**
- Solving this matrix equation gives us **a**
  - so **a** is the (left) eigenvector for **P**
  - corresponds to the principal eigenvector of **P** with the largest eigenvalue
  - transition probability matrices always have larges eigenvalue 1

One way of computing

- 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, $\mathbf{xP}^k = \mathbf{a}$
- 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: $\vec{x} = (0.5, 0.5)$, $P = \begin{pmatrix} 0.25 & 0.75 \\ 0.25 & 0.75 \end{pmatrix}$
- $\vec{x}P = (0.25, 0.75)$
- $\vec{x}P^2 = (0.25, 0.75)$
- Convergence in one iteration!

Pagerank summary

- Preprocessing:
  - Given graph of links, build transition probability 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*
- **In practice, pagerank alone wouldn't work**
- Google paper: http://infolab.stanford.edu/~backrub/google.html

In practice

- Consider the query “video service”
  - Yahoo! has very high pagerank, and contains both words
  - With simple pagerank alone, Yahoo! Would be top-ranked
  - Clearly not desirable

- In practice, composite score is used in ranking
  - Pagerank, cosine similarity, term proximity etc.
  - May apply machine-learned scoring
  - Many other clever heuristics are used

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, phrases, proximity, tiered indexes . . .
  - Rumor has it that PageRank in its original form (as presented here) has a negligible impact on ranking!
  - However, variants of a page's PageRank are still an essential part of ranking.
  - Adressing link spam is difficult and crucial.

Pagerank: Issues and Variants

- How realistic is the random surfer model?
  - What if we modeled the back button?
  - Surfer behavior sharply skewed towards short paths
  - Search engines, bookmarks & directories make jumps non-random.
- Biased Surfer Models
  - Weight edge traversal probabilities based on match with topic/query (non-uniform edge selection)
  - Bias jumps to pages on topic (e.g., based on personal bookmarks & categories of interest)
- Non-uniform teleportation allows topic-specific pagerank and personalized pagerank

Topic Specific Pagerank

- Conceptually, we use a random surfer who teleports, with say 10% probability, using the following rule:
  - Selects a category (say, one of the 16 top level ODP categories) based on a query & user -specific distribution over the categories
  - Teleport to a page uniformly at random within the chosen category

Pagerank applications beyond web search

- A person is reputable if s/he receives many references from reputable people.
- How to compute reputation for people?

- Rent a room in an exhibition center. Find one with the most visit rate.

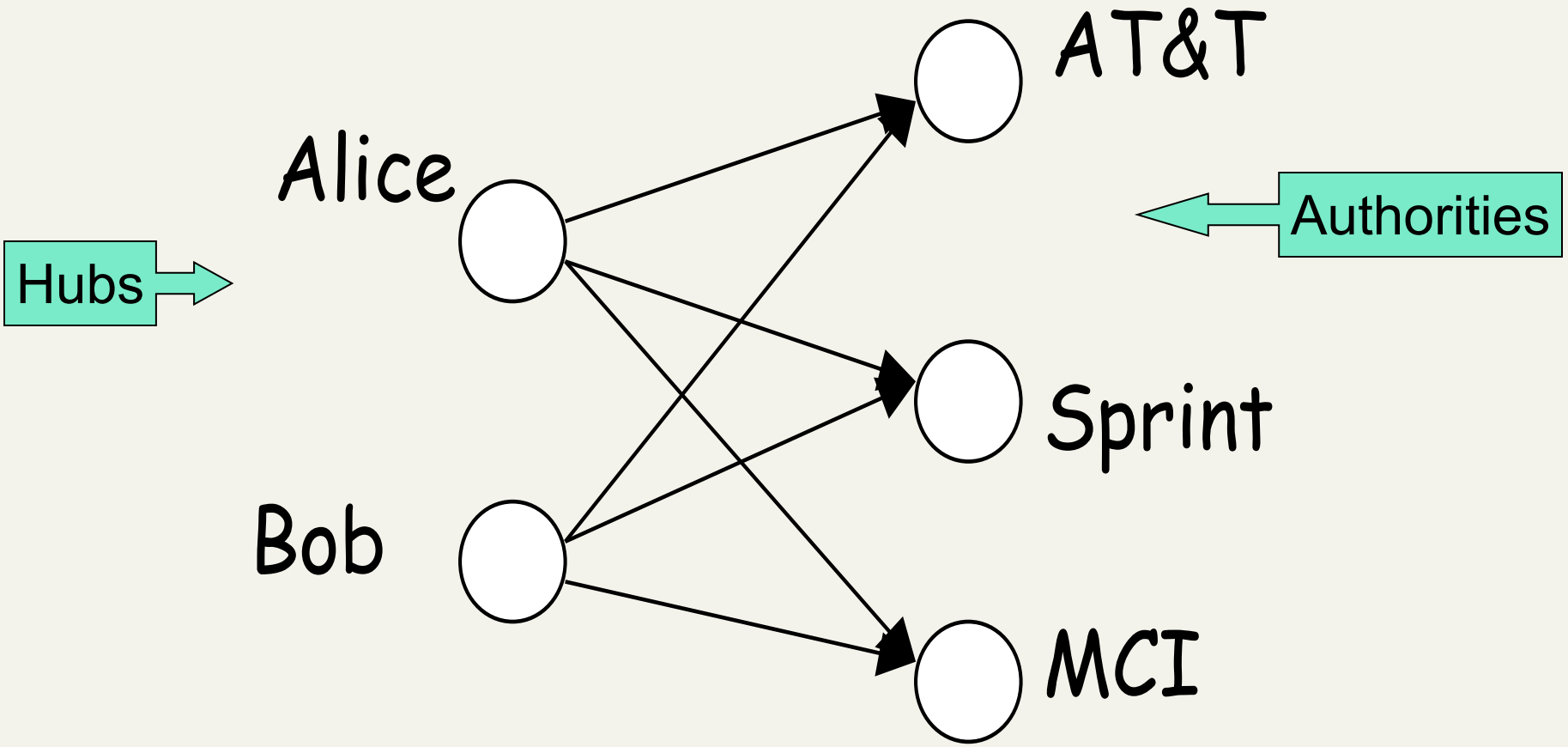
Hyperlink-Induced Topic Search (HITS)

