PageRank and Similar Ideas

Topic-Sensitive PageRank

Spam: TrustRank, Spam Mass

SimRank

HITS (Hubs and Authorities)

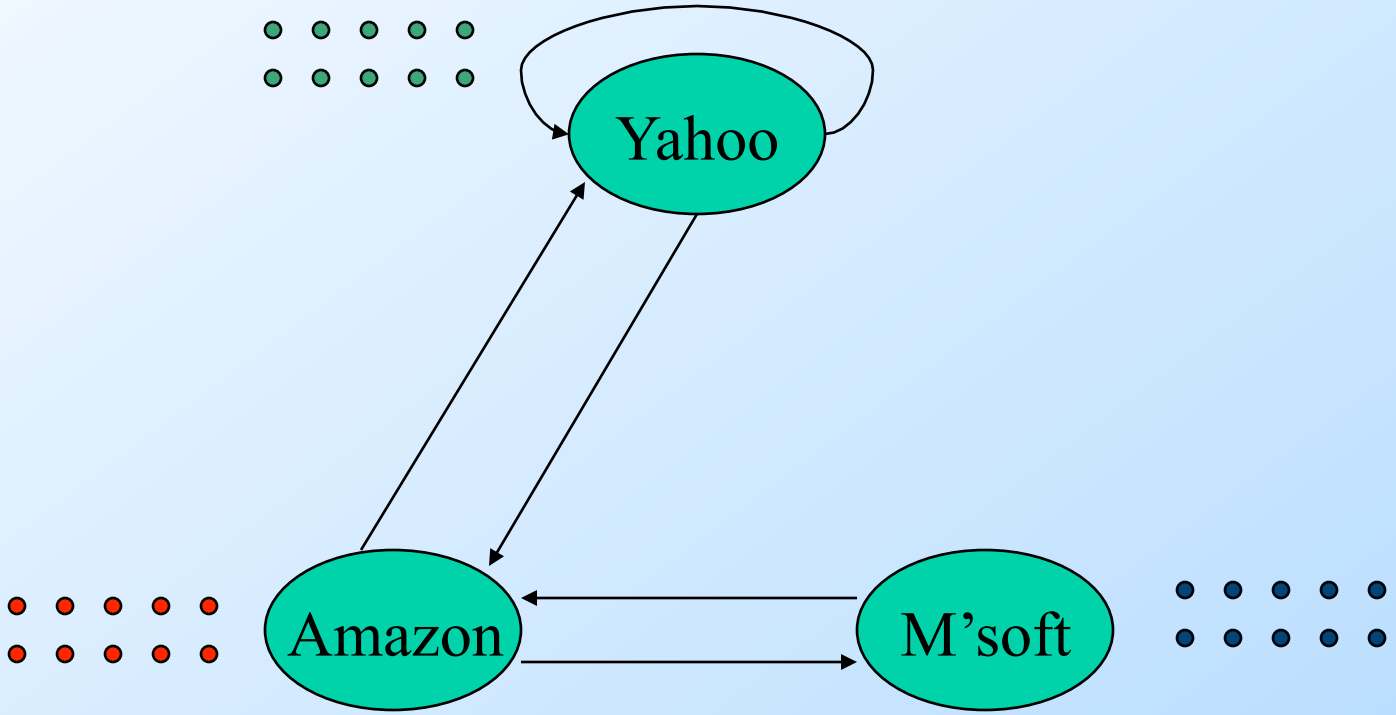
Topic-Sensitive PageRank

Random Walkers

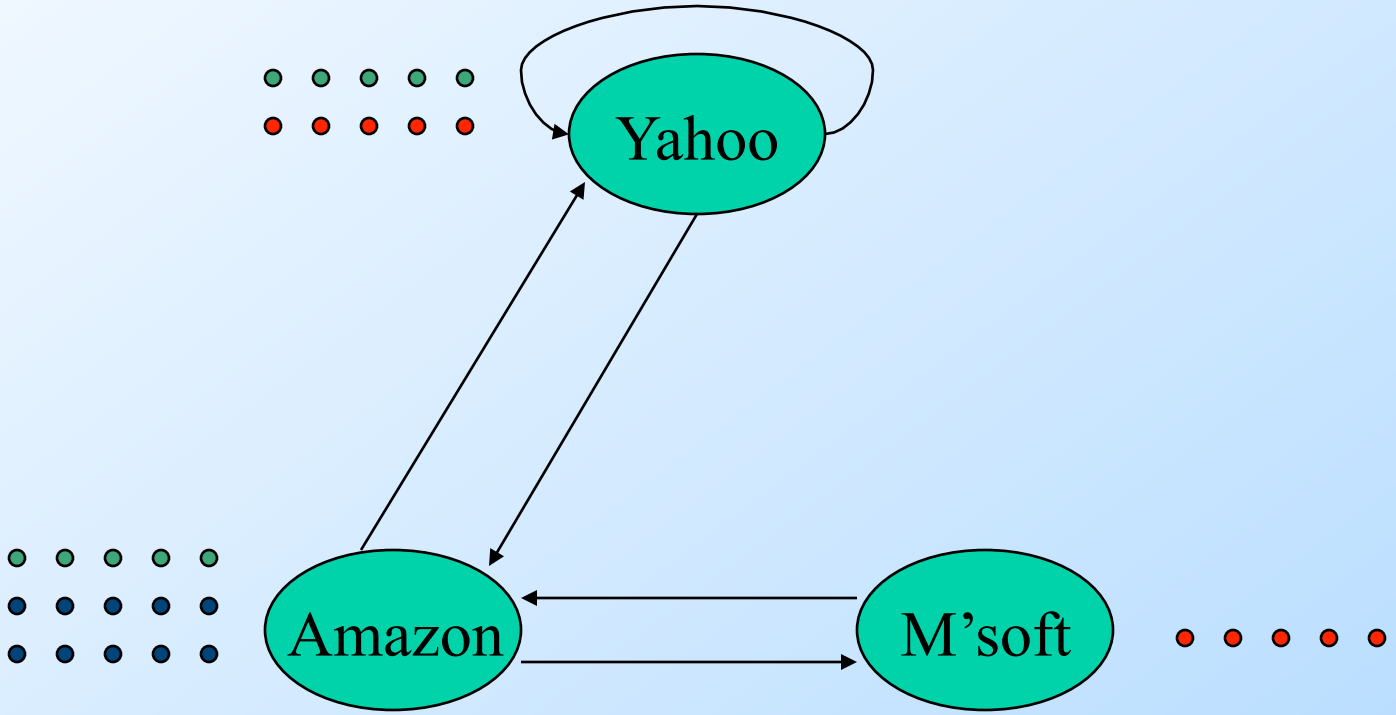
Teleport Sets

Deducing Relevant Topics

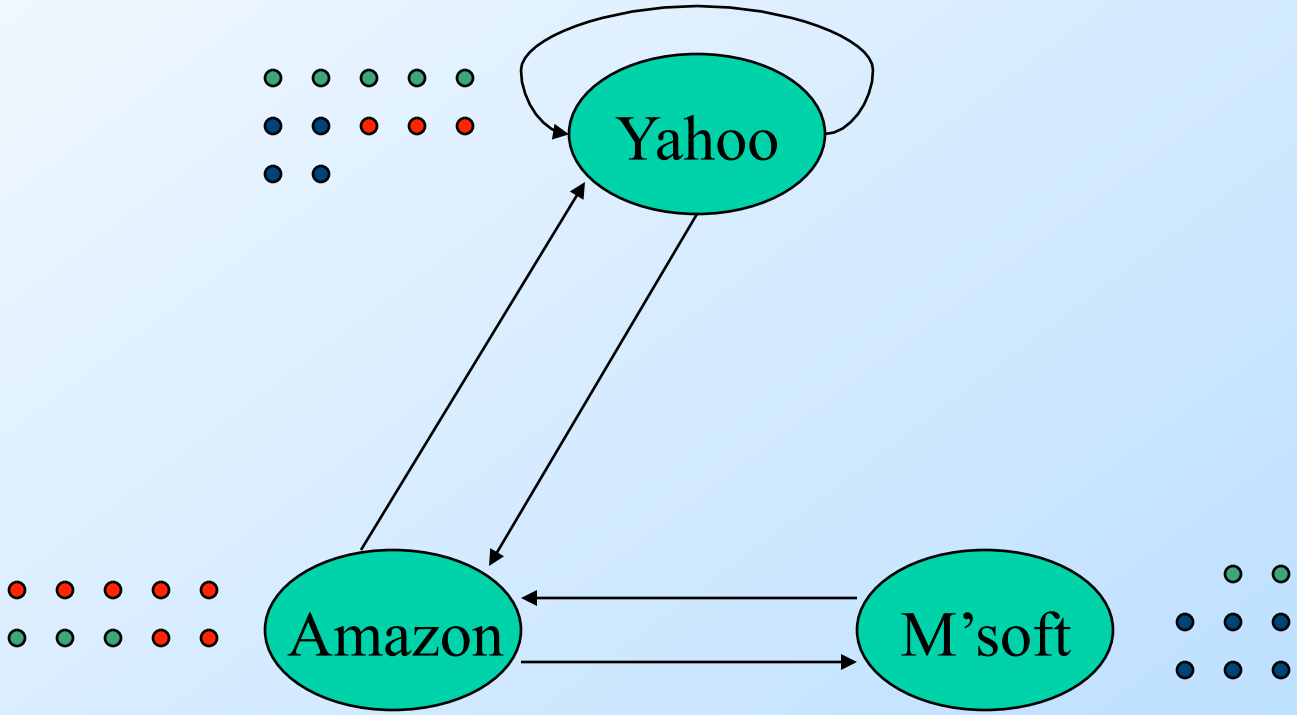
The Walkers



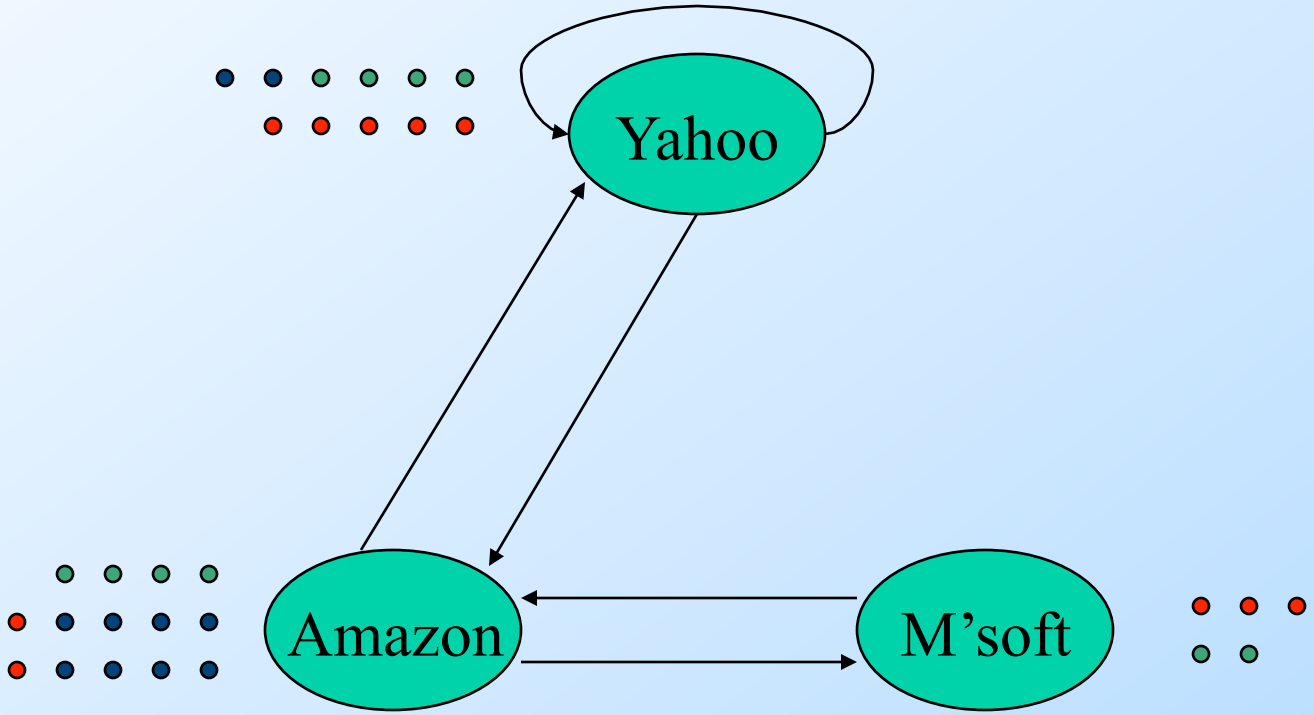
The Walkers



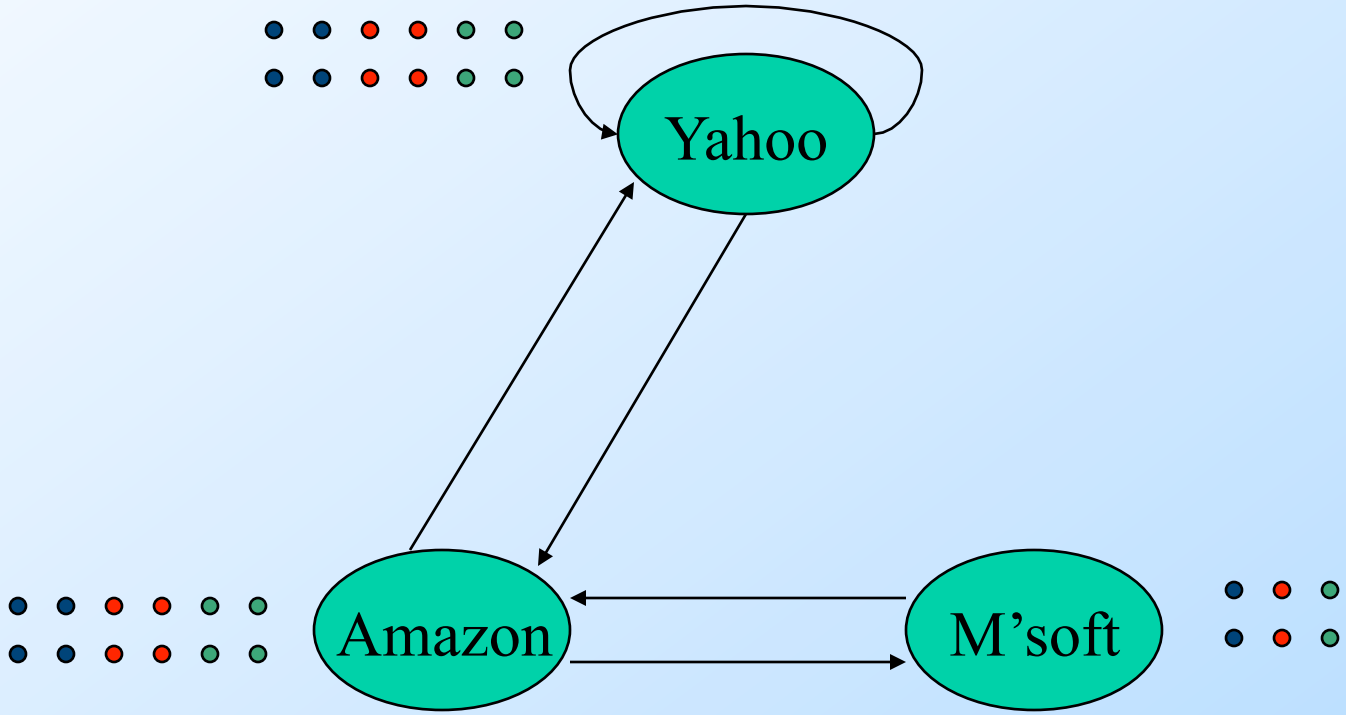
The Walkers



The Walkers



In the Limit …



Topic-Specific Page Rank

- Goal: Evaluate Web pages not just according to their popularity, but by how close they are to a particular topic, e.g. "sports" or "history."
