Analysis of Large Graphs: Link Analysis, PageRank

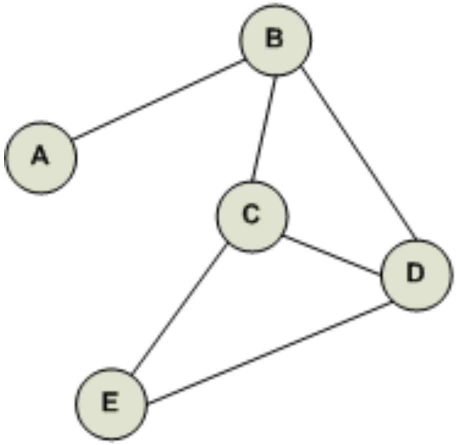
Big Data Analytics CSCI 4030

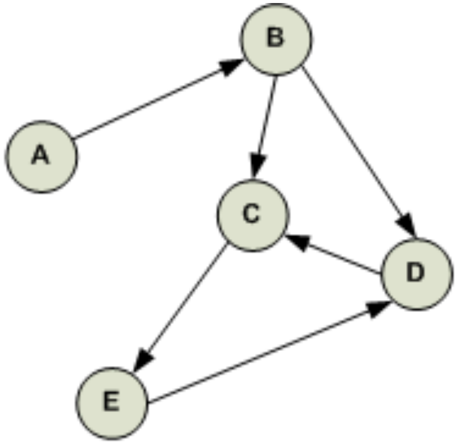
New Topic: Graph Data!

High dim. data	Graph data	Infinite data	Machine learning	Apps
Locality sensitive hashing	**PageRank,** SimRank	Filtering data streams	SVM	Recommender systems
Clustering	Community Detection	Web advertising	Decision Trees	Association Rules
Dimensionality reduction	Spam Detection	Queries on streams	Perceptron, kNN	Duplicate document detection

Graphs

- A **graph** is a representation of a set of objects (e.g., users, computers, …) where some pairs of objects are connected by links
- Objects are called nodes or vertices
- Links are called edges
- The edges may be **directed** or **undirected**



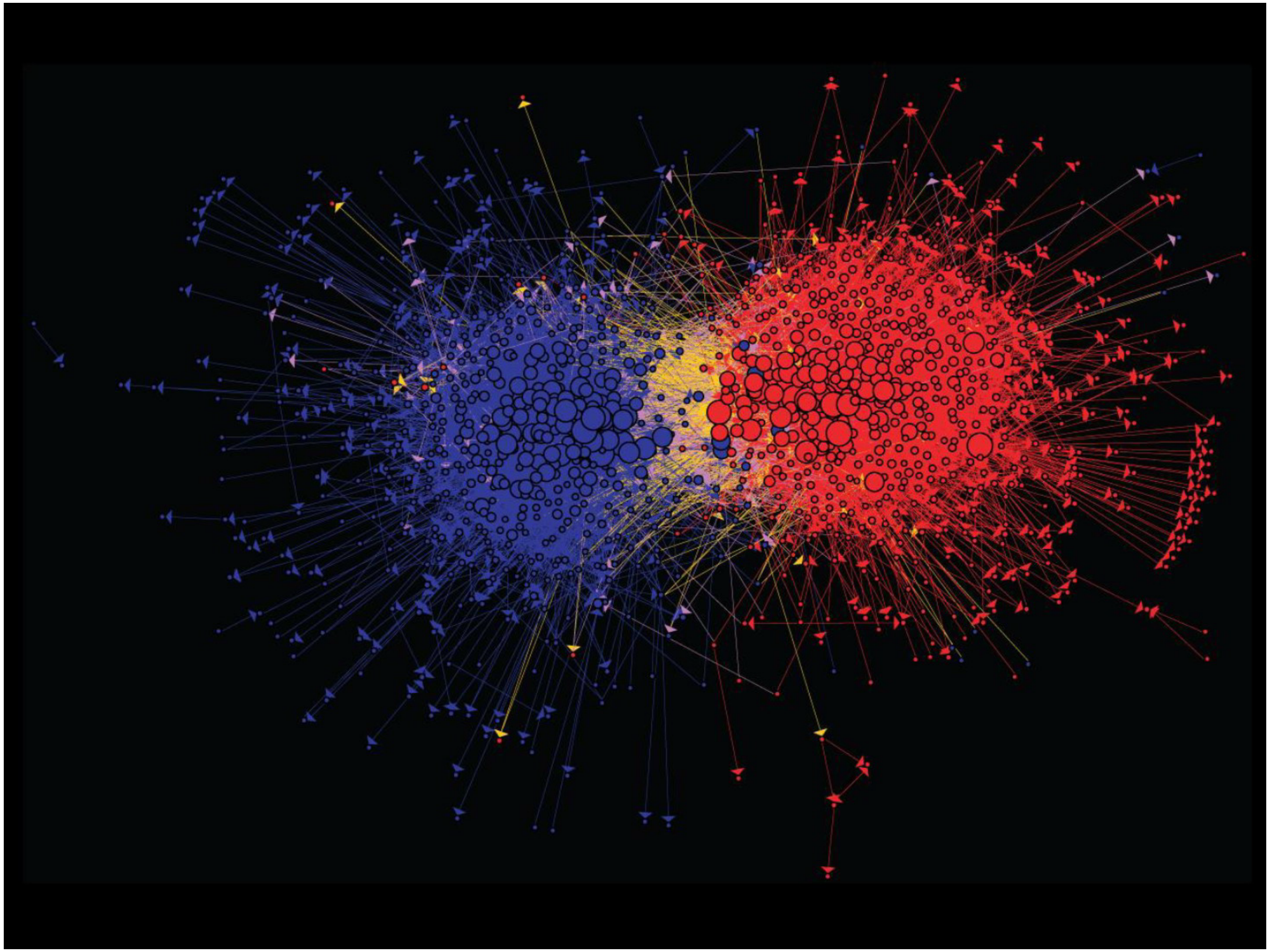


Graph Data: Social Networks

Facebook social graph

4-degrees of separation [Backstrom-Boldi-Rosa-Ugander-Vigna, 2011]

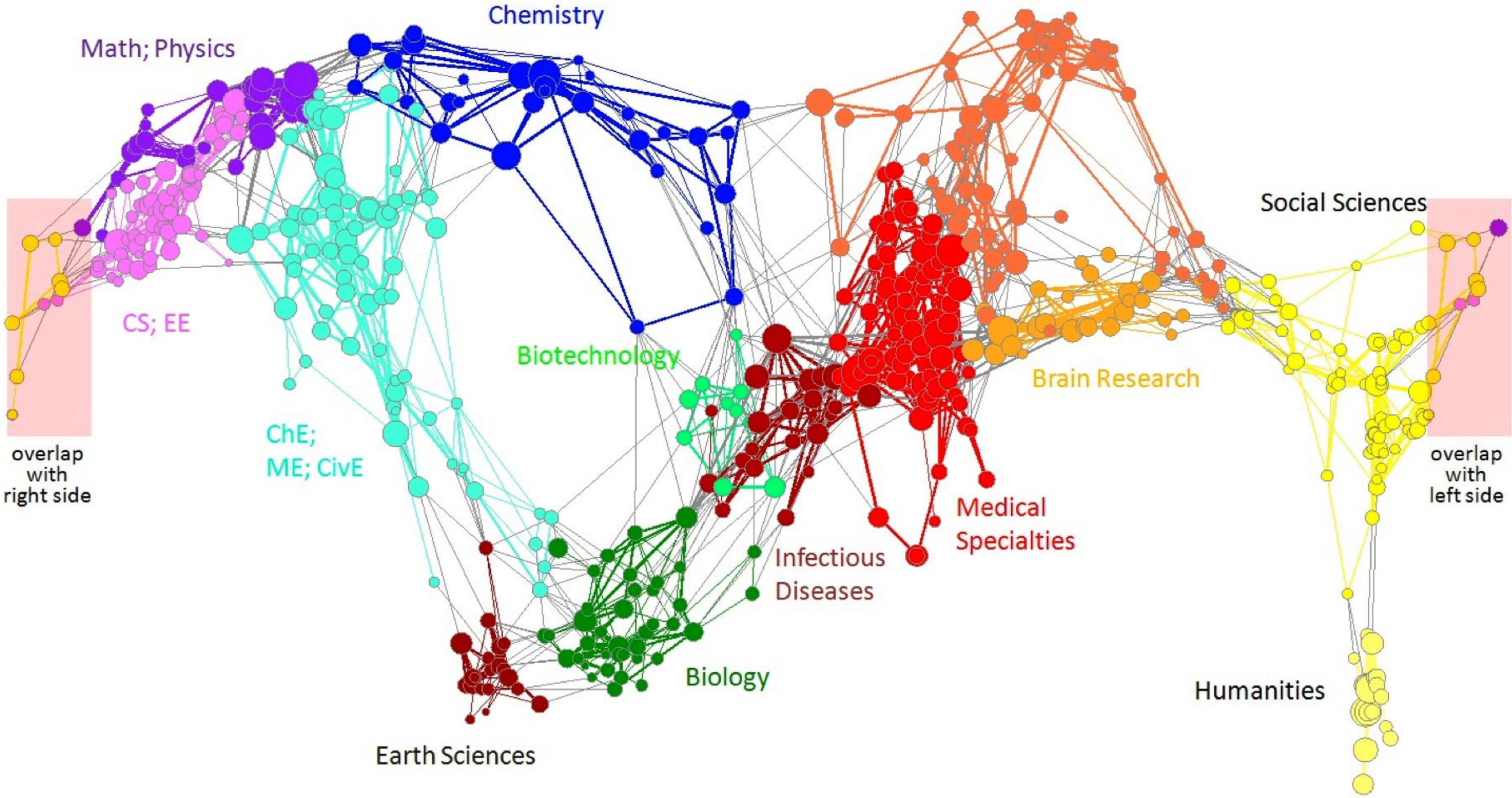
Graph Data: Media Networks



Connections between political blogs

Polarization of the network [Adamic-Glance, 2005]

