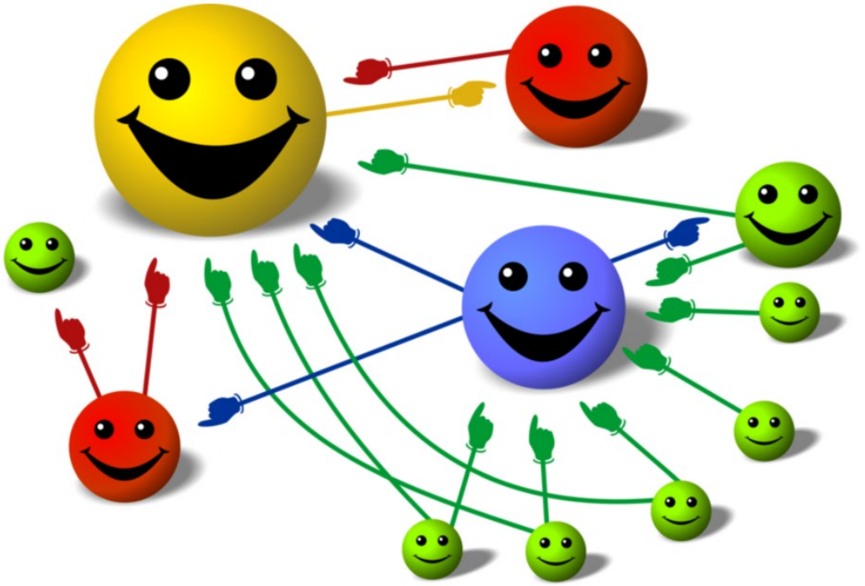
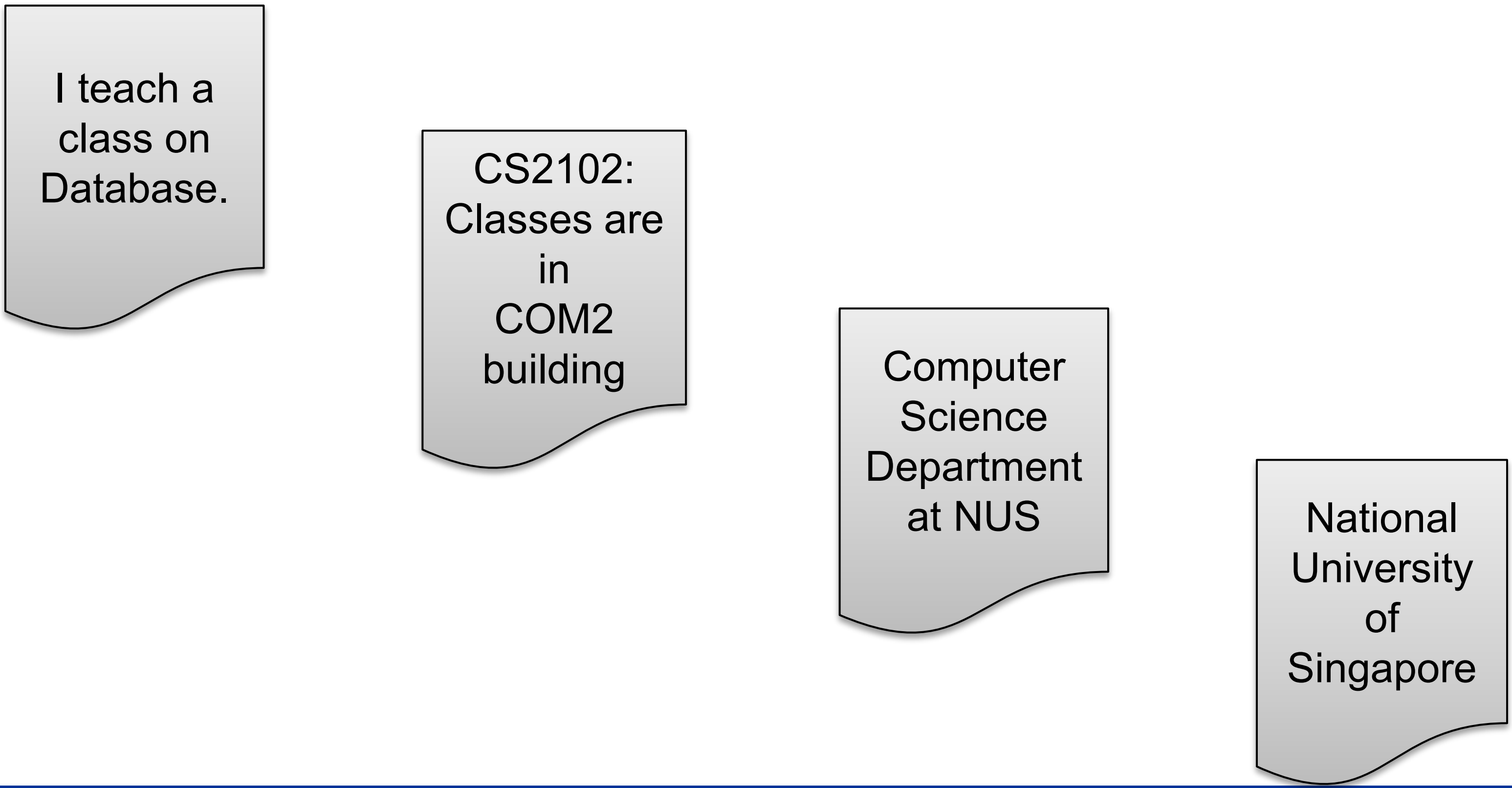
CS5344
Link Analysis