- Allows search queries to be answered based on interests of the user.
  - Example: Query `Trojan` wants different pages depending on whether you are interested in sports or history.

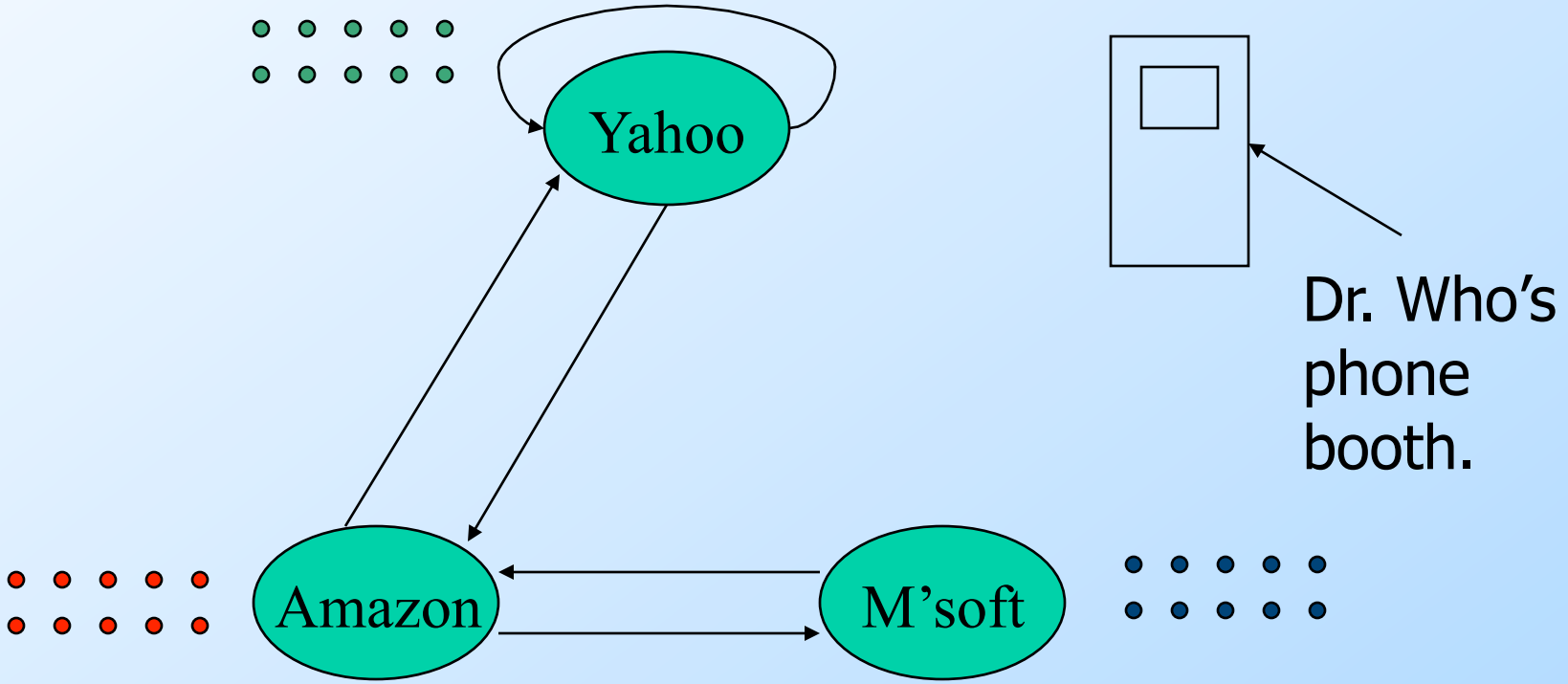
Teleport Sets

- Assume each walker has a small probability of "teleporting" at any tick.
- Teleport can go to:
  1. Any page with equal probability.
     - To avoid dead-end and spider-trap problems.
  2. A topic-specific set of "relevant" pages (*teleport set*).
     - For *topic-sensitive* PageRank.

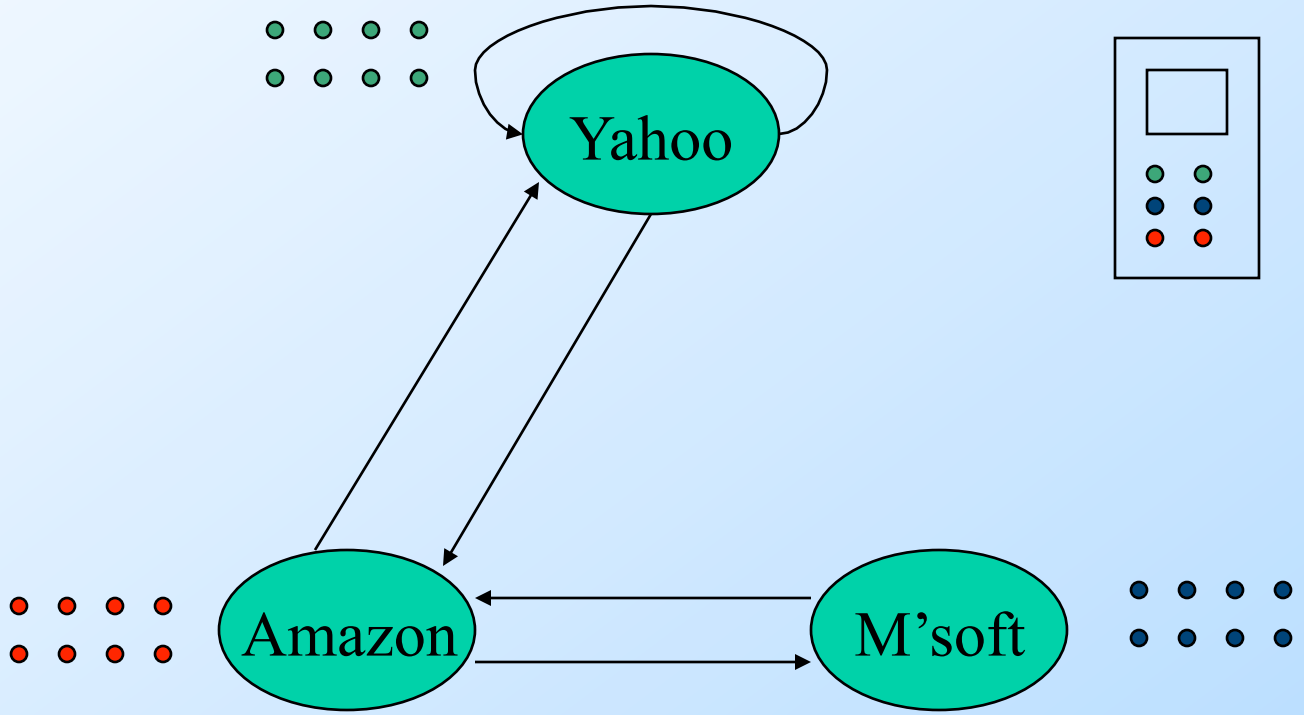
Example: Topic = Software

- Only Microsoft is in the teleport set.
- Assume 20% "tax."