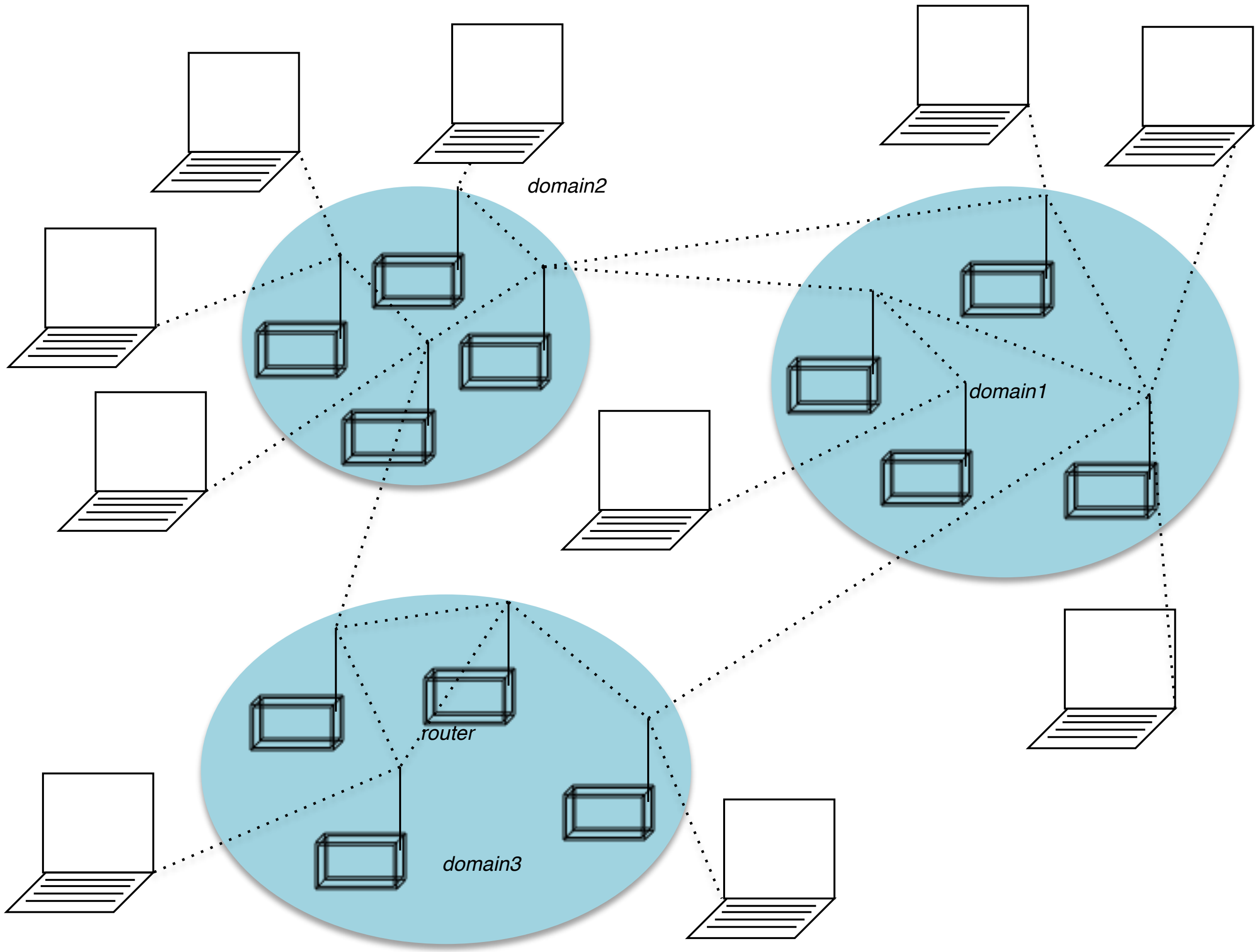
Graph Data: Information Nets



Citation networks and Maps of science

[Börner et al., 2012]

Graph Data: Communication Nets



Internet

Web as a Graph

- **Web as a directed graph:**
  - **Nodes: Webpages**
  - **Edges: Hyperlinks**

I teach a class on Databases.

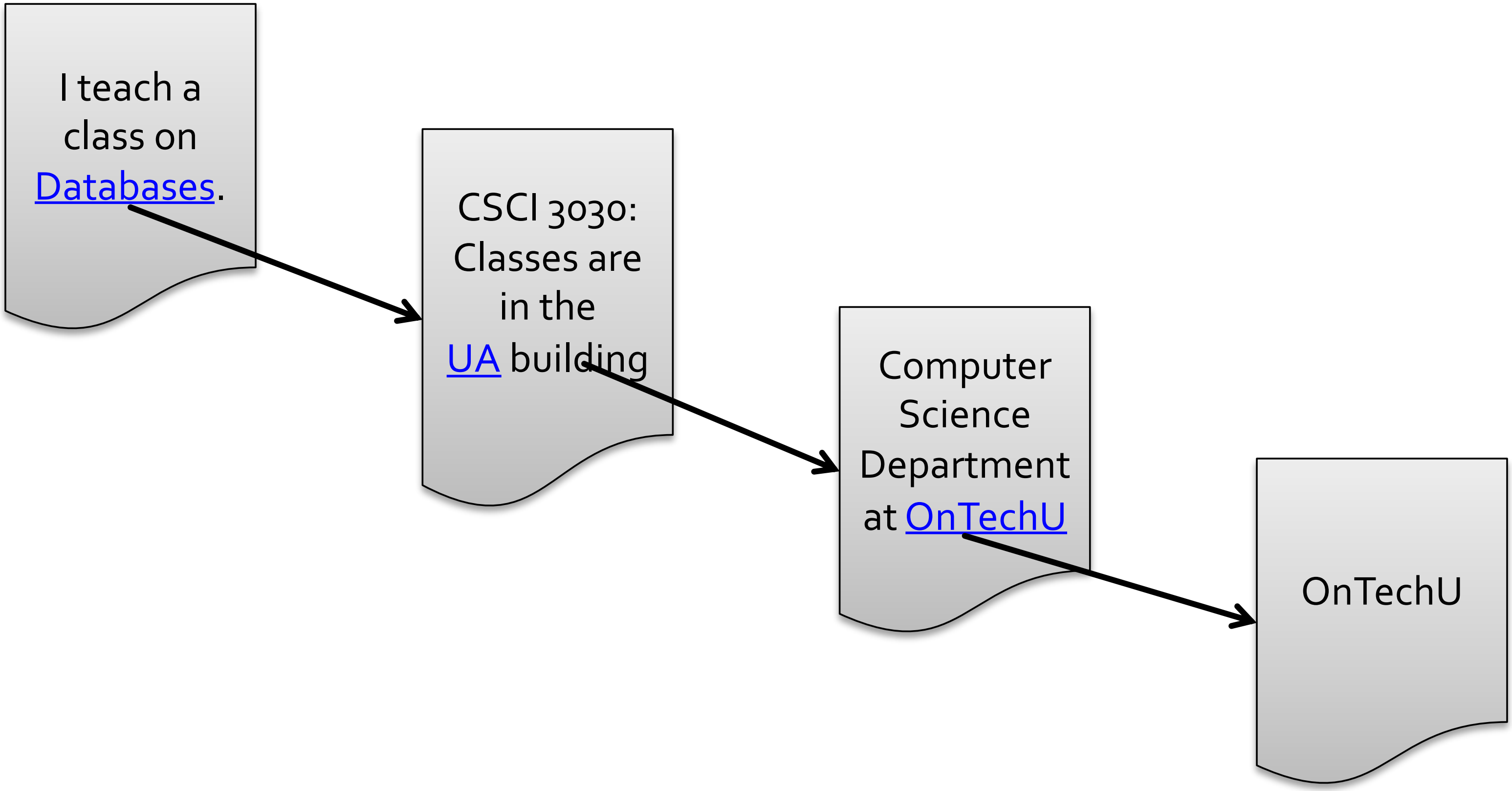
CSCI3030: Classes are in the UA building

Computer Science Department at OnTechU

OnTechU

Web as a Graph

- **Web as a directed graph:**
  - **Nodes: Webpages**
  - **Edges: Hyperlinks**



Broad Question

- **How to make the Web accessible?**
- **First try:** Human curated **Web directories**
  - Yahoo, DMOZ, LookSmart
- **Second try: Web Search**
  - **Information Retrieval**

But: Web is **huge**, full of **untrusted** documents, random things, web spam, etc.