Acknowledgement: Slides adapted from http://www.mmds.org

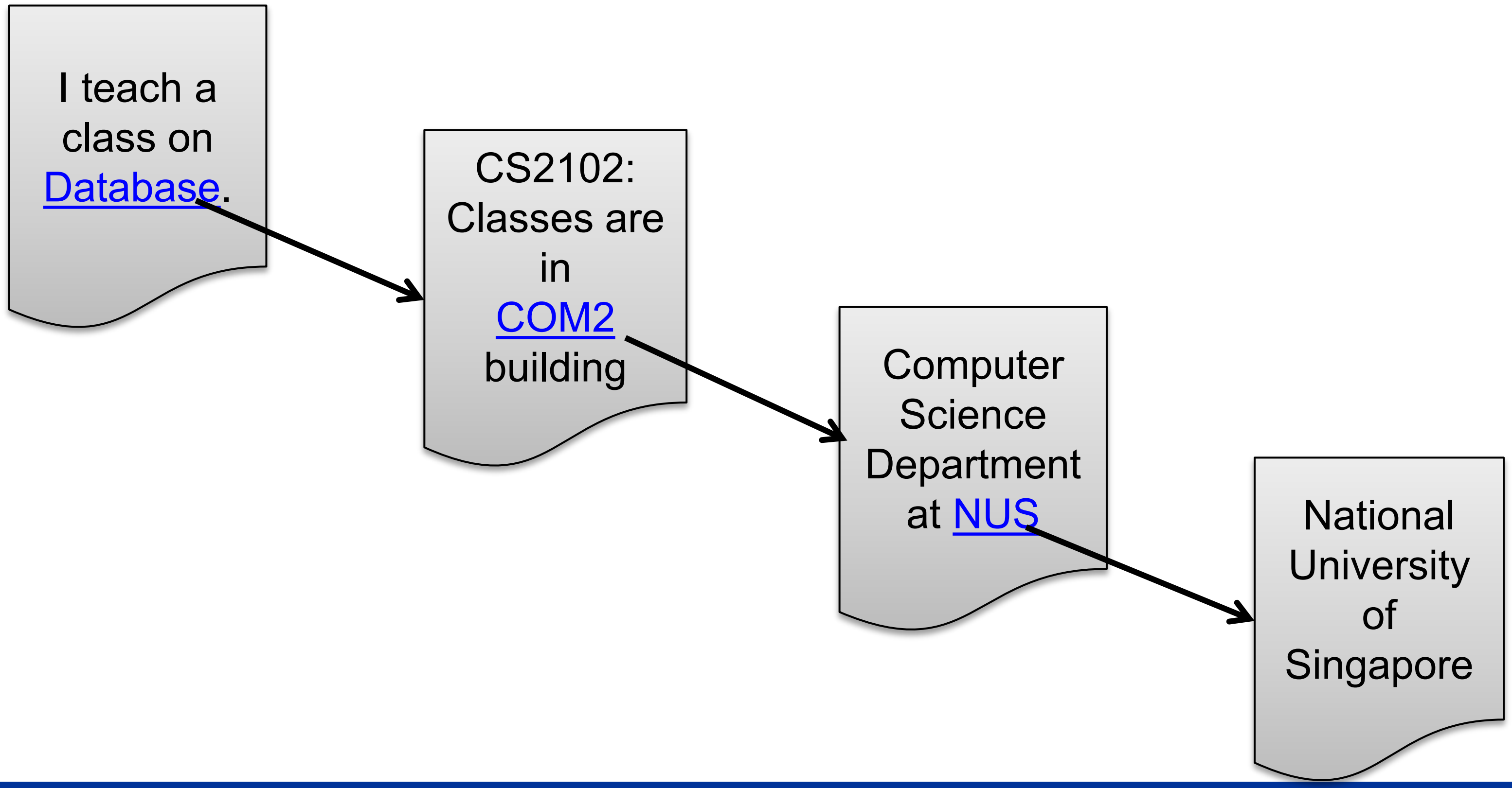
Web as a Graph

- **Nodes: Webpages**
- **Edges: Hyperlinks**

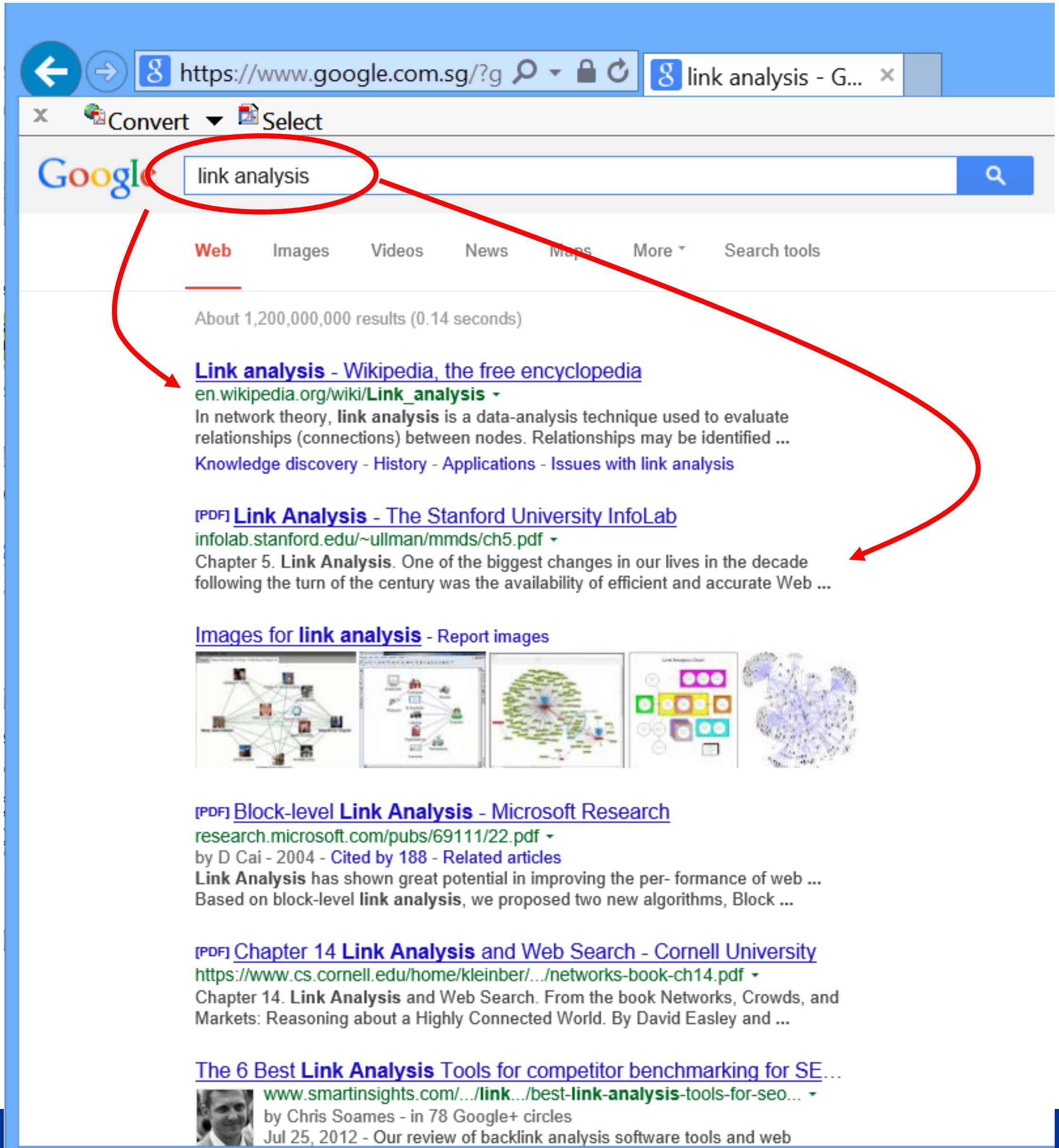


Web as a Graph

- **Nodes: Webpages**
- **Edges: Hyperlinks**



Web Search