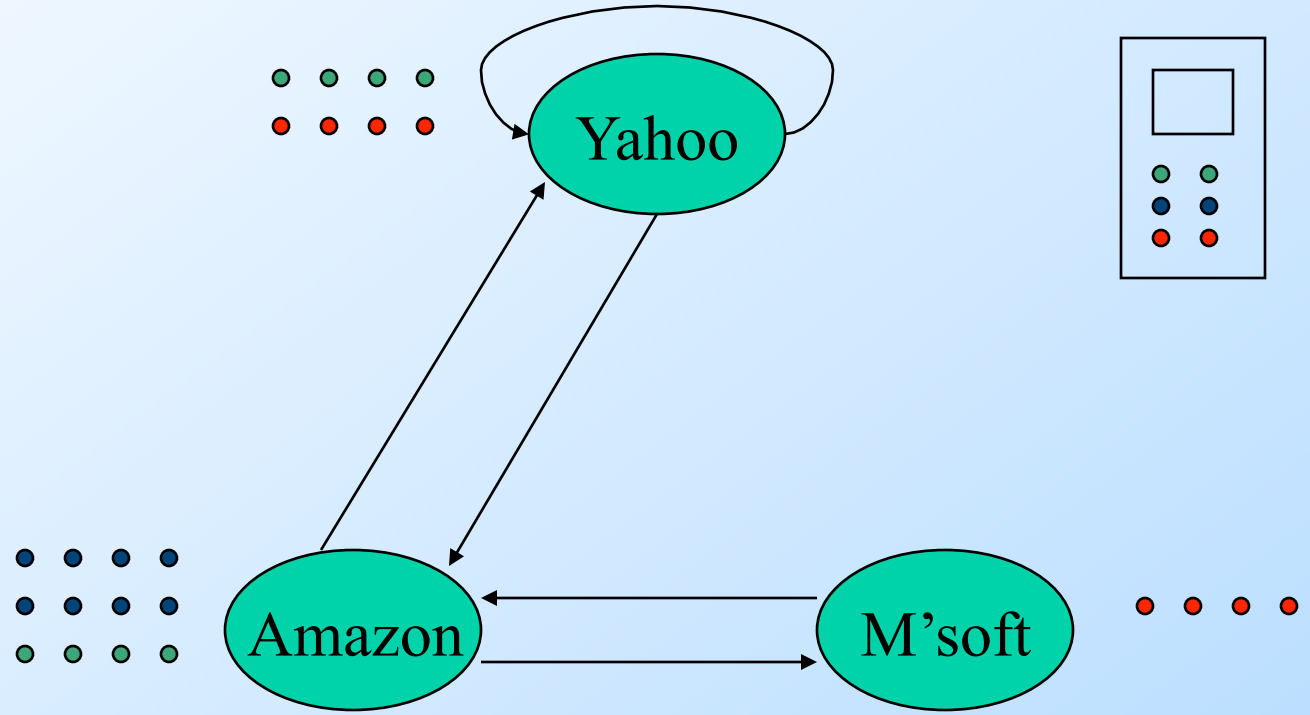
Only Microsoft in Teleport Set



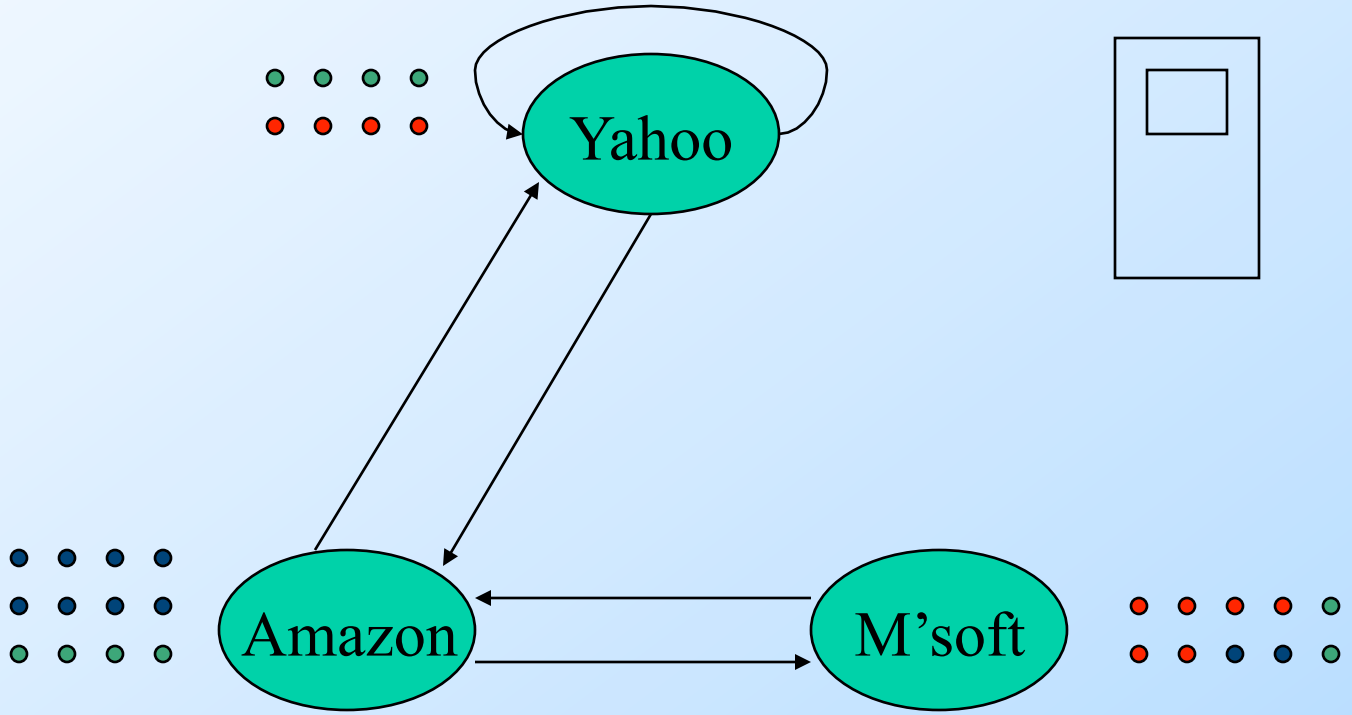
Only Microsoft in Teleport Set



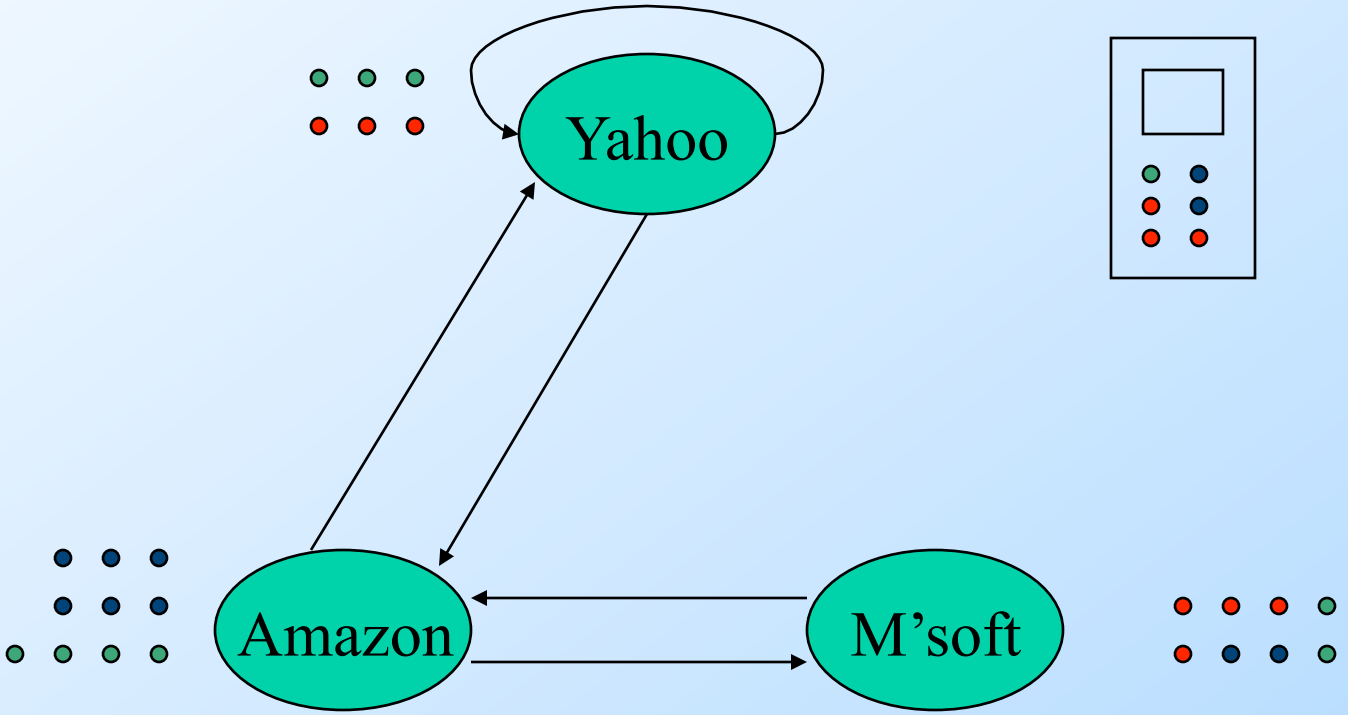
Only Microsoft in Teleport Set



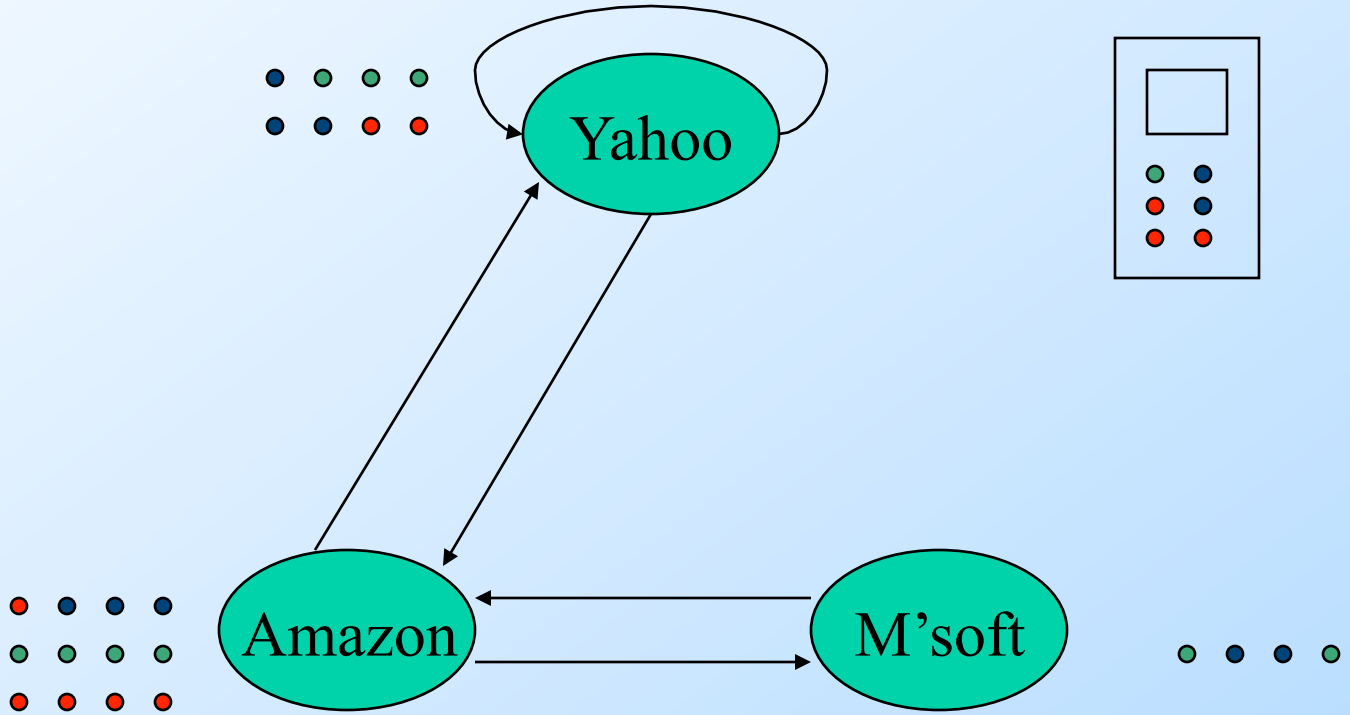
Only Microsoft in Teleport Set



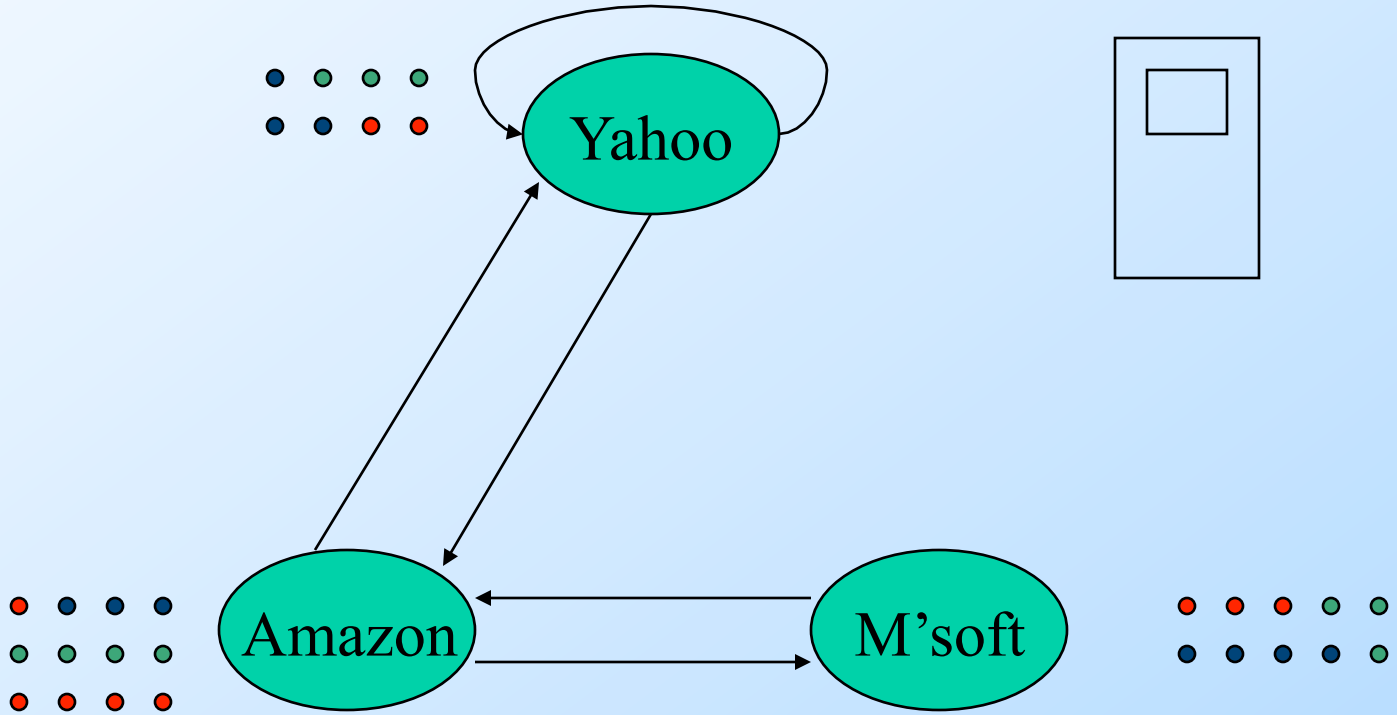
Only Microsoft in Teleport Set



Only Microsoft in Teleport Set



Only Microsoft in Teleport Set



Matrix Formulation

- A_{ij} =
  - $\beta M_{ij} + (1-\beta)/|S|$ if i is in S
  - βM_{ij} otherwise
- Compute as for regular PageRank:
  - Multiply by M, then add a vector.
  - Maintains sparseness.

Discovering the Topic

- Create different PageRanks for different topics.
  - E.g., the 16 DMOZ top-level nodes.
- Several ways to guess what topic the queryer is interested in.
  - Words in previous pages viewed.
  - Bookmarked pages.
  - Expressed preferences.

Link Spam

History of Spam

Spam Farms

TrustRank

Spam Mass

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 = pages created for spamming
- SEO industry might disagree!
  - SEO = search engine optimization
- Approximately 10-15% of web pages are spam

Early Search Engines

1. *Crawl the Web* (follow links from page to page, finding and copying as many pages as they could).
2. Index pages by the words they contained.
3. Respond to *search queries* (lists of words) with the pages containing those words.

Early Page Ranking

- Attempt to order pages matching a search query by "importance."
- First search engines considered:
  1. Number of times query words appeared.
  2. Prominence of word position, e.g. title, header.

The First Spammers

- As people began to use search engines to find things on the Web, those with commercial interests tried to exploit search engines to bring people to their own site – whether they wanted to be there or not.
- Example: shirt-seller might pretend to be about "movies."

The First Spammers – (2)

- How do you make your page appear to be about movies?
- Add the word `movie` 1000 times to your page.
- Set its color to the background color, so only search engines would see it.