Web Search: 2 Challenges

2 challenges of web search:

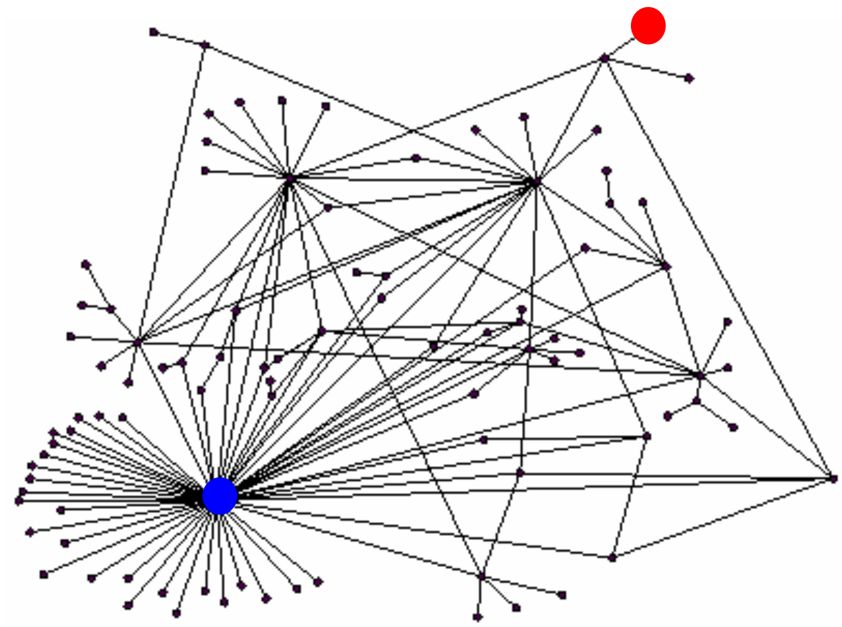
- **(1) Web contains many sources of information Who to "trust"?**
  - **Trick:** Trustworthy pages may point to each other!
- **(2) What is the "best" answer to query "newspaper"?**
  - No single right answer
  - **Trick:** Pages that actually know about newspapers might all be pointing to many newspapers

Ranking Nodes on the Graph

- **All web pages are not equally "important"**

  www.joe-schmoe.com vs. www.stanford.edu

- **Let's rank the pages by the link structure!**



Link Analysis Algorithms

- We will cover the following **Link Analysis approaches** for computing **importances** of nodes in a graph:

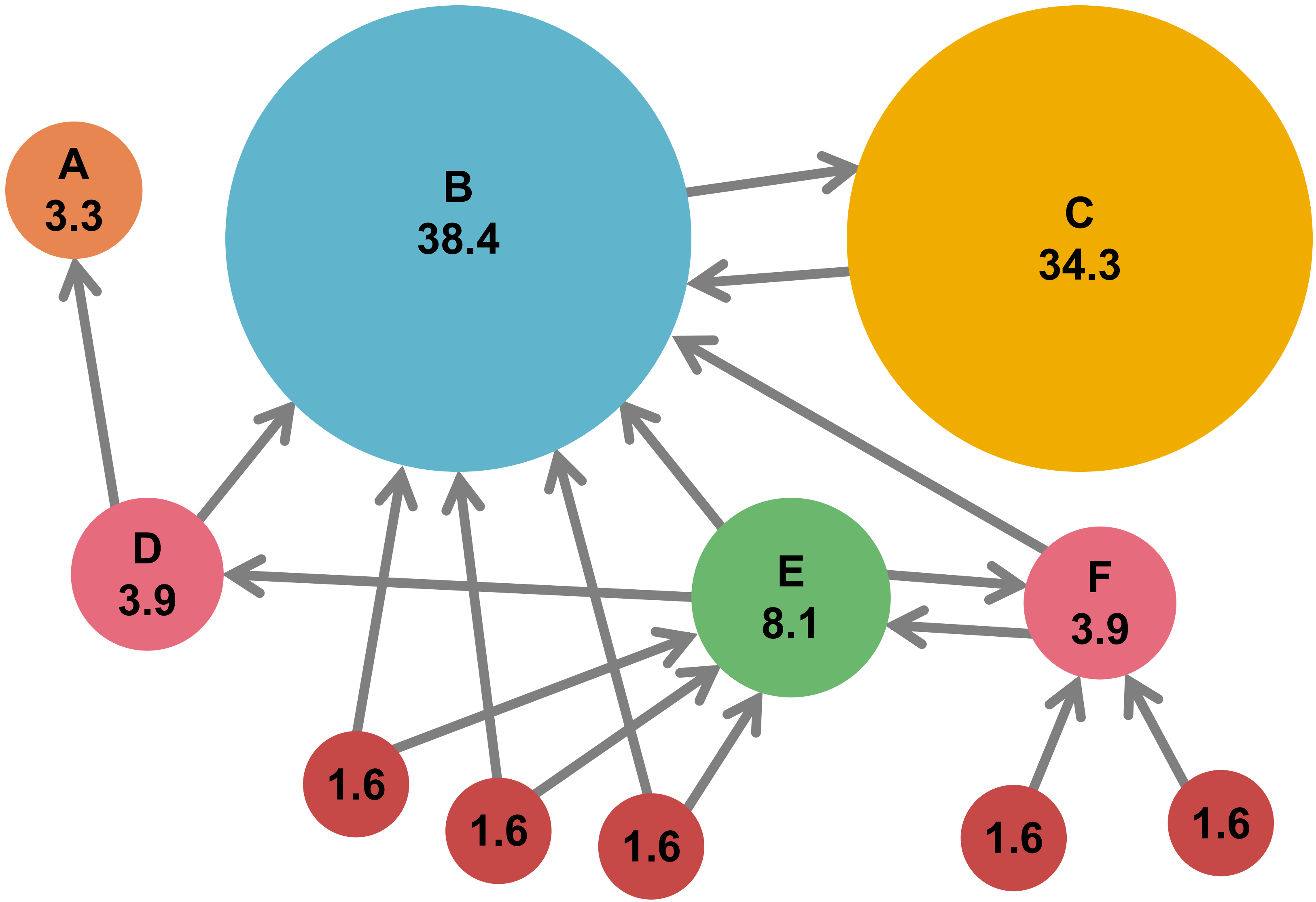
  - Page Rank
  - Topic-Specific (Personalized) Page Rank
  - Web Spam Detection Algorithms

PageRank:
The “Flow” Formulation

Links as Votes

- **Idea: Links as votes**
  - **Page is more important if it has more links**
    - In-coming links? Out-going links?
- **Think of in-links as votes:**
  - www.stanford.edu has 23,400 in-links
  - www.joe-schmoe.com has 1 in-link
- **Are all in-links are equal?**
  - **Links from important pages count more**
  - Recursive question!

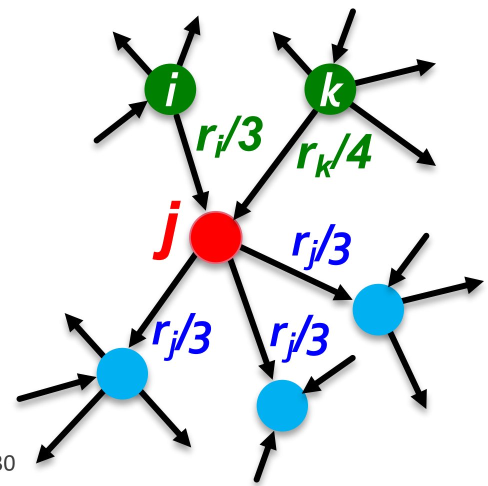
Example: PageRank Scores



Simple Recursive Formulation

- Each link's vote is proportional to the **importance** of its **source page**
- If page ***j*** with importance r_j has ***n*** out-links, each link gets r_j / n votes
- Page ***j***'s own importance is the sum of the votes on its in-links

$$r_j = r_i/3 + r_k/4$$



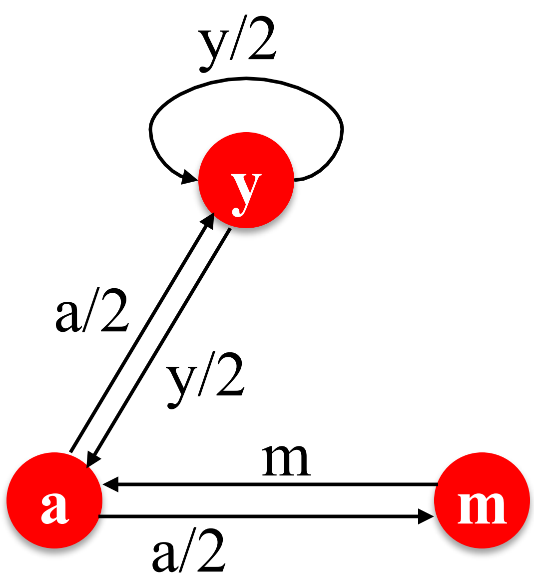
PageRank: The "Flow" Model

- A "vote" from an important page is worth more
- A page is important if it is pointed to by other important pages
- Define a "rank" r_j for page j

$$r_j = \sum_{i \to j} \frac{r_i}{d_i}$$

d_i … out-degree of node i

The web in 1839



"Flow" equations:

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

Solving the Flow Equations

- **3 equations, 3 unknowns, no constants**
  - No unique solution

Flow equations:

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

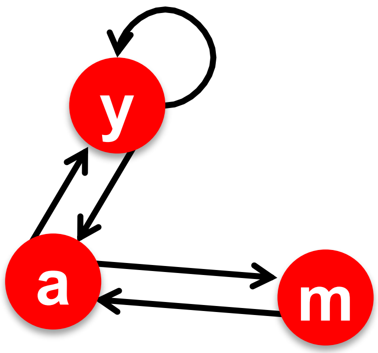
- **Additional constraint forces uniqueness:**
  - $\boldsymbol{r_y + r_a + r_m = 1}$
  - **Solution:** $\boldsymbol{r_y = \frac{2}{5},\ r_a = \frac{2}{5},\ r_m = \frac{1}{5}}$
- **Gaussian elimination method** (an algorithm for solving linear equations) **works for small examples, but we need a better method for large web-size graphs**
- **We need a new formulation!**

PageRank: Matrix Formulation

- **Stochastic adjacency matrix $\boldsymbol{M}$**
  - Let page i has d_i out-links
  - If $i \rightarrow j$, then $M_{ji} = \frac{1}{d_i}$ else $M_{ji} = 0$
    - $\boldsymbol{M}$ is a **column stochastic matrix**
      - Columns sum to 1
- **Rank vector $\boldsymbol{r}$:** vector with an entry per page
  - r_i is the importance score of page i
  - $\sum_i r_i = 1$
- **The flow equations can be written**

$$\boldsymbol{r} = \boldsymbol{M} \cdot \boldsymbol{r}$$

Example: Flow Equations & M



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

r* = *M·r

$$r_y = r_y/2 + r_a/2$$
$$r_a = r_y/2 + r_m$$
$$r_m = r_a/2$$

$$\begin{bmatrix} y \\ a \\ m \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 1 \\ 0 & \frac{1}{2} & 0 \end{bmatrix} \begin{bmatrix} y \\ a \\ m \end{bmatrix}$$

Eigenvector Formulation

- **The flow equations can be written**

$$\boldsymbol{r} = \boldsymbol{M} \cdot \boldsymbol{r}$$

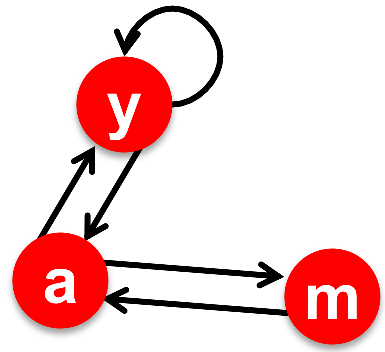
- The **rank vector** ***r*** is an **eigenvector** of the stochastic web matrix ***M***
  - ***M*** is column stochastic (with non-negative entries)
- **We can now efficiently solve for *r*!**
  The method is called Power iteration

Power Iteration Method

- **Given a web graph with *n* nodes, where the nodes are pages and edges are hyperlinks**
- **Power iteration:** a simple iterative scheme
  - Suppose there are *N* web pages
  - Initialize: $\mathbf{r}^{(0)} = [1/N,....,1/N]^T$
  - Iterate: $\mathbf{r}^{(t+1)} = \mathbf{M} \cdot \mathbf{r}^{(t)}$
  - Stop when $|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}| < \varepsilon$

PageRank: How to solve?

- **Power Iteration:**
  - $\mathbf{r}^{(0)} = [1/N,\ldots.,1/N]^T$
  - $\mathbf{r}^{(t+1)} = \mathbf{M} \cdot \mathbf{r}^{(t)}$
  - $|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}|_1 < \varepsilon$



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$$\mathbf{r_y} = \mathbf{r_y}/2 + \mathbf{r_a}/2$$
$$\mathbf{r_a} = \mathbf{r_y}/2 + \mathbf{r_m}$$
$$\mathbf{r_m} = \mathbf{r_a}/2$$

- **Example:**

$$\begin{pmatrix} r_y \\ r_a \\ r_m \end{pmatrix} = \begin{array}{cccccc} 1/3 & 1/3 & 5/12 & 9/24 & & 6/15 \\ 1/3 & 3/6 & 1/3 & 11/24 & \ldots & 6/15 \\ 1/3 & 1/6 & 3/12 & 1/6 & & 3/15 \end{array}$$

Iteration 0, 1, 2, …

Why Power Iteration works?

- **Power iteration:**
  - $\boldsymbol{r}^{(1)} = \boldsymbol{M} \cdot \boldsymbol{r}^{(0)}$
  - $\boldsymbol{r}^{(2)} = \boldsymbol{M} \cdot \boldsymbol{r}^{(1)} = \boldsymbol{M}\left(\boldsymbol{M}\boldsymbol{r}^{(1)}\right) = \boldsymbol{M}^{2} \cdot \boldsymbol{r}^{(0)}$
  - $\boldsymbol{r}^{(3)} = \boldsymbol{M} \cdot \boldsymbol{r}^{(2)} = \boldsymbol{M}\left(\boldsymbol{M}^{2}\boldsymbol{r}^{(0)}\right) = \boldsymbol{M}^{3} \cdot \boldsymbol{r}^{(0)}$
- **Claim:**
  Sequence $\boldsymbol{M} \cdot \boldsymbol{r}^{(0)}, \boldsymbol{M}^{2} \cdot \boldsymbol{r}^{(0)}, \ldots \boldsymbol{M}^{k} \cdot \boldsymbol{r}^{(0)}, \ldots$ will converge (under which **conditions**?!)

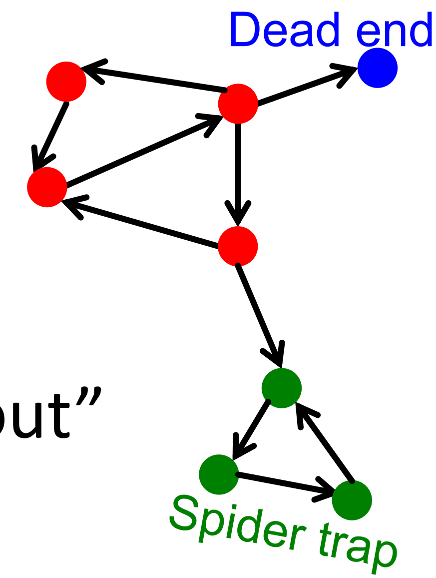
PageRank:
The Google Formulation

PageRank: Three Questions

$$r = Mr$$

- Does this converge?
- Does it converge to what we want?
- Are results reasonable?

PageRank: Problems

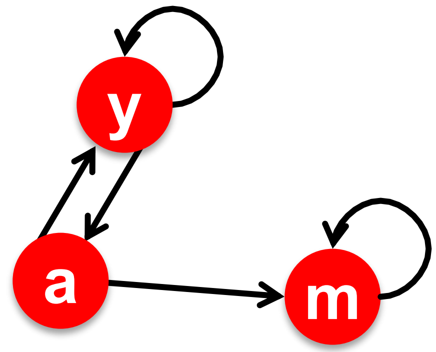


2 problems:

- **(1)** Some pages are **dead ends** (have no out-links)
  - Such pages cause importance to "leak out"
- **(2) Spider traps:** (all out-links are within the group)
  - And eventually spider traps absorb all importance

Problem: Spider Traps

- **Power Iteration:**



m is a spider trap

	y	a	m
y	½	½	0
a	½	0	0
m	0	½	1

$$\mathbf{r_y} = \mathbf{r_y}/2 + \mathbf{r_a}/2$$
$$\mathbf{r_a} = \mathbf{r_y}/2$$
$$\mathbf{r_m} = \mathbf{r_a}/2 + \mathbf{r_m}$$

- **Example:**

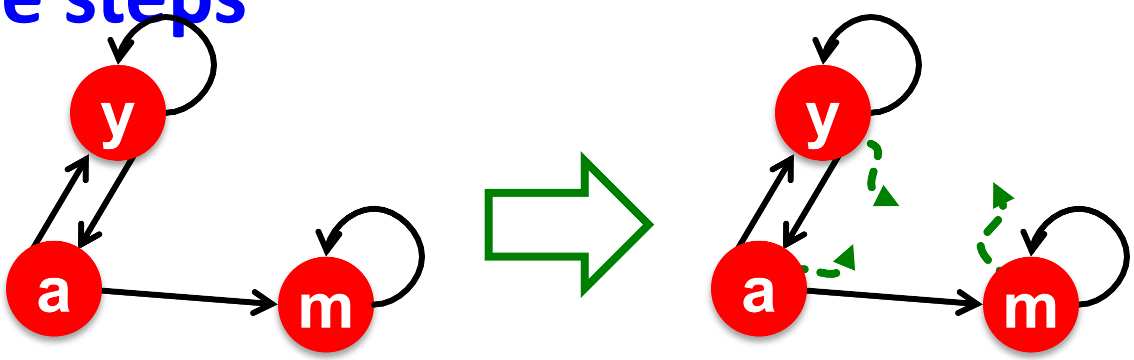
$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \dots & 0 \\ 1/3 & 3/6 & 7/12 & 16/24 & & 1 \end{matrix}$$

Iteration 0, 1, 2, …

All the PageRank score gets "trapped" in node m.