- How does the search engine decide which page should be ranked higher?

Web Search - Challenges

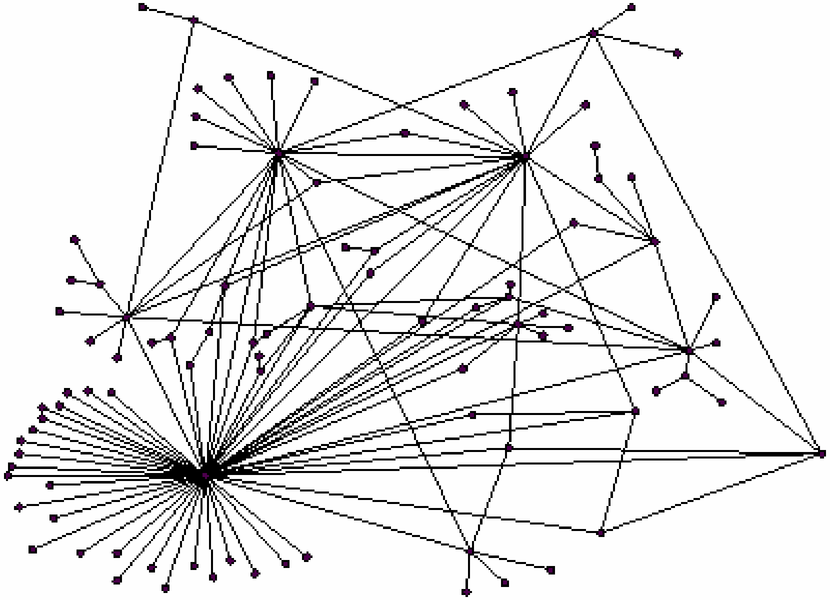
- Web contains many sources of information.
  - Who to “trust”?
- What is the “best” answer to query “newspaper”?
  - No single right answer