The First Spammers – (3)

- Or, run the query `movie` on your target search engine.
- See what page came first in the listings.
- Copy it into your page, invisibly.
- These and similar techniques are *term spam*.

The First Spammers – (4)

- Rapidly, the promise of search engines disappeared.
- Spam dominated the listings to the extent that responses to search queries were useless.

The Google Solution to Term Spam

1. Believe what people say about you, rather than what you say about yourself.
   - Consider words in the *anchor text* (words that appear underlined to represent the link) and its surrounding text.
2. PageRank as a tool to measure the "importance" of Web pages.

Why Google Works

- Our hypothetical shirt-seller loses.
- His page isn't very important, so it won't be ranked high for shirts or movies.
- Saying he is about movies doesn't help, because others don't say he is about movies.

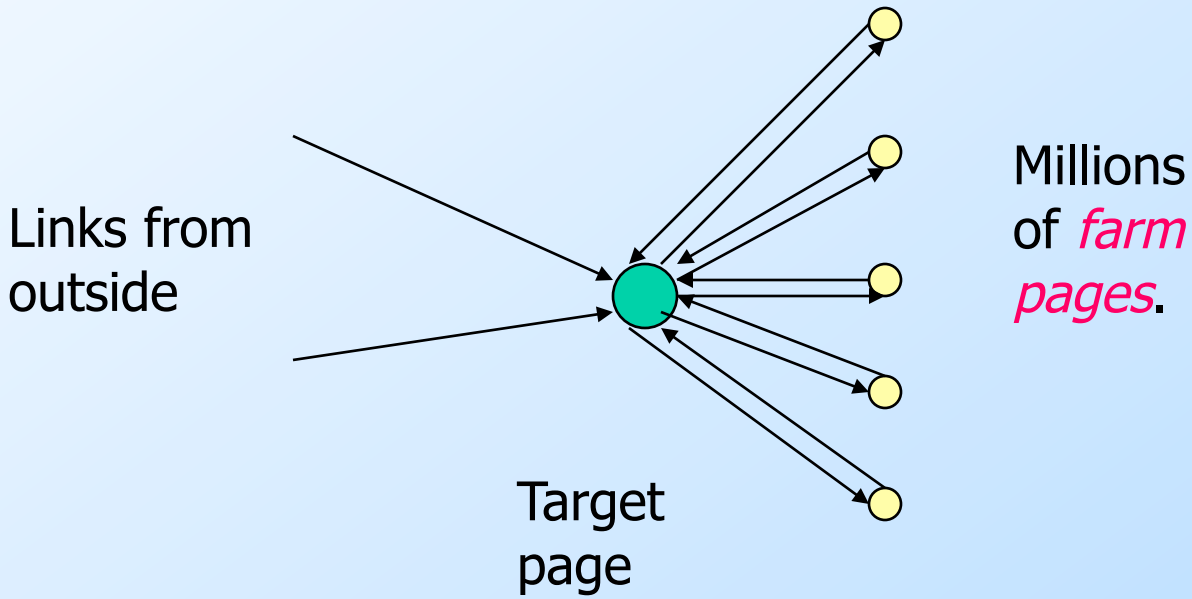
Simple Spam Techniques Fail

- Example: shirt-seller creates 1000 pages, each of which links to his with `movie` in the anchor text.
- These pages have no links in, so they get little PageRank.
- So the shirt-seller can't beat truly important movie pages like IMDB.

Round 2: *Link Spam*

- Once Google became the dominant search engine, spammers began to work out ways to fool Google.
- *Spam farms* were developed to concentrate PageRank on a single page.

Structure of a Typical Spam Farm



Farm Pages

- Even with taxation, farm pages can preserve most of the PageRank that the farm starts with.
- And it amplifies externally supplied PageRank by a significant factor.

External Links

- Where do external links come from?
- Blog pages allow spammers to add comments, e.g., "I agree. See www.mySpamFarm.com."

Combating Link Spam

1. Detection and blacklisting of structures that look like spam farms.
   - Leads to another war – hiding and detecting spam farms.
2. *TrustRank* = topic-specific PageRank with a teleport set of "trusted" pages.
   - Example: .edu domain, plus similar domains for non-US schools.