- In response to a query, instead of an ordered list of pages each meeting the query, find two sets of inter-related pages:
  - *Hub pages* are good lists of links to pages answering the information need
    - e.g., "Bob's list of cancer-related links
  - *Authority pages* are direct answers to the information need
    - occur recurrently on good hubs for the subject
- Most approaches to search do not make the distinction between the two sets

Hubs and Authorities

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

Examples of hubs and authorities

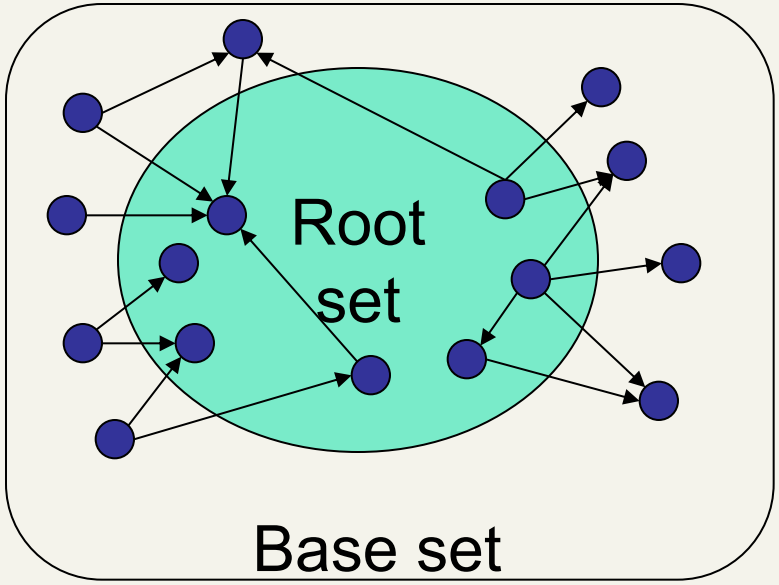


Long distance telephone companies

High-level scheme

- Do a regular web search first
- Call the search results the root set
- 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
- From these, identify a small set of top hub and authority pages
- Iterative algorithm

Visualization

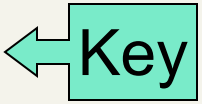


Assembling the base set

- Root set typically 200-1000 nodes
- Base set may have up to 5000 nodes
- How do you find the base set nodes?
  - Follow out-links by parsing root set pages
  - Get in-links from a *connectivity server*, get pages
    - This assumes our inverted index supports searches for links, in addition to terms