Link Analysis

- **The Web is not just a collection of documents**
  - The hyperlinks are important
- **A link from page *A* to page *B* may indicate**
  - *A* is related to *B*, or
  - *A* is recommending, citing, voting for, or endorsing *B*
- **Types of links:**
  - Referential – *click here and get back home*
  - Informational – *click here to get more detail*
- **Links influence the ranking of web pages and thus have commercial value**

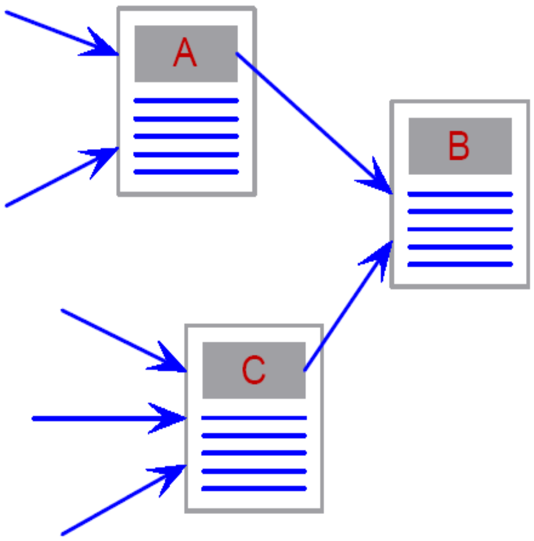
Importance of Web Pages

- Not all web pages are equally important
- A page is important if it is pointed to by other important pages (recursion)