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
- Dumping
  - of a large number of unrelated terms
  - e.g., copy entire dictionaries
- Weaving
  - Copy legitimate pages and insert spam terms at random positions (to hide the spamming)
- Phrase Stitching
  - Glue together sentences and phrases from different sources (also hides spamming)

Detecting Term Spam

- Analyze text using statistical methods e.g., Naïve Bayes classifiers
  - Similar to email spam filtering
- Also useful: detecting approximate duplicate pages

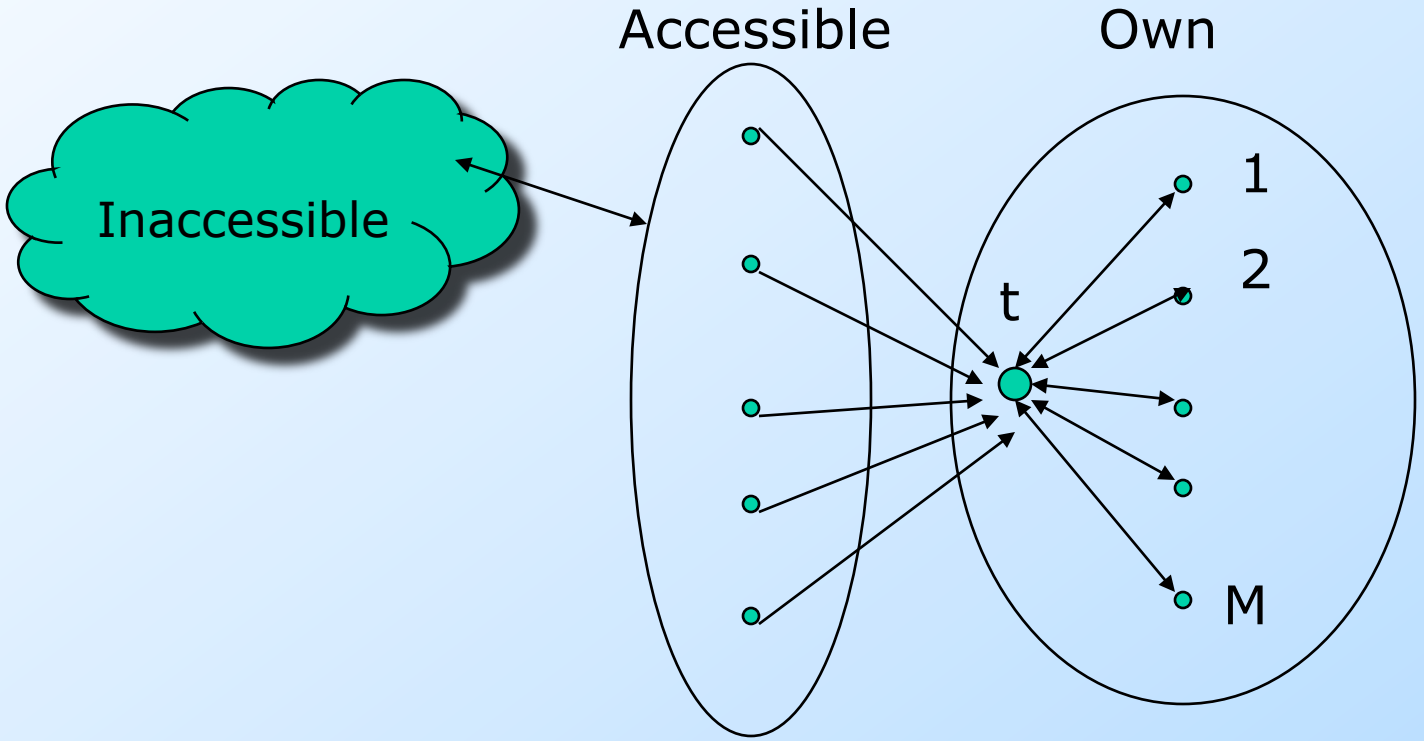
Link Spam

- 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

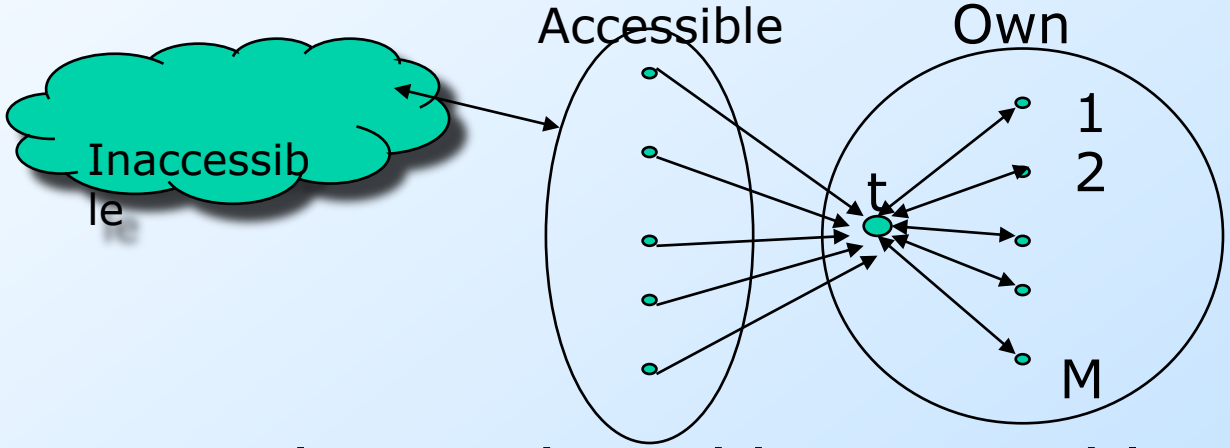
- Spammer's goal
  - Maximize the page rank of target page t
- Technique
  - Get as many links from accessible pages as possible to target page t
  - Construct "link farm" to get page rank multiplier effect

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

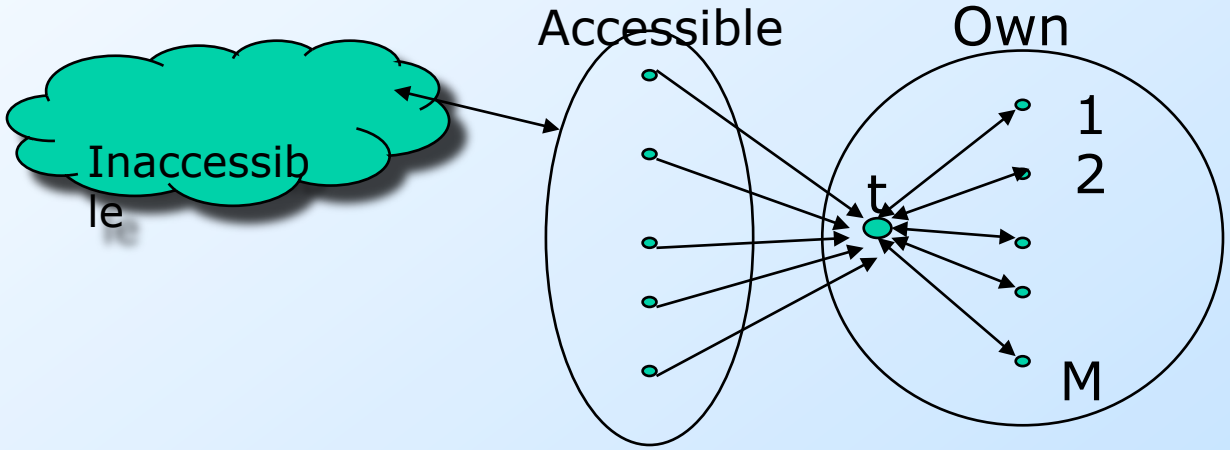
Rank of each "farm" page = $\beta y/M + (1-\beta)/N$

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

$= x + \beta^2 y + \beta(1-\beta)M/N + \boxed{(1-\beta)/N}$ 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

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"
- Perform a topic-sensitive PageRank with teleport set = trusted pages.
- Use a threshold value and mark all pages below the trust threshold as spam

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 TrustRank, so need make all good pages reachable from seed set by short paths

Approaches to Picking Seed Set

1. Pick top pages by PageRank.
   - Theory is that you can't get a bad page's rank really high.
2. Use domains whose membership is controlled, like .edu, .mil, .gov

Spam Mass

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

Spam Mass Estimation

$r(p)$ = PageRank of page p

$r^{+}(p)$ = PageRank of p with teleport into "good" pages only = TrustRank

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

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

SimRank

Random Walks from a Fixed Node
on k-Partite Graphs

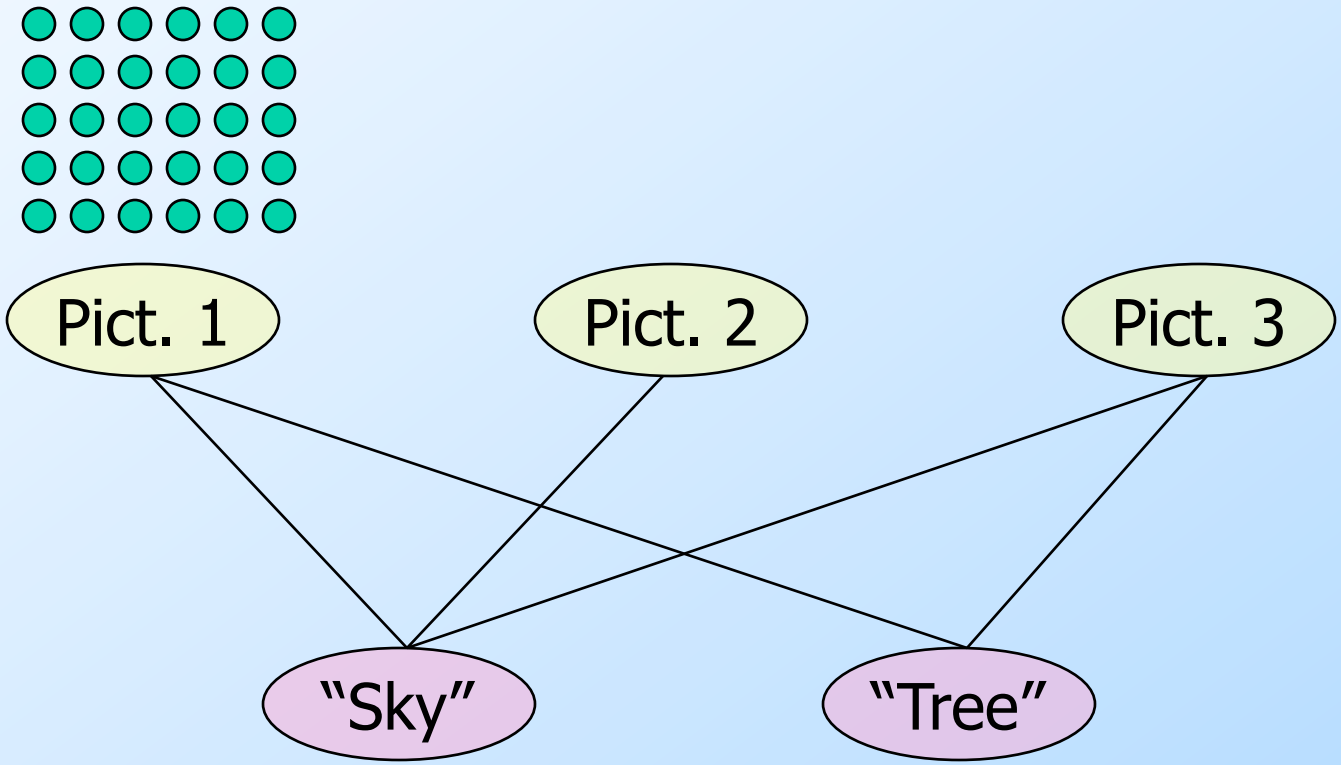
SimRank

- Setting: a *k* -partite graph with *k* types of nodes.
  - Example: picture nodes and tag nodes.
- Perform a random-walk with restart from a particular node N.
  - I.e., teleport set = {N}.
- Resulting probability distribution measures similarity to N.