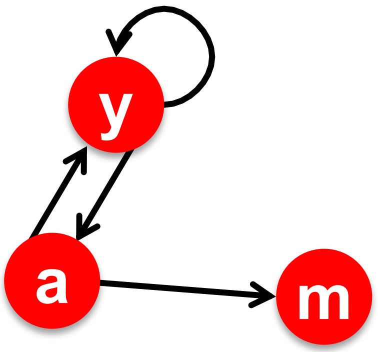
Solution: Teleports!

- **The Google solution for spider traps: At each time step, the random surfer has two options**
  - With probability β, follow a link at random
  - With probability **1-**β, jump to some random page
  - Common values for β are in the range 0.8 to 0.9
- **Surfer will teleport out of spider trap within a few time steps**



Problem: Dead Ends

- **Power Iteration:**



$$\begin{array}{c|c|c|c|} & y & a & m \\ \hline y & \frac{1}{2} & \frac{1}{2} & 0 \\ \hline a & \frac{1}{2} & 0 & 0 \\ \hline m & 0 & \frac{1}{2} & 0 \\ \hline \end{array}$$

$$\mathbf{r_y = r_y/2 + r_a/2}$$
$$\mathbf{r_a = r_y/2}$$
$$\mathbf{r_m = r_a/2}$$

- **Example:**

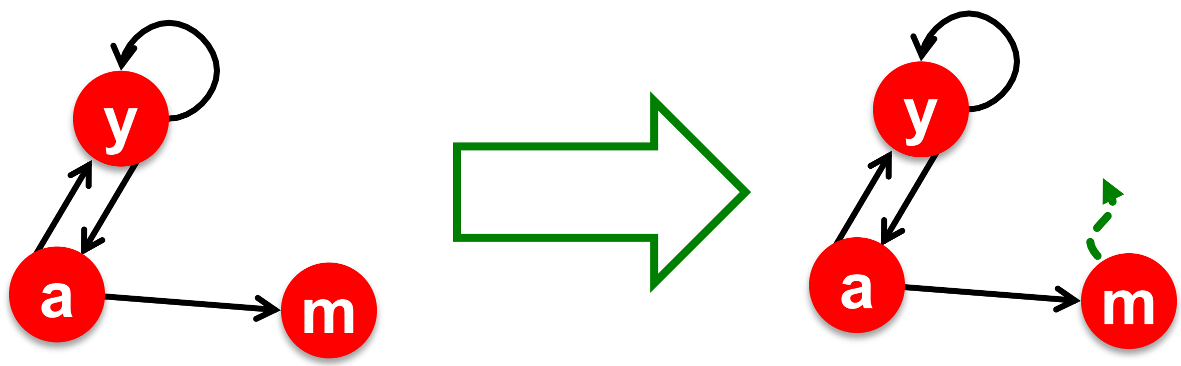
$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{array}{cccccc} 1/3 & 2/6 & 3/12 & 5/24 & & 0 \\ 1/3 & 1/6 & 2/12 & 3/24 & \ldots & 0 \\ 1/3 & 1/6 & 1/12 & 2/24 & & 0 \end{array}$$

Iteration 0, 1, 2, …

Here the PageRank "leaks" out since the matrix is not stochastic.

Solution: Always Teleport!

- **Teleports:** Follow random teleport links with probability 1.0 from dead-ends
  - Adjust matrix accordingly



	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

	y	a	m
y	½	½	⅓
a	½	0	⅓
m	0	½	⅓

Why Teleports Solve the Problem?

Why are dead-ends and spider traps a problem and why do teleports solve the problem?

- **Spider-traps** are not a problem, but with traps PageRank scores are **not** what we want
  - **Solution:** Never get stuck in a spider trap by teleporting out of it in a finite number of steps
- **Dead-ends** are a problem
  - The matrix is not column stochastic so our initial assumptions are not met
  - **Solution:** Make matrix column stochastic by always teleporting when there is nowhere else to go

Solution: Random Teleports

- <u>**Google's solution that does it all:**</u>
  At each step, random surfer has two options:
  - With probability β, follow a link at random
  - With probability $1-\beta$, jump to some random page

- **PageRank equation** [Brin-Page, 98]

$$r_j = \sum_{i \rightarrow j} \beta \frac{r_i}{d_i} + (1-\beta)\frac{1}{N}$$

d_i : out-degree of node i
***N*: total number of webpages**

The Google Matrix

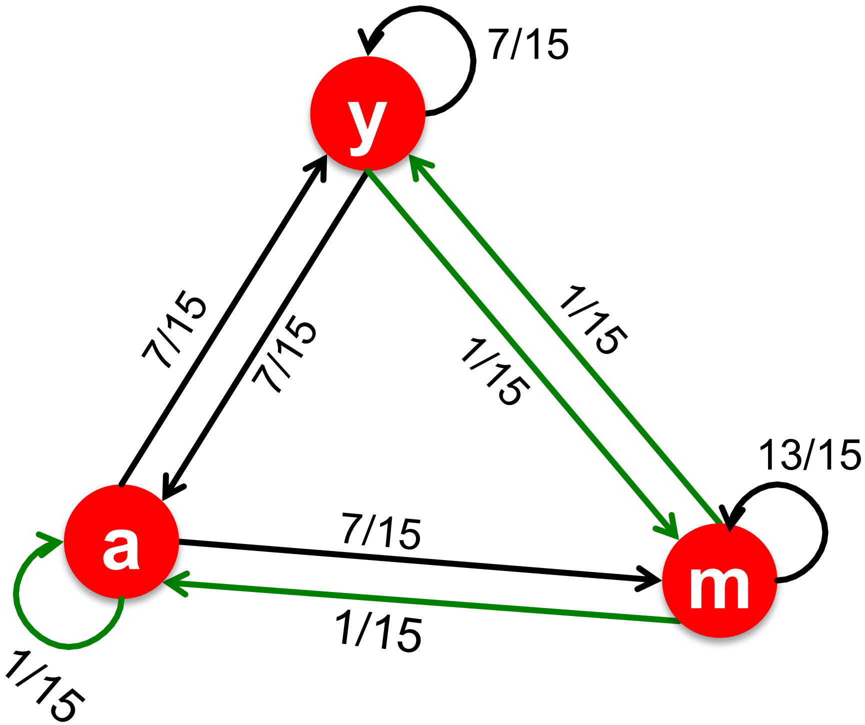
- **PageRank equation** [Brin-Page, '98]
- **The Google Matrix *A*:**

[1/N]$_{N\times N}$…N by N matrix where all entries are 1/N

$$A = \beta\, M + (1-\beta)\left[\frac{1}{N}\right]_{N\times N}$$

- **We have a recursive equation:**
  And the Power method still works!
- **What is β ?**
  - In practice $\beta = 0.8, 0.9$ (make 5 steps on avg., jump)

Random Teleports ($\beta = 0.8$)



$$0.8 \overset{\mathbf{M}}{\begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}} + 0.2 \overset{[1/N]_{N\times N}}{\begin{bmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{bmatrix}}$$

$$\begin{array}{c} y \\ a \\ m \end{array} \begin{bmatrix} 7/15 & 7/15 & 1/15 \\ 7/15 & 1/15 & 1/15 \\ 1/15 & 7/15 & 13/15 \end{bmatrix}$$

A

$$\begin{array}{c} y \\ a \\ m \end{array} = \begin{array}{c} 1/3 \\ 1/3 \\ 1/3 \end{array} \quad \begin{array}{c} 0.33 \\ 0.20 \\ 0.46 \end{array} \quad \begin{array}{c} 0.24 \\ 0.20 \\ 0.52 \end{array} \quad \begin{array}{c} 0.26 \\ 0.18 \\ 0.56 \end{array} \quad \cdots \quad \begin{array}{c} 7/33 \\ 5/33 \\ 21/33 \end{array}$$

Some Problems with Page Rank

- **Measures generic popularity of a page**
  - Biased against topic-specific authorities
  - **Solution:** Topic-Specific PageRank (**next**)

- **Sensitive to Link spam**
  - Artificial link topographies created in order to boost page rank
  - **Solution:** TrustRank

How do we actually compute the PageRank?