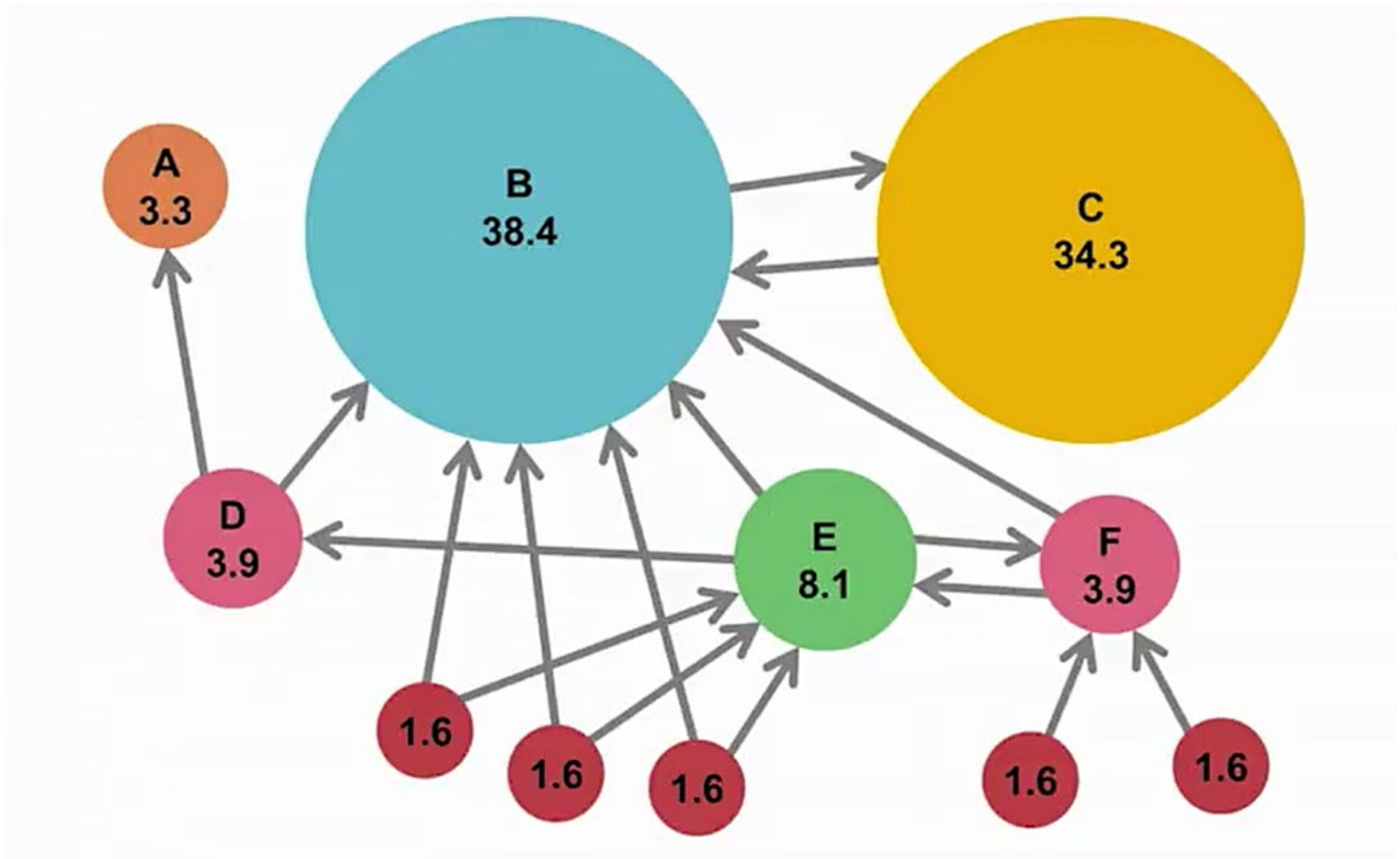


PageRank

- **Idea: Links as votes**
  - A page is more important if it has more links
- **Incoming links to a page is a measure of importance and authority of the page**
  - www.stanford.edu has 23,400 in-links
  - www.joe-schmoe.com has 1 in-link
- **Are all incoming links equal?**
  - Links from important pages count more



Example PageRank Scores

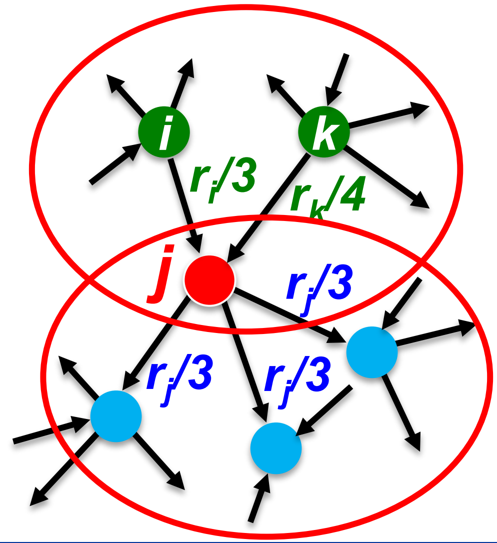


- A "vote" from an important page is worth more
- A page is important if it is pointed to by other important pages

Recursive Formulation

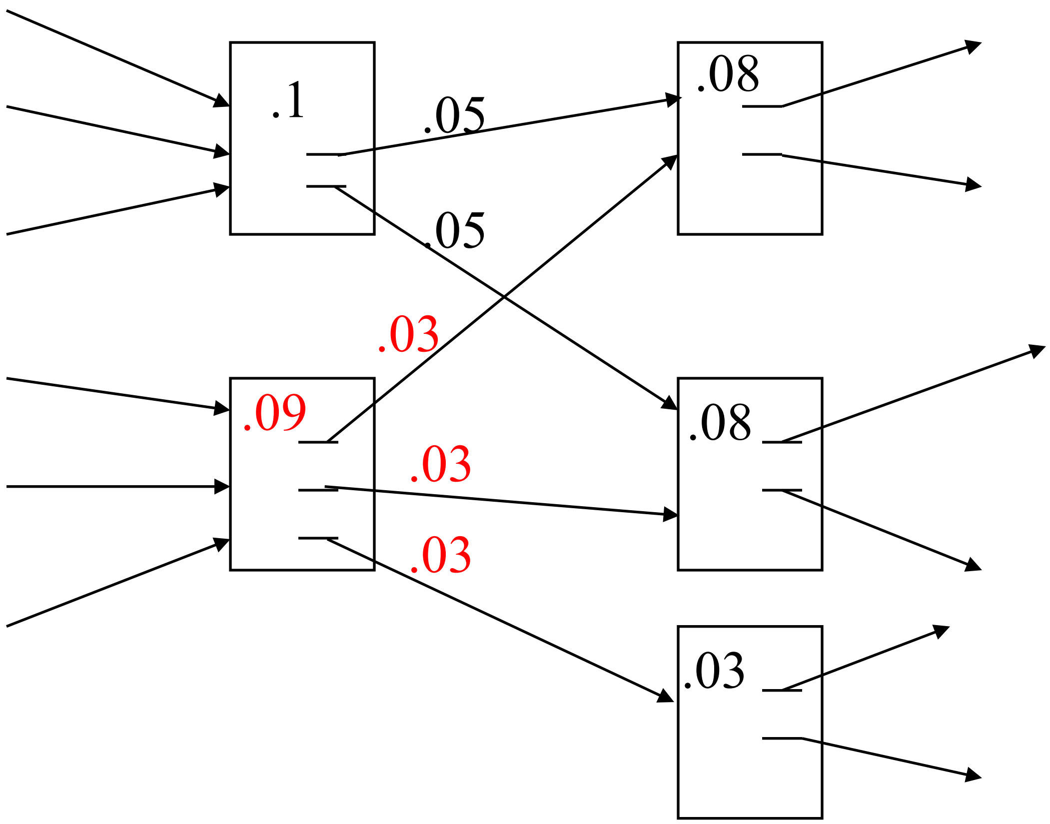
- A link's vote is proportional to the importance of its source page
- If page j with importance r_j has n out-links, each link gets r_j / n votes
- Page j's own importance is the sum of the votes on its in-links