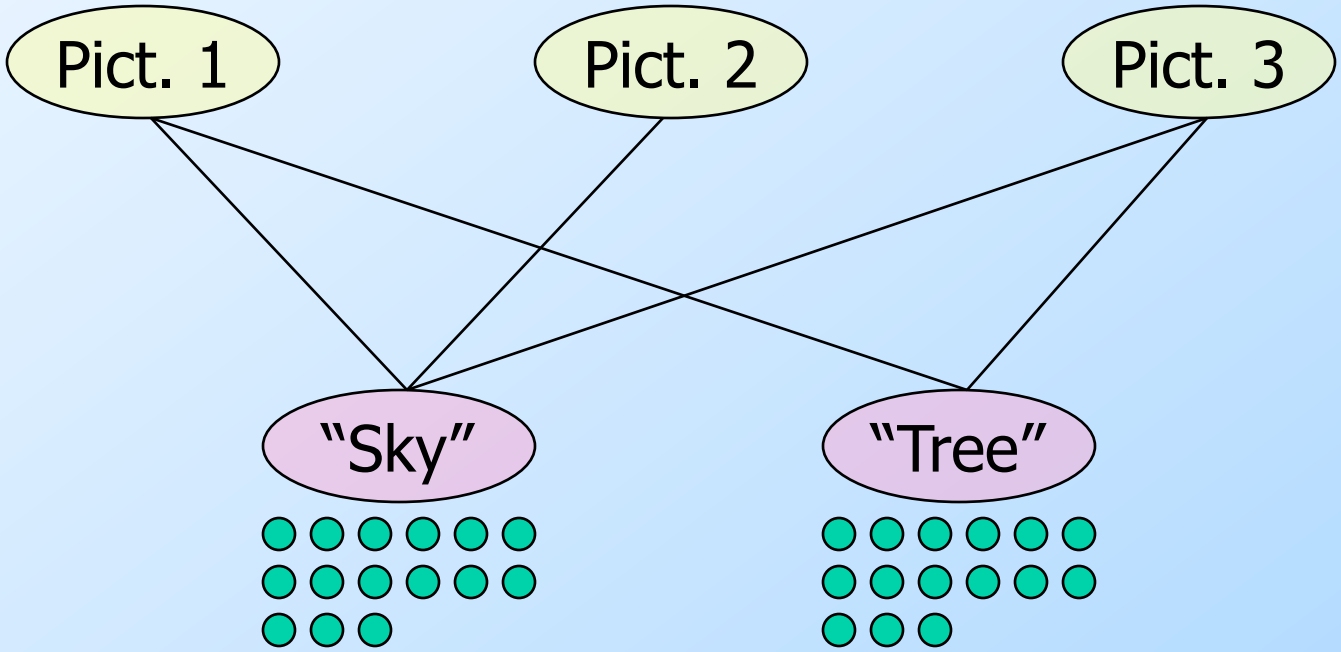
SimRank – (2)

- Problem: must be done once for each node of one type.
- But suitable for sub-Web-scale applications.
- Example: CleverSense measures similarity of the 400K US restaurants by key phrases in their reviews.
  - Startup based on CS345A.