Distilling hubs and authorities

- Compute, for each page *x* in the base set, a hub score *h(x)* and an authority score *a(x)*
- Initialize: 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
  - output pages with highest *a()* scores as top authorities
  - so we output two ranked lists

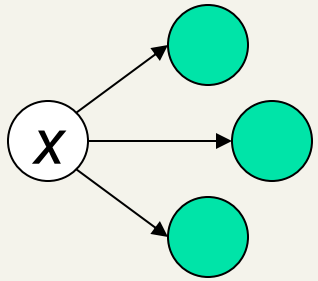


Iterative update

- Iterate these two steps until convergence

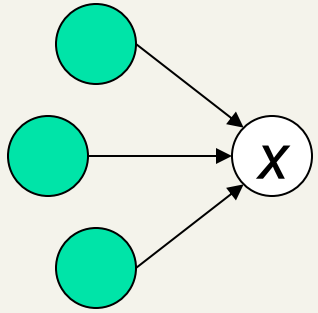
for all x:

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



for all x:

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



Scaling

- To prevent the *h()* and *a()* 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?

- Relative values of scores will converge after a few iterations
- In fact, suitably scaled, *h()* and *a()* scores settle into a steady state!
  - proof of this comes later

- In practice, ~5 iterations get you close to stability

Japan Elementary Schools

Hubs

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

Authorities

- The American School in Japan
- The Link Page
- ‰ª□è□s— § ˆä“c□¬Šw□Zƒz□[ƒ□ƒy□[ƒW
- Kids' Space
- ˆÀ□é□s— § ˆÀ□é□¼•”□¬Šw□Z
- ‹{□é‹³ˆç ‘åŠw•□ ‘®□¬Šw□Z
- KEIMEI GAKUEN Home Page (Japanese)
- Shiranuma Home Page
- fuzoku-es.fukui-u.ac.jp
- welcome to Miasa E&J school
- □_“Þ□ìŒ § □E‰¡•I□s— § ’ †□ì□¼□¬Šw□Z‚Ìƒy
- 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
- Is HITS query-independent?
  - Typical use, no
- Iterative computation after text index retrieval - significant overhead.

PageRank vs. HITS: Discussion

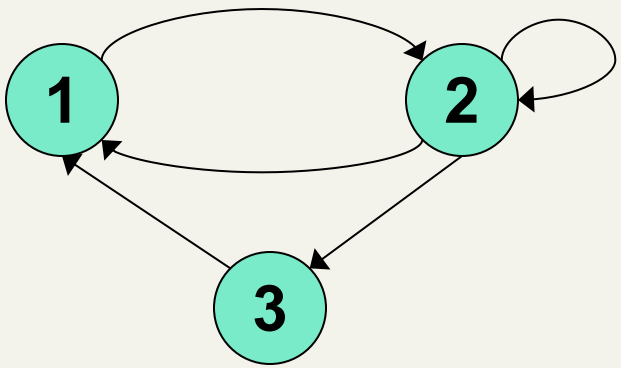
- The PageRank and HITS make two different design choices concerning (i) the eigenproblem formalization (ii) the set of pages to apply the formalization to
- These two are orthogonal
  - We could also apply HITS to the entire web and PageRank to a small base set
- On the web, a good hub almost always is also a good authority
- The actual difference between PageRank ranking and HITS ranking is therefore not as large as one might expect

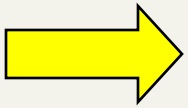
HITS applications beyond web search

- Researchers publish/present papers in conferences. A conference is reputable if it hosts many reputable researchers to publish/present their papers. A researcher is reputable if s/he publishes/presents many papers in reputable conferences.
- How to compute reputation for conferences? How to compute reputation for researchers?

Proof of convergence

- $n \times n$ adjacency matrix **A**:
  - each of the n pages in the base set has a row and column in the matrix.
  - Entry $A_{ij} = 1$ if page i links to page j, else = 0.





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

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

- **h**=**Aa**.
- **a**=**A**^{t}**h**.

Recall A^t is the transpose of A.

Substituting, **h**=**AA**^{t}**h** and **a**=**A**^{t}**Aa**.

Thus, **h** is an eigenvector of **AA**^{t} and **a** is an eigenvector of **A**^{t}**A**.

Further, our algorithm is a particular, known algorithm for computing eigenvectors: the *power iteration* method.

Guaranteed to converge.