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



Flow Model

- **Can view it as a process of PageRank "flowing" from pages to the pages they point to**



Flow Model

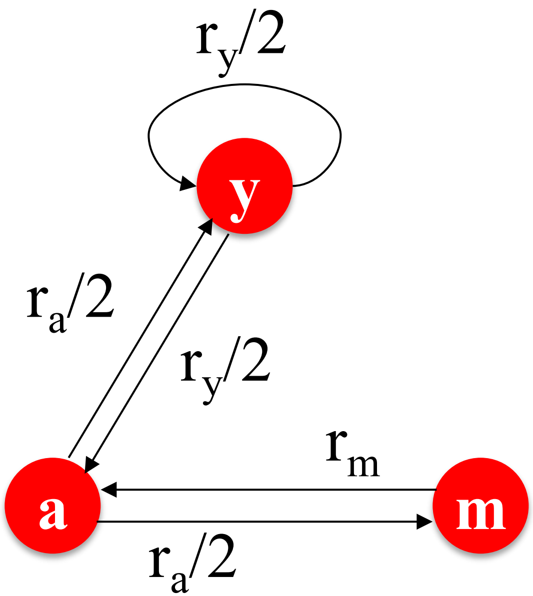
- **Define a rank r_j for page j**

$$r_j = \sum_{i \to j} \frac{r_i}{\mathrm{d}_i}$$

d_i is the out-degree of node i

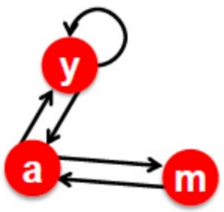
- **Flow Equations**

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



Matrix Formulation

- **Stochastic adjacency matrix *M***
  - Let page *i* has d_i outlinks
  - If $i \rightarrow j$, then $M_{ji} = 1/d_i$ else $M_{ji} = 0$
  - ***M*** is a column stochastic matrix
    - **Columns sum to 1**
- **Rank vector *r***
  - Vector with one entry per page
  - r_i is the importance score of page *i*
- **Flow equations can be written in matrix form *r* = *M* · *r***

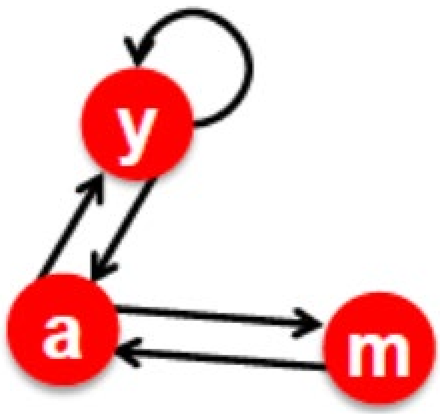


	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

$$\sum_i r_i = 1$$

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

Example Flow Equations and M



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

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

$$\equiv$$

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

Power Iteration Method

- **Given a web graph with *n* nodes, where the nodes are pages and edges are hyperlinks**
- **Power iteration – simple iterative scheme**
  - Suppose there are N web pages
  - Initialize $\mathbf{r}^{(0)} = [\ 1/N, \ldots., 1/N\ ]^T$
  - Iterate: $\mathbf{r}^{(t+1)} = \mathbf{M} \cdot \mathbf{r}^{(t)}$
- **Stop when $|\mathbf{r}^{(t+1)} - \mathbf{r}^{(t)}|_1 < \varepsilon$**
  - $|\mathbf{x}|_1 = \sum_{i \in [1,N]} |x_i|$ is the L_1 norm
  - Can use any other vector norm e.g., Euclidean

$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d}_i}$$

d_i …. out-degree of node i

Example

- **Power Iteration:**

  Set $r_j = 1/N$

  1: $r'_j = \sum_{i \to j} \frac{r_i}{d_i}$

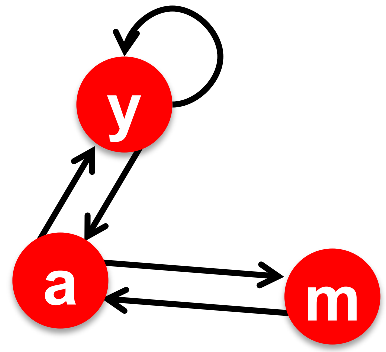
  2: $r = r'$

  Goto 1

- **Example:**

$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{matrix} 1/3 \\ 1/3 \\ 1/3 \end{matrix}$$

Iteration 0, 1, 2, …



	y	a	m
y	½	½	0
a	½	0	1
m	0	½	0

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

PageRank

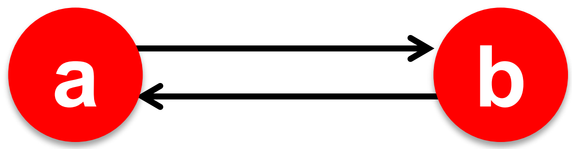
$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d}_\mathrm{i}}$$

or equivalently

$$r = Mr$$

- **Questions:**
  1. Does this converge?
  2. Does it converge to what we want?