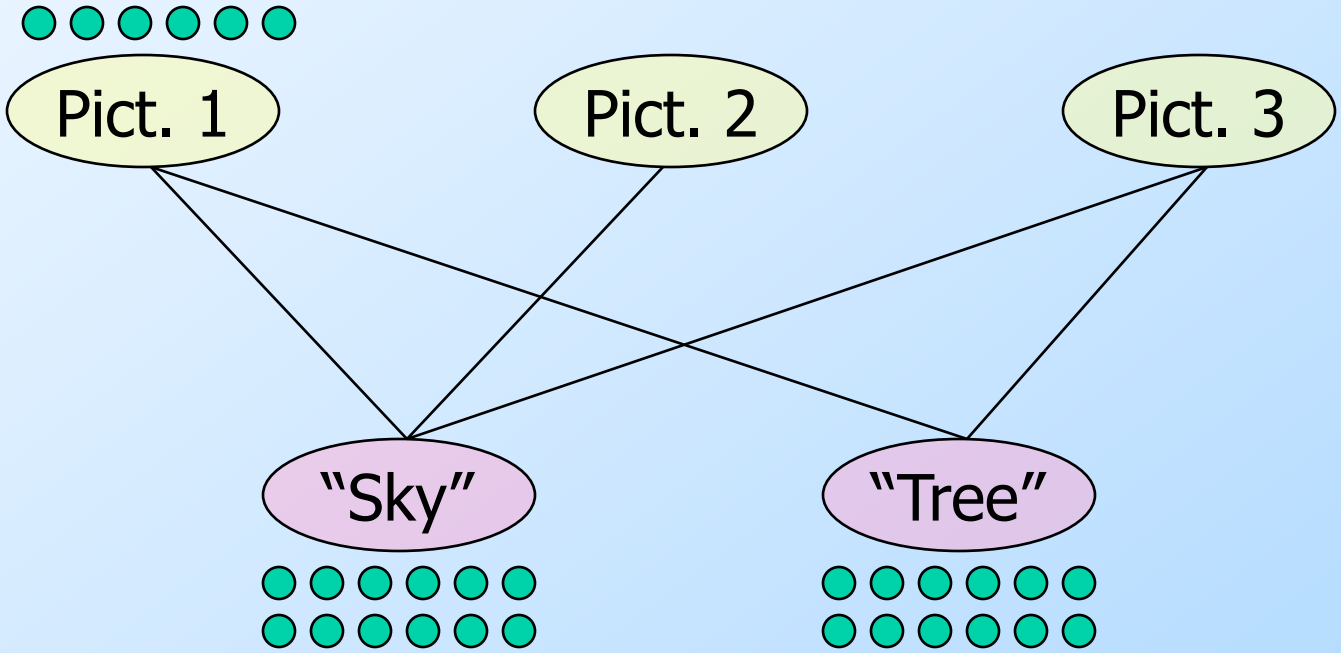
Example: Similarity to Pict. 1



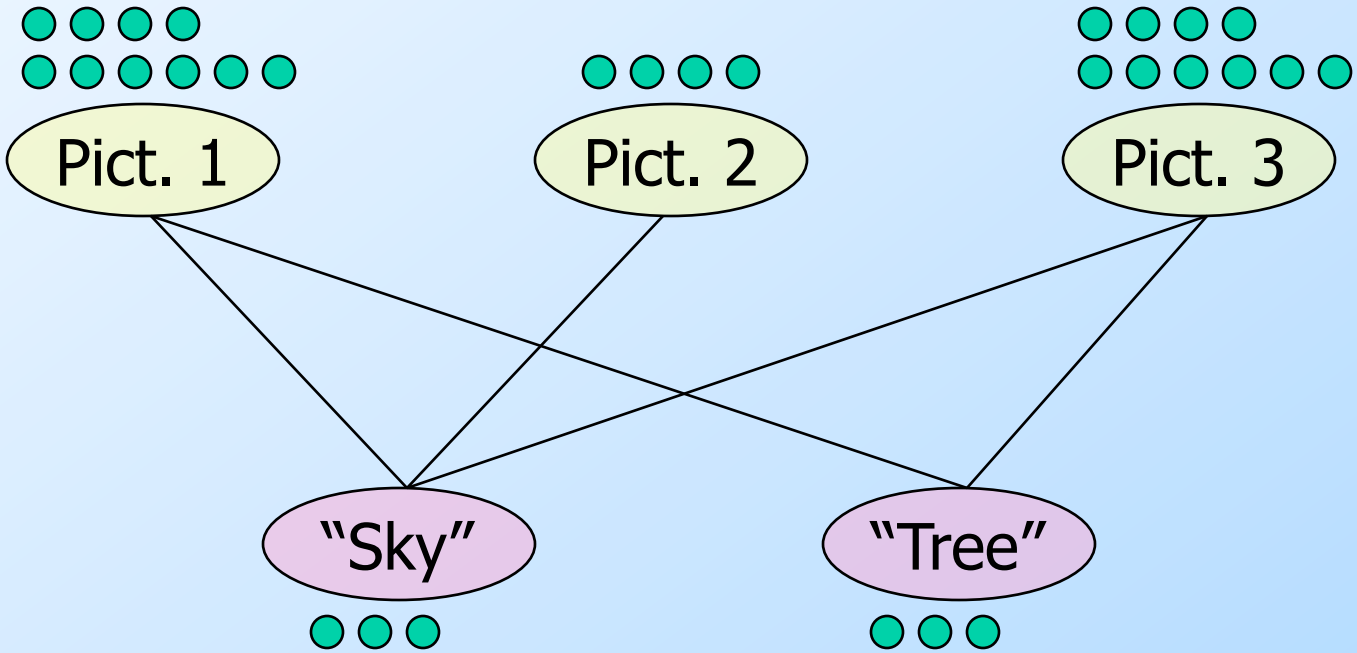
Example: Walk One Step



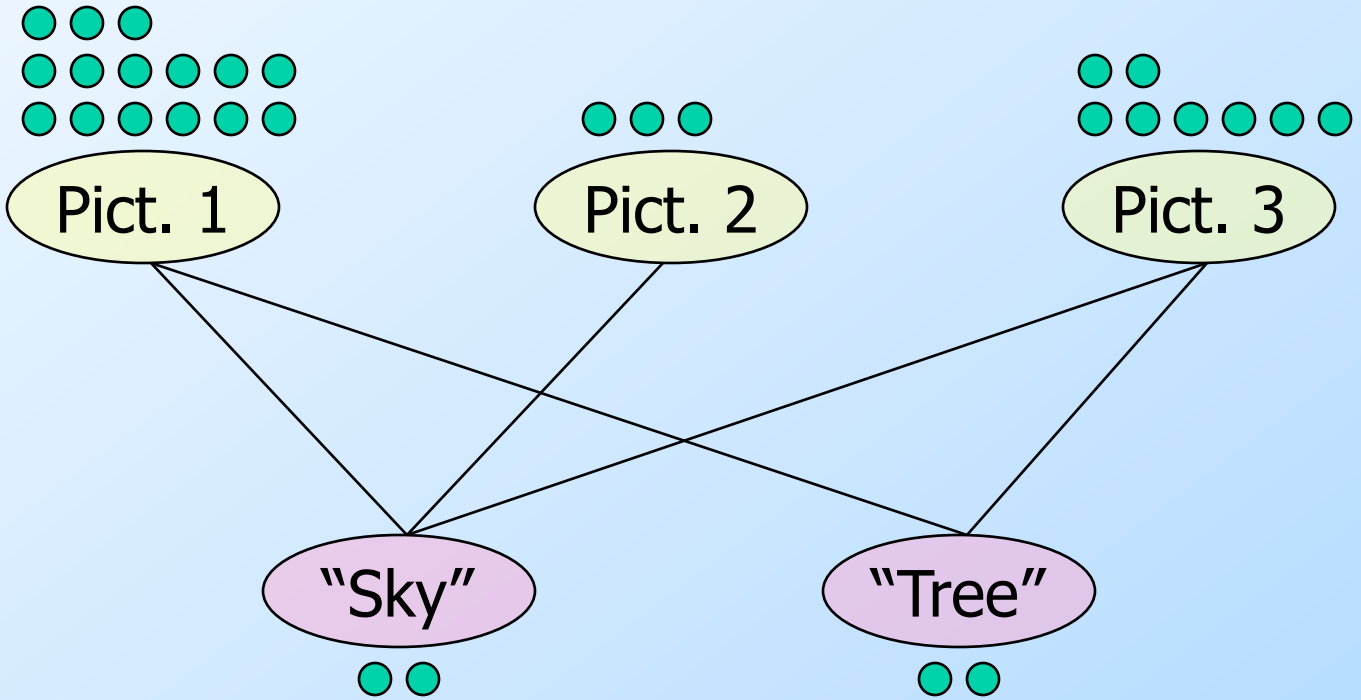
Example: Tax 20%



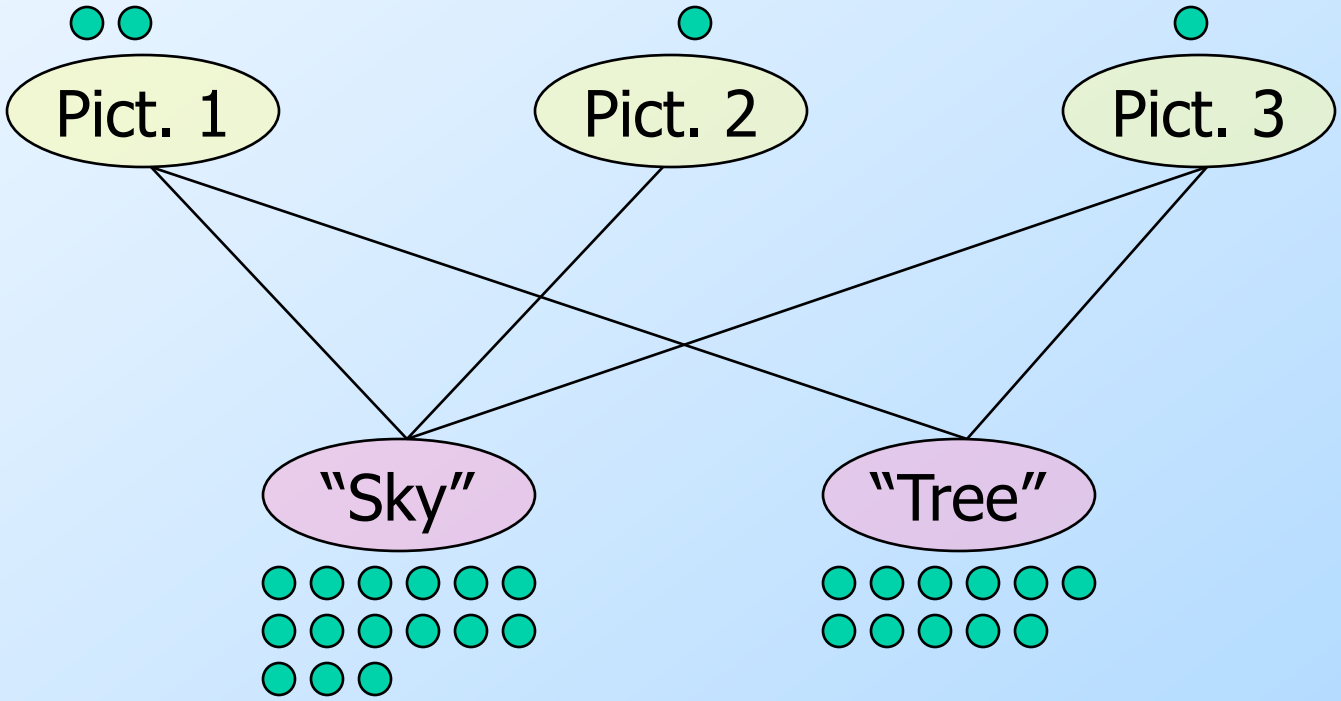
Example: Walk Second Step



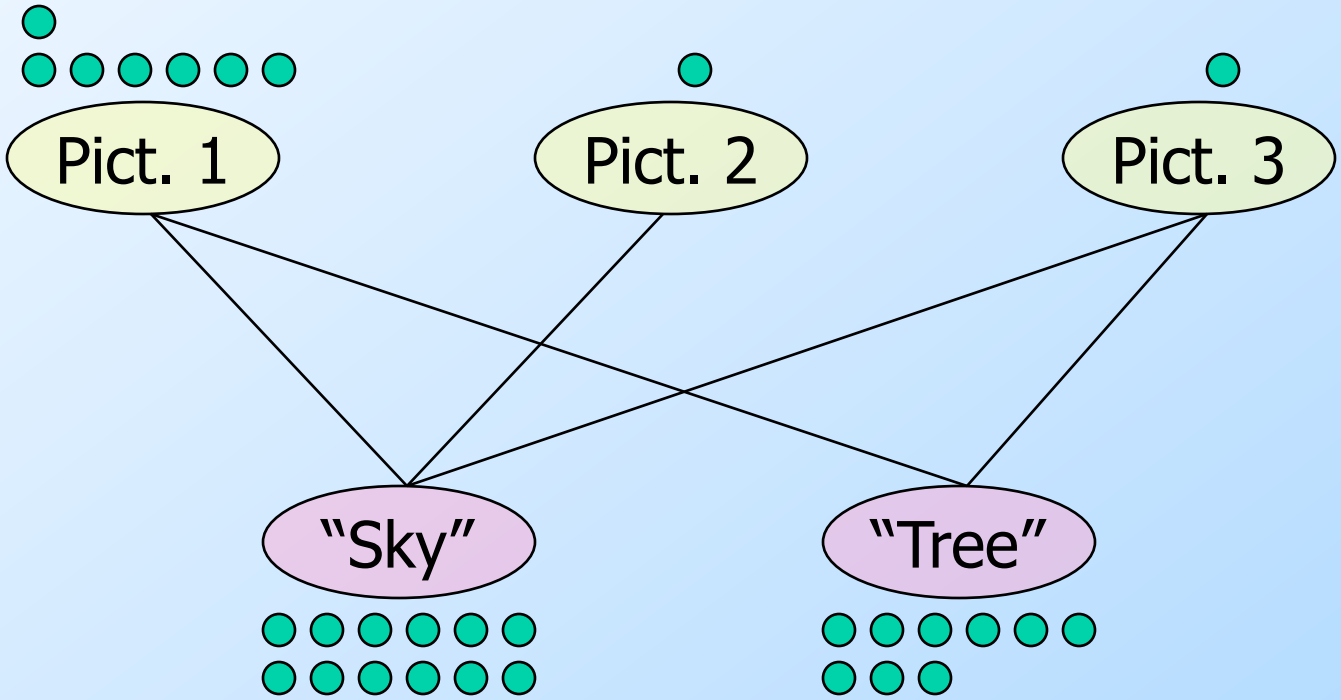
Example: Tax 20%



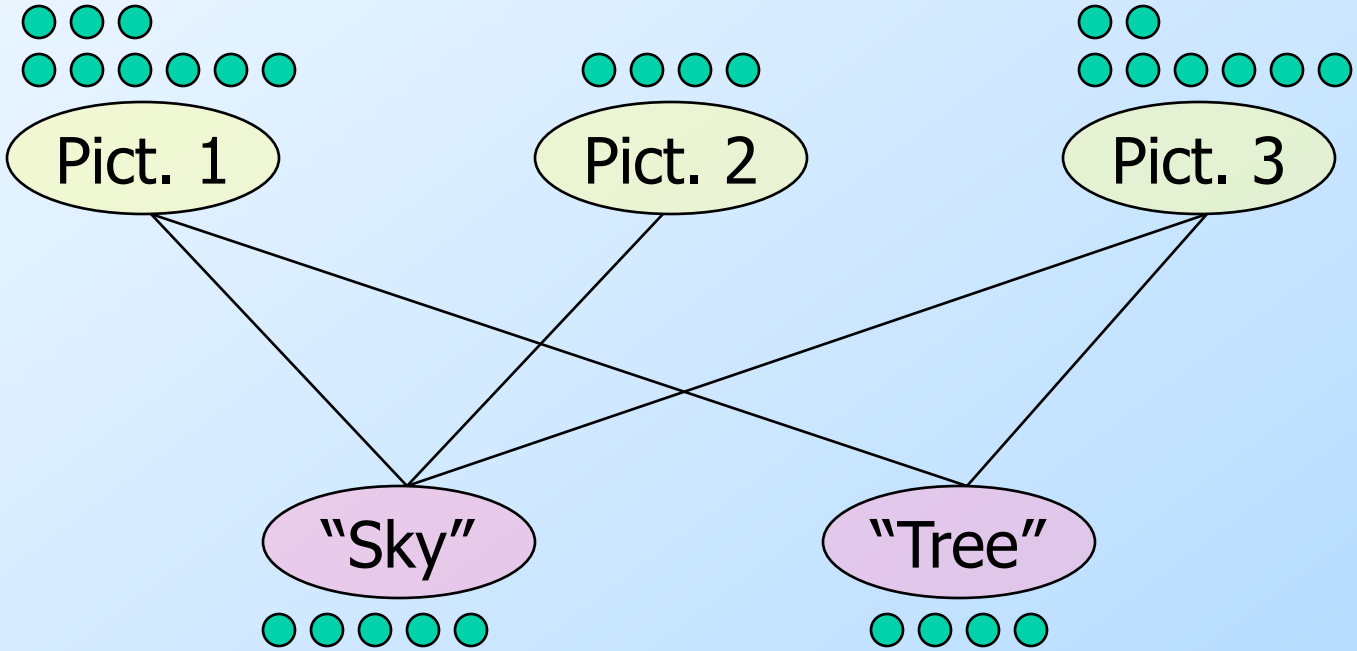
Example: Walk Third Step



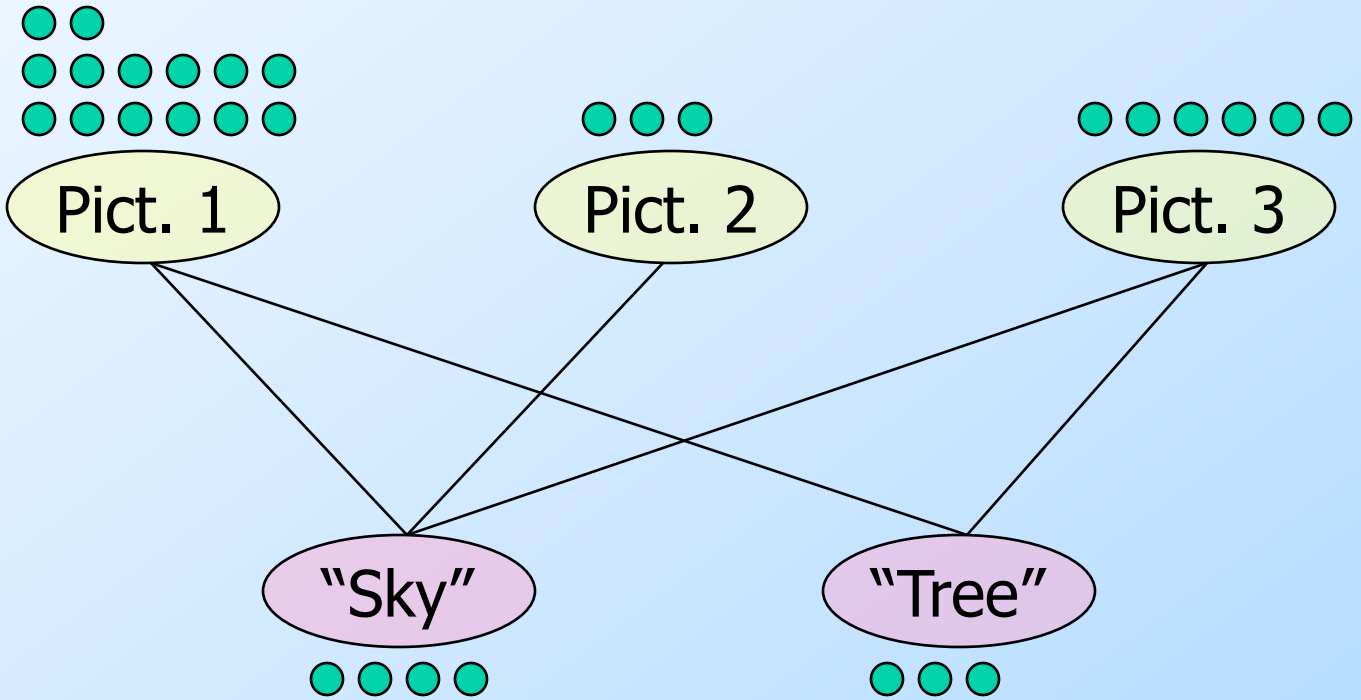
Example: Tax 20%



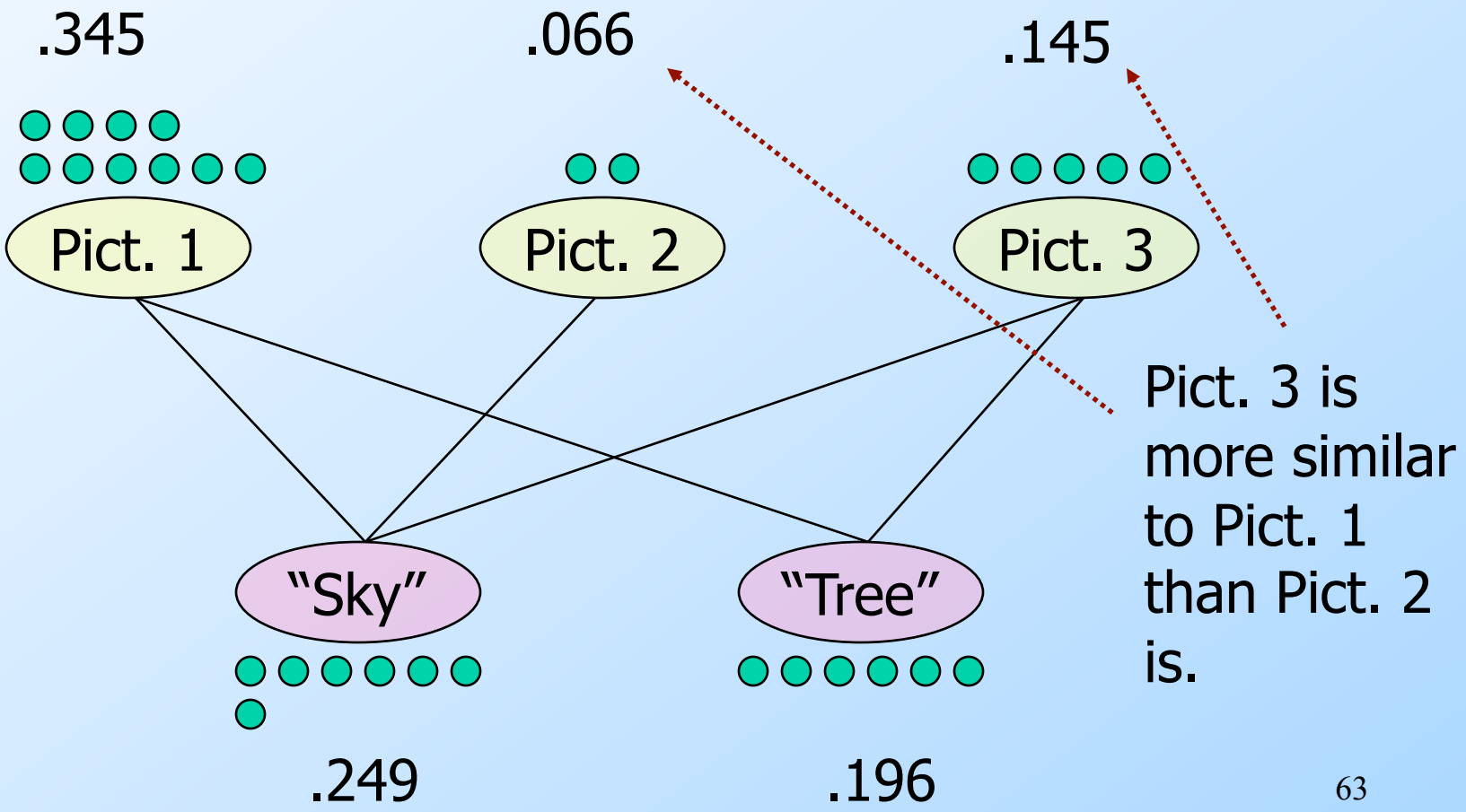
Example: Walk Fourth Step



Example: Tax 20%



Example: In the Limit



Hubs and Authorities

Matrix Formulation

Bipartite Cores and Secondary Cores

Hubs and Authorities

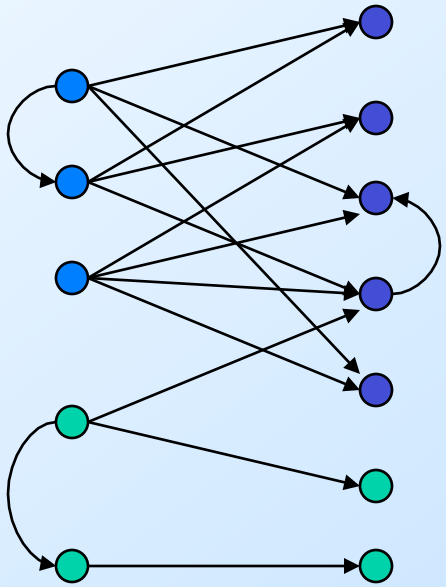
- HITS (*Hypertext-Induced Topic Selection*) is a measure of importance of pages or documents, similar to PageRank.
  - Proposed at approximately the same time (1998).
  - But never changed the world.

HITS Model

- Interesting documents fall into two classes
- Authorities are pages containing useful information
  - E.g., course home pages
- Hubs are pages that link to authorities
  - On-line list of links to CS courses.

Idealized view

Hubs Authorities



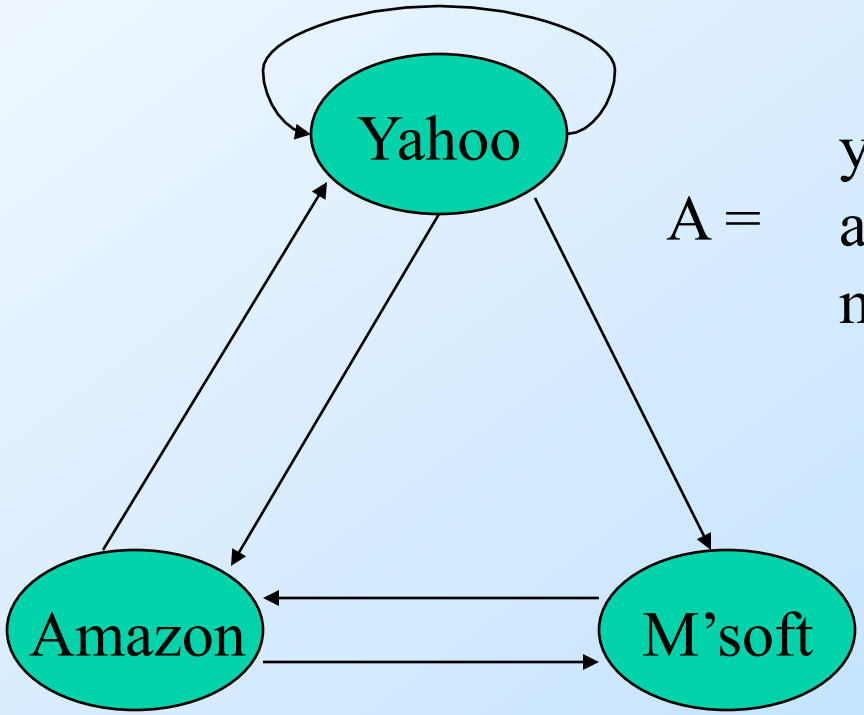