Computing Page Rank

- **Key step is matrix-vector multiplication**
  - $\boldsymbol{r}^{\text{new}} = \boldsymbol{A} \cdot \boldsymbol{r}^{\text{old}}$
- Easy if we have enough main memory to hold **A**, $\mathbf{r}^{\text{old}}$, $\mathbf{r}^{\text{new}}$
- **Say N = 1 billion pages**
  - We need 4 bytes for each entry (say)
  - 2 billion entries for vectors, approx 8GB
  - Matrix **A** has N^2 entries
    - 10^{18} is a large number!
- Google invented **MapReduce** and **Hadoop** mainly to calculate PageRank

Some Problems with Page Rank

- **Measures generic popularity of a page**
  - Biased against topic-specific authorities
  - **Solution:** Topic-Specific PageRank (**next**)

- **Sensitive to Link spam**
  - Artificial link topographies created in order to boost page rank
  - **Solution:** TrustRank

Topic-Specific PageRank

Topic-Specific PageRank

- **Instead of generic popularity, can we measure popularity within a topic?**
- **Example:** Query “Trojan” wants different pages depending on whether you are interested in sports, history and computer security
- **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 <u>interests</u> of the user**

Topic-Specific PageRank

- Random walker has a small probability of teleporting at any step
- **Teleport can go to:**
  - **Standard PageRank: Any page with equal probability**
    - To avoid dead-end and spider-trap problems
  - **Topic Specific PageRank: A topic-specific set of "relevant" pages (teleport set)**
- **Idea: Bias the random walk**
  - When walker teleports, she pick a page from a set ***S***
  - ***S*** contains only pages that are relevant to the topic
    - E.g., Open Directory (DMOZ) pages for a given topic/query

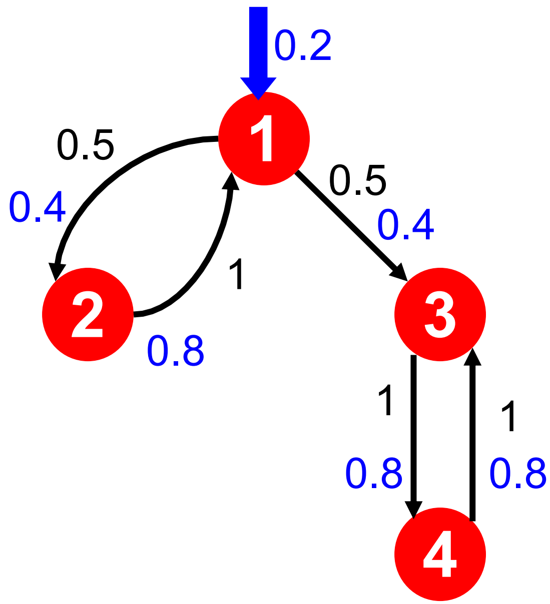
Matrix Formulation

- **To make this work all we need is to update the teleportation part of the PageRank formulation:**

$$A_{ij} = \begin{cases} \beta M_{ij} + (1-\beta)/|S| & \text{if } i \in S \\ \beta M_{ij} + 0 & \text{otherwise} \end{cases}$$

  - ***A*** is stochastic!
- We weighted all pages in the teleport set ***S*** equally
- **Compute as for regular PageRank:**
  - Multiply by ***M***, then add a vector

Example: Topic-Specific PageRank



Suppose $S = \{1\}$, $\beta = 0.8$

Node	Iteration 0	1	2	…	stable
1	0.25	0.4	0.28		0.294
2	0.25	0.1	0.16		0.118
3	0.25	0.3	0.32		0.327
4	0.25	0.2	0.24		0.261

S={1}, β=0.90:
r=[0.17, 0.07, 0.40, 0.36]
S={1} , β=0.8:
r=[0.29, 0.11, 0.32, 0.26]
S={1}, β=0.70:
r=[0.39, 0.14, 0.27, 0.19]

S={1,2,3,4}, β=0.8:
r=[0.13, 0.10, 0.39, 0.36]
S={1,2,3} , β=0.8:
r=[0.17, 0.13, 0.38, 0.30]
S={1,2} , β=0.8:
r=[0.26, 0.20, 0.29, 0.23]
S={1} , β=0.8:
r=[0.29, 0.11, 0.32, 0.26]

Discovering the Topic Vector S

- **Create different PageRanks for different topics**
  - The 16 DMOZ top-level categories:
    - arts, business, sports,…
- **Which topic ranking to use?**
  - User can pick from a menu
  - Can use the **context** of the query
    - E.g., query is launched from a web page talking about a known topic
    - History of queries e.g., “basketball” followed by “Jordan”
  - User context, e.g., user’s bookmarks, …

TrustRank:
Combating the Web Spam

What is Web Spam?

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

Web Search

- **Early search engines:**
  - Crawl the Web
  - Index pages by the **words** they contained
  - 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

First Spammers

- As people began to use search engines, those with commercial interests tried to **exploit search engines**

- **Example:**
  - Shirt-seller might pretend to be about "movies"
- **Techniques for achieving high relevance/importance for a web page**

First Spammers: Term Spam

- **How do you make your page appear to be about movies?**
  - **(1)** Add the word movie 1,000 times to your page
  - Set text color to the background color, so only search engines would see it
  - **(2)** Or, run the query “movie” on your target search engine
  - See what page came first in the listings
  - Copy it into your page, make it “invisible”
- **These and similar techniques are term spam**

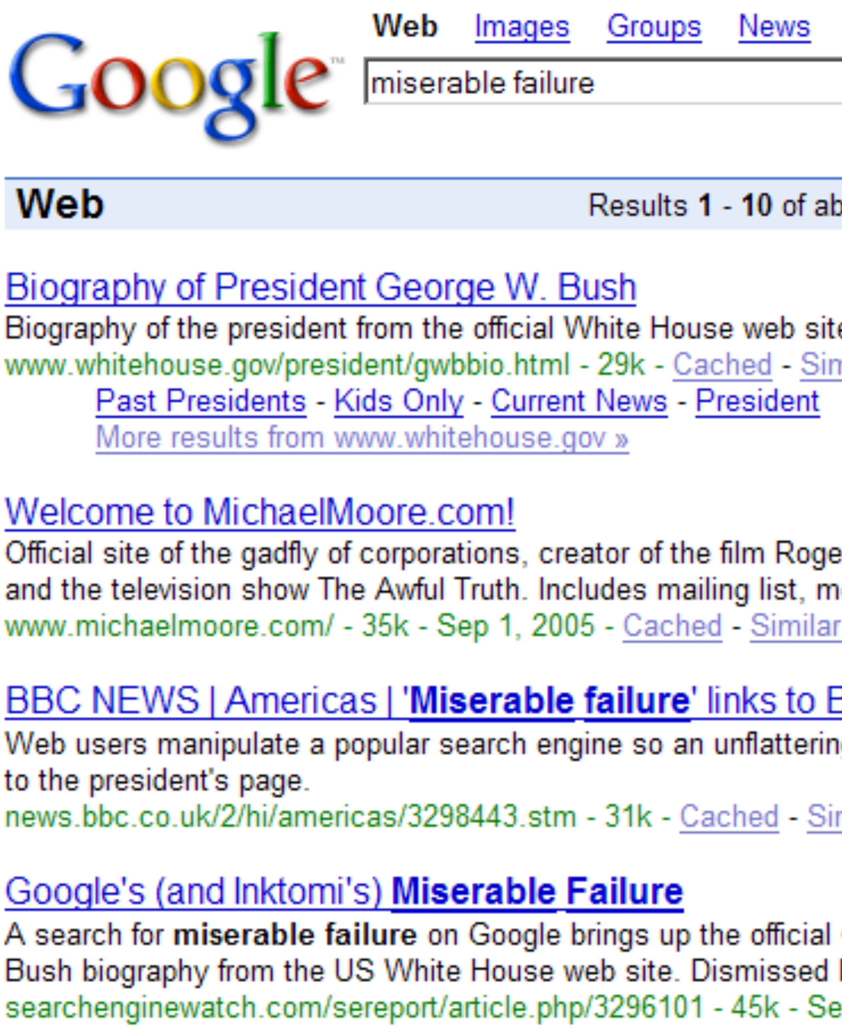
Google's Solution to Term Spam

1) Believe what people say about you, rather than what you say about yourself

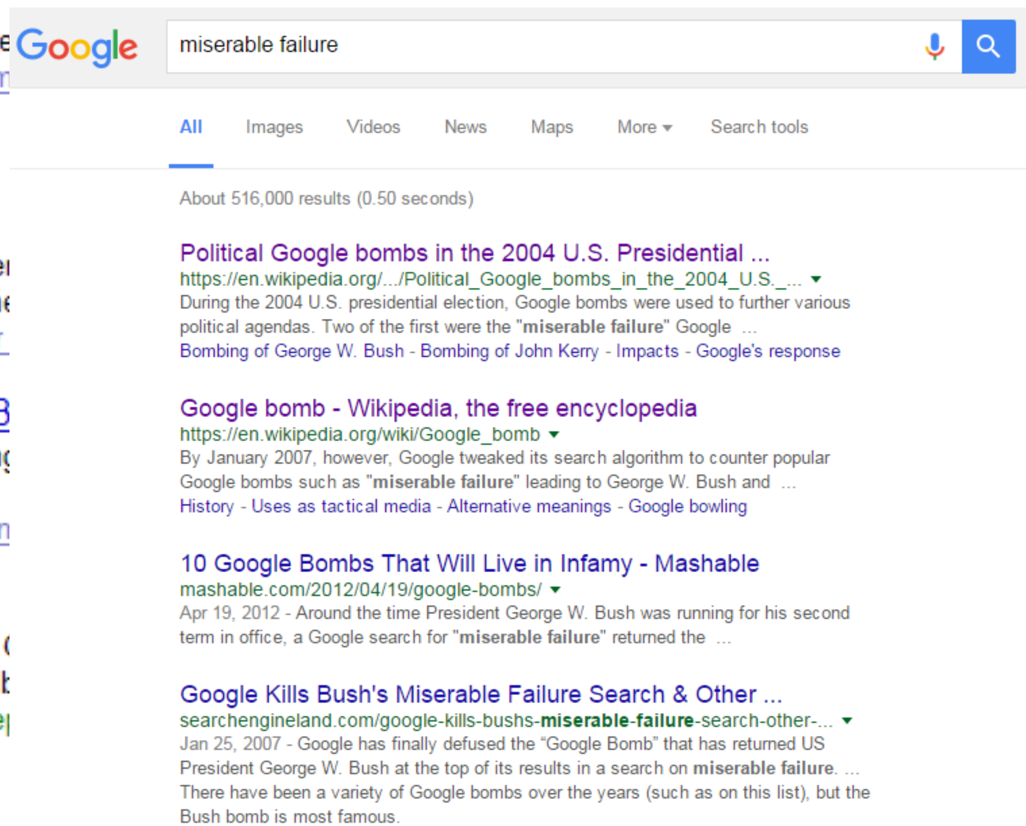
- Use words in the anchor text (<u>words that appear underlined to represent the link</u>) and its surrounding text