Does this converge?

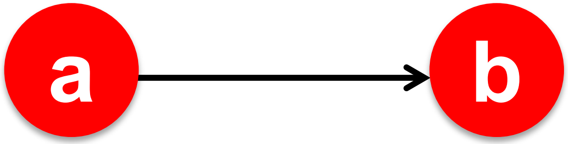


$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

$$\begin{matrix} r_a \\ r_b \end{matrix} = \begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{matrix}$$

Iteration 0, 1, 2, …

Does it converge to what we want?



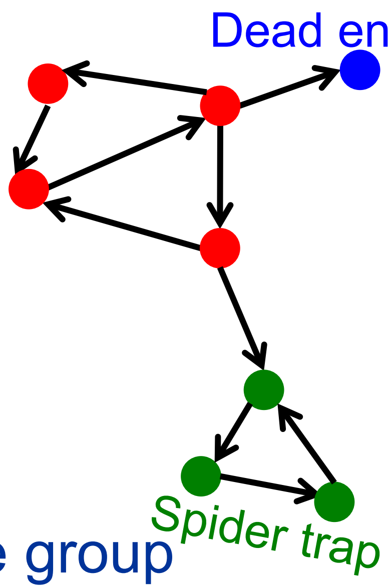
$$r_j^{(t+1)} = \sum_{i \to j} \frac{r_i^{(t)}}{\mathrm{d_i}}$$

$$\begin{matrix} \mathbf{r_a} \\ \mathbf{r_b} \end{matrix} = \begin{matrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$

Iteration 0, 1, 2, …

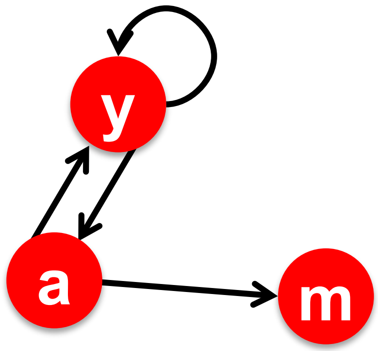
Problems on Real Web

- **Imagine a random web surfer**
  - At any time *t*, surfer is on some page *i*
  - At time *t+1*, surfer follows an out-link from *i* uniformly at random
  - Ends up on some page *j* linked from *i*
- **Dead ends**
  - A page has no out-links
  - Random walk has "nowhere" to go to
  - Such pages cause importance to "leak out"
- **Spider traps**
  - A group of pages have no out-links out of the group
  - Random walk gets "stuck" in a trap
  - Eventually spider traps absorb all importance



Problem: Dead Ends

- A page with no out-links
- Random walk has "nowhere" to go to
- All importance "leaks out of" the Web!
- Matrix is not stochastic so initial assumptions are not met



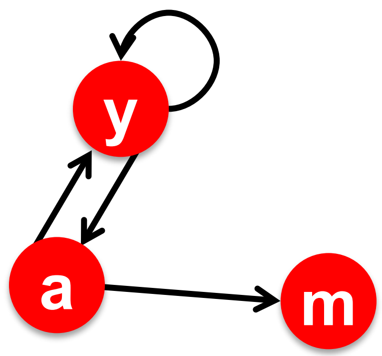
	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

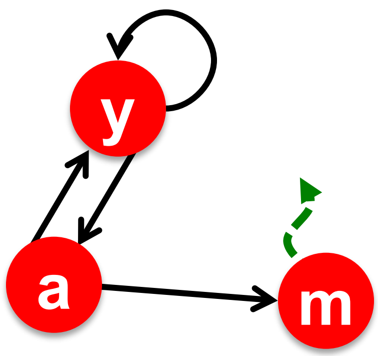
$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/6 \\ 1/6 \end{bmatrix} \begin{bmatrix} 1/4 \\ 1/6 \\ 1/12 \end{bmatrix} \begin{bmatrix} 5/24 \\ 1/8 \\ 1/12 \end{bmatrix} \cdots\cdots \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Iteration 0, 1, 2, …