Mutually Recursive Definition

- 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



$A =$

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

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 a second scale factor

Iterative Algorithm

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

Example

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

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

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

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

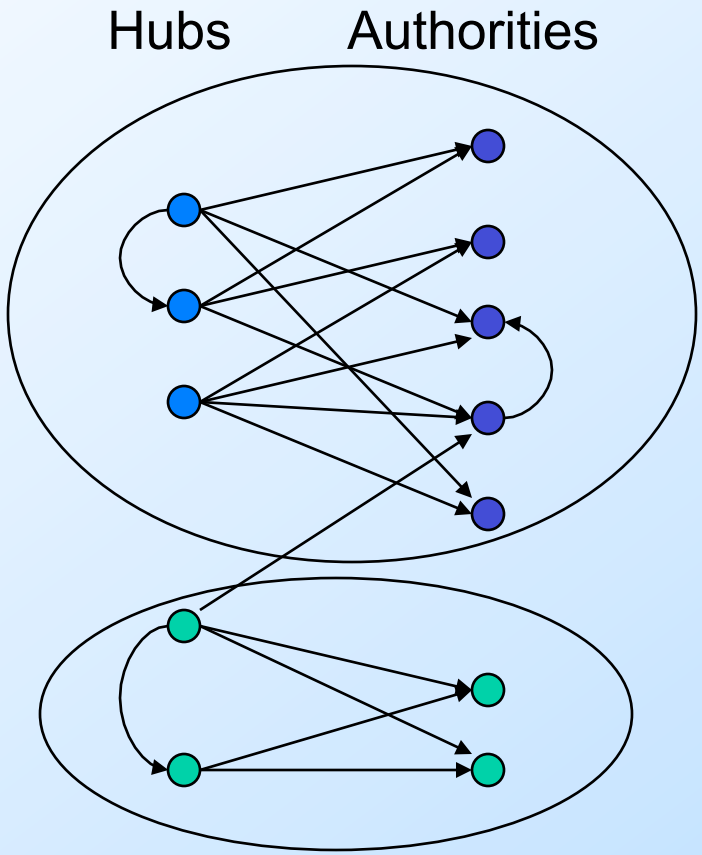
$\mathbf{h} = \lambda\mu AA^T\mathbf{h}$

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

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?

Non-Primary Eigenvectors

- AA^T and A^TA have the same set of eigenvalues
  - An eigenpair is the pair of eigenvectors with the same eigenvalue
  - The primary eigenpair (largest eigenvalue) is what we get from the iterative algorithm
- Non-primary eigenpairs correspond to other bipartite cores
  - The eigenvalue is a measure of the density of links in the core

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
- Technically, not exactly equivalent to non-primary eigenpair approach