2) PageRank as a tool to measure the "importance" of Web pages

Google Bomb



Google
Web Images Groups News Froogle Local more »
miserable failure
Search
Advanced Search
Preferences
Web
Results 1 - 10 of about 969,000 for miserable failure. (0.06 seconds)
Biography of President George W. Bush
Biography of the president from the official White House web site
www.whitehouse.gov/president/gwbbio.html - 29k - Cached - Sim
Past Presidents - Kids Only - Current News - President
More results from www.whitehouse.gov »
Welcome to MichaelMoore.com!
Official site of the gadfly of corporations, creator of the film Roger
and the television show The Awful Truth. Includes mailing list, me
www.michaelmoore.com/ - 35k - Sep 1, 2005 - Cached - Similar
BBC NEWS | Americas | 'Miserable failure' links to B
Web users manipulate a popular search engine so an unflattering
to the president's page.
news.bbc.co.uk/2/hi/americas/3298443.stm - 31k - Cached - Sin
Google's (and Inktomi's) Miserable Failure
A search for miserable failure on Google brings up the official
Bush biography from the US White House web site. Dismissed b
searchenginewatch.com/sereport/article.php/3296101 - 45k - Sep



Google
miserable failure
All Images Videos News Maps More Search tools
About 516,000 results (0.50 seconds)
Political Google bombs in the 2004 U.S. Presidential ...
https://en.wikipedia.org/.../Political_Google_bombs_in_the_2004_U.S._...
During the 2004 U.S. presidential election, Google bombs were used to further various political agendas. Two of the first were the "miserable failure" Google ...
Bombing of George W. Bush - Bombing of John Kerry - Impacts - Google's response
Google bomb - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/Google_bomb
By January 2007, however, Google tweaked its search algorithm to counter popular Google bombs such as "miserable failure" leading to George W. Bush and ...
History - Uses as tactical media - Alternative meanings - Google bowling
10 Google Bombs That Will Live in Infamy - Mashable
mashable.com/2012/04/19/google-bombs/
Apr 19, 2012 - Around the time President George W. Bush was running for his second term in office, a Google search for "miserable failure" returned the ...
Google Kills Bush's Miserable Failure Search & Other ...
searchengineland.com/google-kills-bushs-miserable-failure-search-other-...
Jan 25, 2007 - Google has finally defused the "Google Bomb" that has returned US President George W. Bush at the top of its results in a search on miserable failure. ...
There have been a variety of Google bombs over the years (such as on this list), but the Bush bomb is most famous.


SPAM FARMING

Google vs. Spammers: Round 2!

- Once Google became the dominant search engine, spammers began to work out ways to fool Google

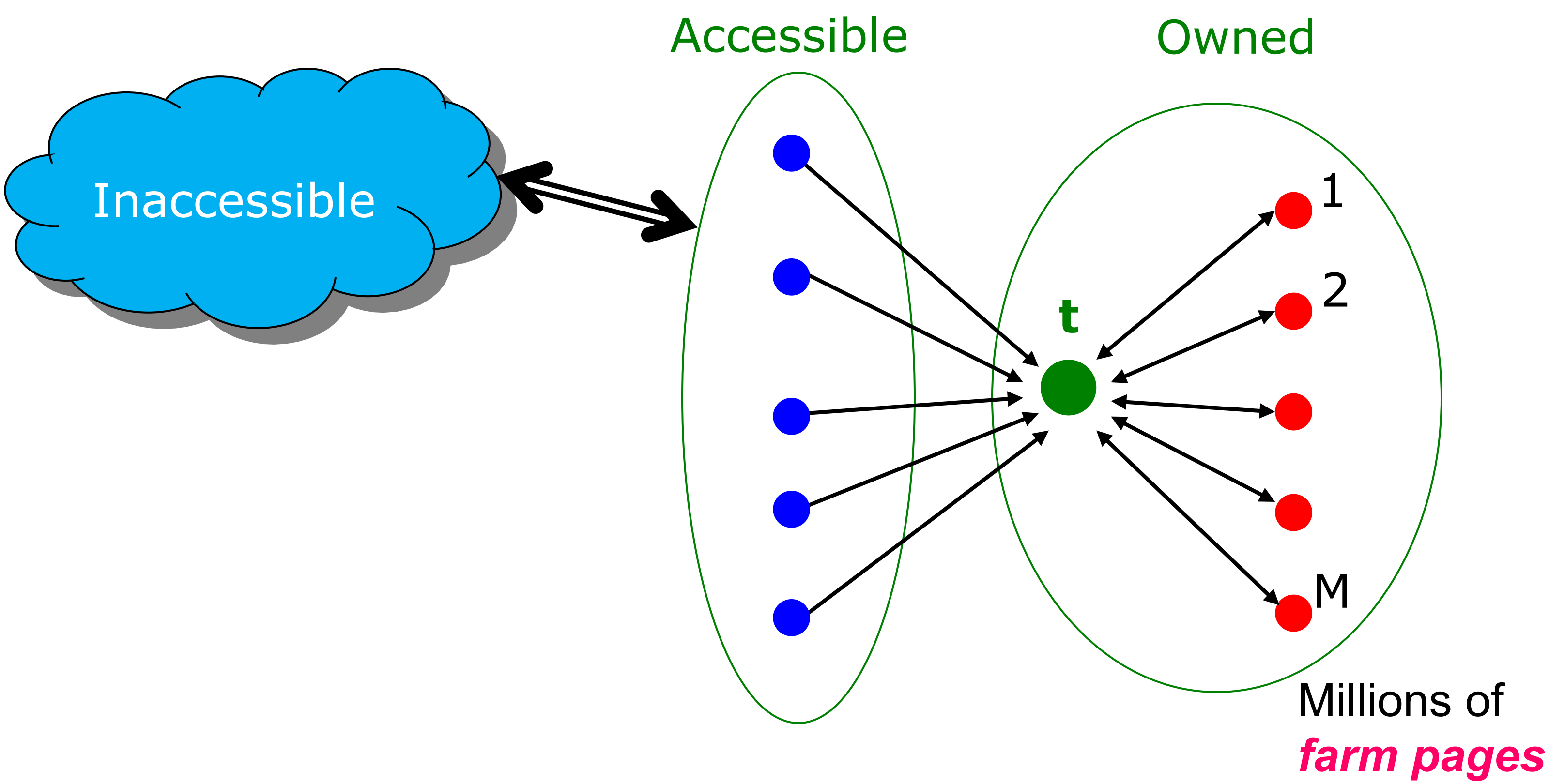
- **Spam farms** were developed to concentrate PageRank on a single page

- **Link spam:**
  - Creating link structures that boost PageRank of a particular page

Link Spamming

- **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
  - **Owned pages**
    - Completely controlled by spammer
    - May span multiple domain names

Link Farms



Get as many links from accessible pages as possible to target page t

TrustRank: Combating the Web Spam

Combating Spam

- **Combating link spam**
  - **Detection and blacklisting of structures that look like spam farms**
    - Leads to another war – hiding and detecting spam farms

  - **TrustRank**

  - **Spam Mass**

TrustRank: Idea

- **Basic principle: Approximate isolation**
  - It is rare for a "good" page to point to a "bad" (spam) page
- Choose a set of seed pages that are identified as **good** the **trusted pages**
- Perform a topic-sensitive PageRank with **teleport set = trusted pages**
  - **Propagate trust through links**

Picking the Seed Set

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

Approaches to Picking Seed Set

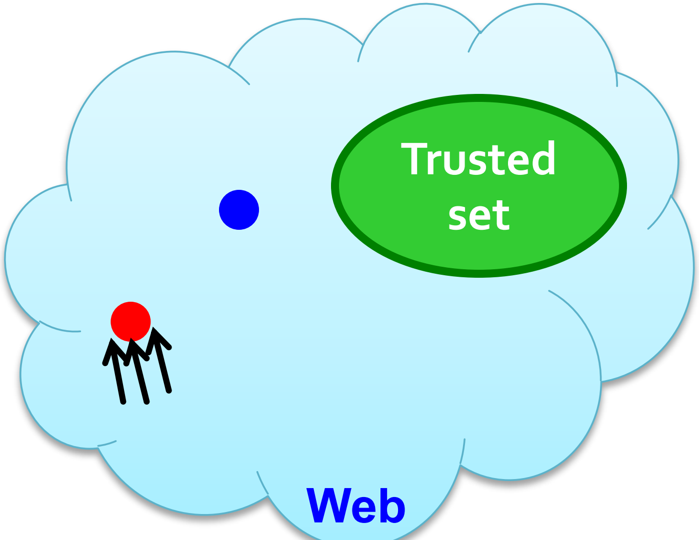
- Suppose we want to pick a seed set of ***k*** pages
- **How to do that?**
- **(1) PageRank:**
  - Pick the top ***k*** pages by PageRank
  - Theory is that you can't get a bad page's rank really high
- **(2) Use trusted domains** whose membership is controlled, like .edu, .mil, .gov

Spam Mass

- r_p = PageRank of page ***p***
- r_p^+ = PageRank of ***p*** with teleport into **trusted** pages only

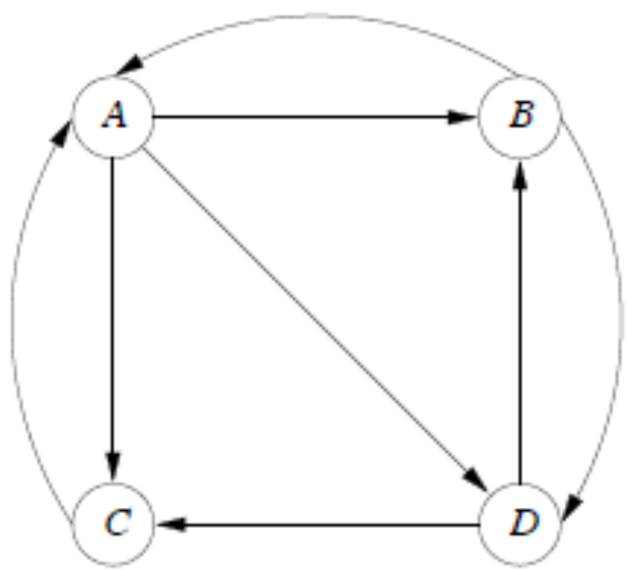
- **Then:** What fraction of a page's PageRank comes from **spam** pages?

- **Spam mass of *p* =** $\frac{r_p - r_p^+}{r_p}$
  - Pages with high spam mass are spam.



Quiz: Page Rank

- Given an example of the Web
  - Compute transition matrix
  - What is the meaning of the transition matrix?
  - Compute Page Rank.



Quiz: Page Rank

- Answer
  - The transition matrix for the presented Web is

$$M = \begin{bmatrix} 0 & 1/2 & 1 & 0 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 0 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

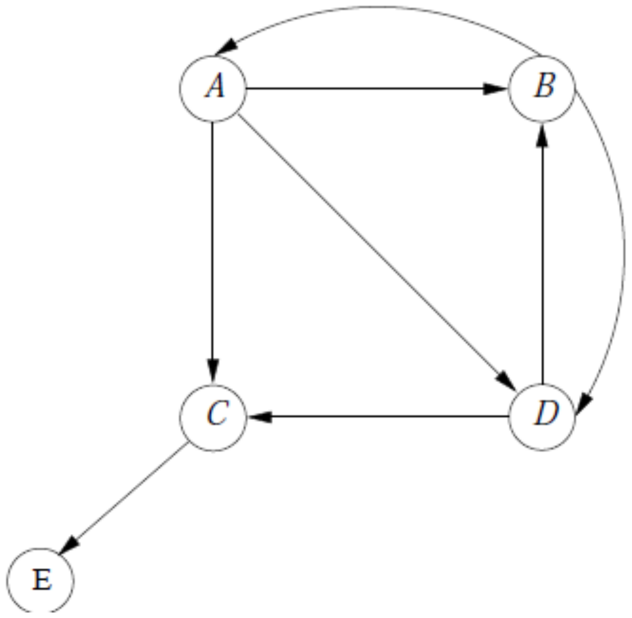
Quiz: Page Rank

- The sequence of approximations to the limit that we get by multiplying at each step by M is:

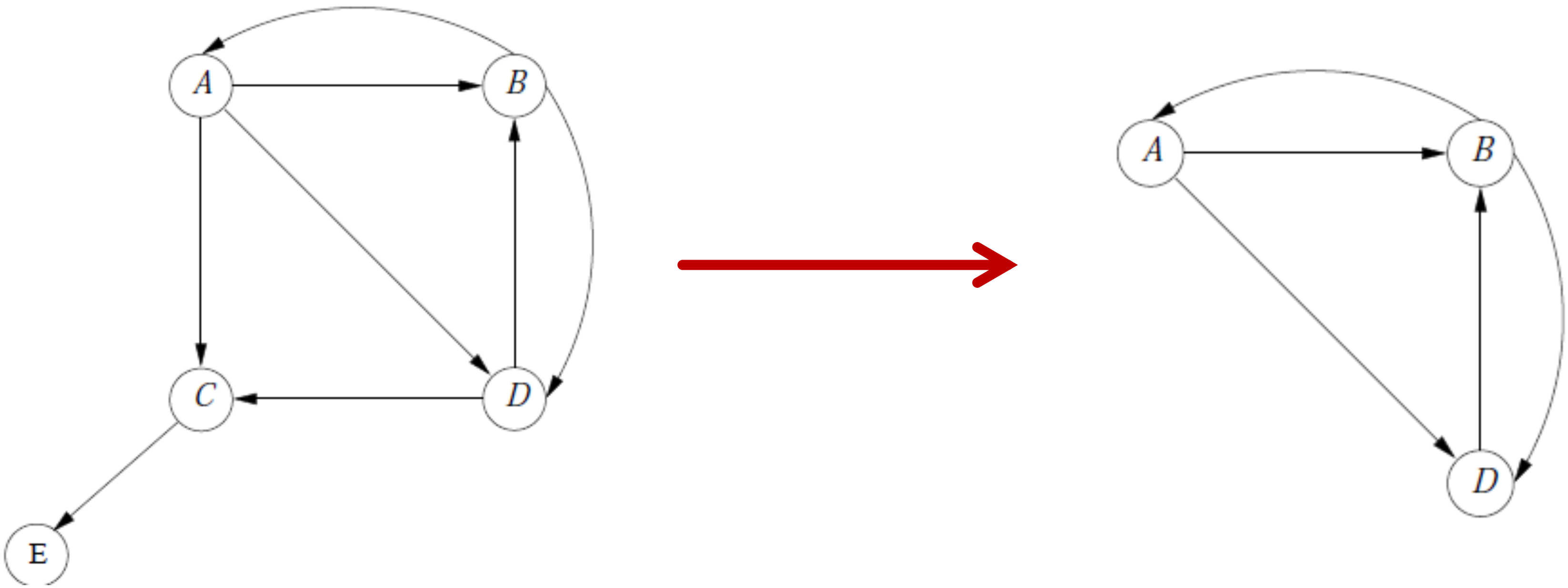
$$\begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}, \begin{bmatrix} 9/24 \\ 5/24 \\ 5/24 \\ 5/24 \end{bmatrix}, \begin{bmatrix} 15/48 \\ 11/48 \\ 11/48 \\ 11/48 \end{bmatrix}, \begin{bmatrix} 11/32 \\ 7/32 \\ 7/32 \\ 7/32 \end{bmatrix}, \dots, \begin{bmatrix} 3/9 \\ 2/9 \\ 2/9 \\ 2/9 \end{bmatrix}$$

Quiz: Dead Ends

- One way to deal with dead ends is to drop the dead ends from the graph, and also drop their incoming arcs. Doing so may create more dead ends, which have to be dropped, recursively.
- Apply this method on the dead ends of the following graph and draw the new graph.
- Compute the transition matrix of the new graph.



Quiz: Dead Ends



$$\begin{bmatrix} 0 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 1/2 & 1/2 & 0 \end{bmatrix}$$

Summary

- PageRank is an algorithm that assigns a real number, called its PageRank, to each page on the Web.
- The PageRank of a page is a measure of how important the page is, or how likely it is to be a good response to any search query.

Summary

- Dead Ends: A dead end is a Web page with no links out. The presence of dead ends will cause the PageRank of some or all of the pages to go to 0 in the iterative computation

Summary

- Spider Traps: A spider trap is a set of nodes that, while they may link to each other, have no links out to other nodes.
- To counter the effect of spider traps (and of dead ends, if we do not eliminate them), PageRank is normally computed in a way that modifies the simple iterative multiplication by the transition matrix (teleport)

Summary

- If we know the queryer is interested in a certain topic, then it makes sense to bias the PageRank in favor of pages on that topic.

- TrustRank: One way to ameliorate the effect of link spam is to compute a topic-sensitive PageRank called TrustRank, where the teleport set is a collection of trusted pages

Actions

- Review slides!
- Read Chapter 5 (Link Analysis) from course book.
  - You can find electronic version of the book on Blackboard.