Solution: Teleport

- Adjust the matrix to allow a surfer to jump to some random page from dead ends

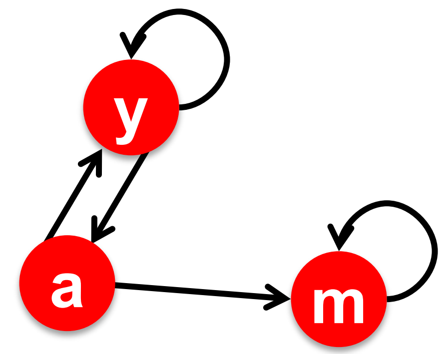




	y	a	m
y	½	½	0
a	½	0	0
m	0	½	0

	y	a	m
y	½	½	⅓
a	½	0	⅓
m	0	½	⅓

Problem: Spider Traps



- A group of pages with no links out of the group
- Random walk gets "stuck" in a trap
- Accumulate all the importance of the Web

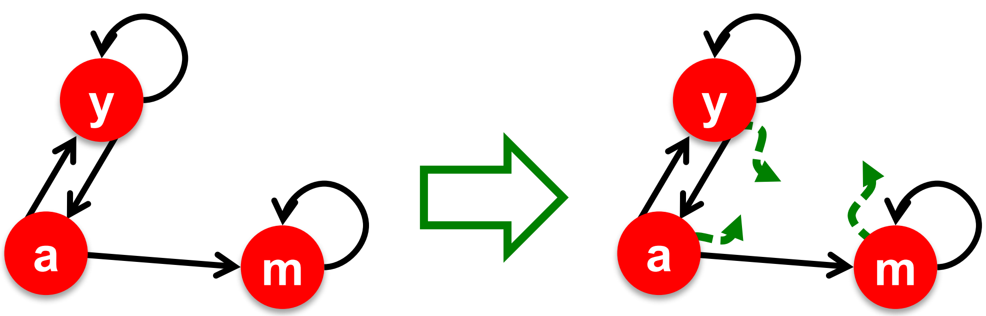
	y	a	m
y	½	½	0
a	½	0	0
m	0	½	1

All the PageRank score gets "trapped" in node m

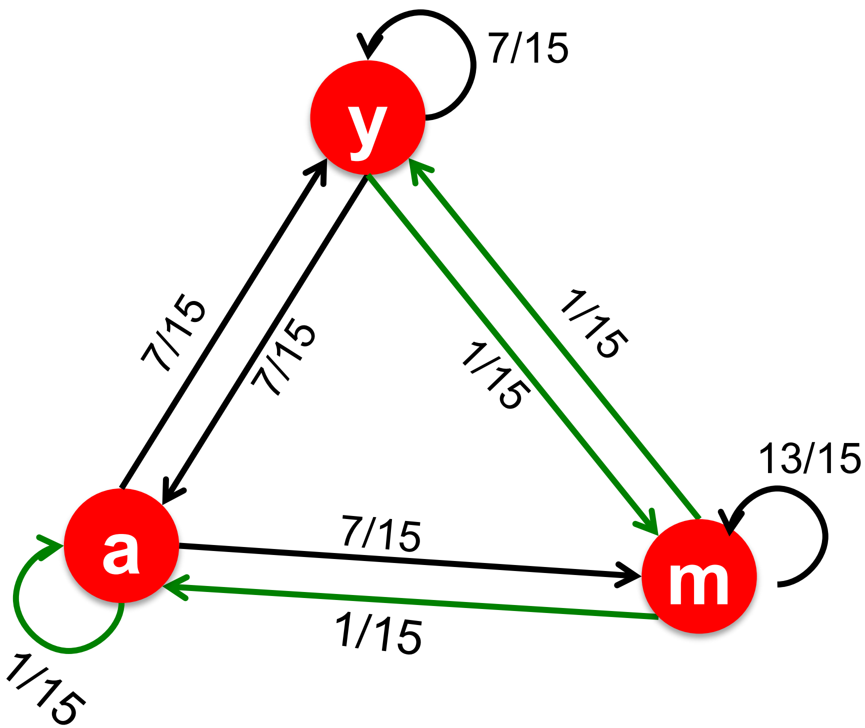
$$\begin{bmatrix} r_y \\ r_a \\ r_m \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix} \begin{bmatrix} 1/4 \\ 1/6 \\ 7/12 \end{bmatrix} \begin{bmatrix} 5/24 \\ 1/8 \\ 2/3 \end{bmatrix} \quad \cdots\cdots \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Solution: Teleport

- **At each time step, a 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**



Random Teleports ($\beta = 0.8$)



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

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

$$\begin{matrix} \mathbf{r}_y \\ \mathbf{r}_a \\ \mathbf{r}_m \end{matrix} = \begin{matrix} 1/3 \\ 1/3 \\ 1/3 \end{matrix} \quad \begin{matrix} 0.33 \\ 0.20 \\ 0.46 \end{matrix} \quad \begin{matrix} 0.24 \\ 0.20 \\ 0.52 \end{matrix} \quad \begin{matrix} 0.26 \\ 0.18 \\ 0.56 \end{matrix} \quad \cdots \quad \begin{matrix} 7/33 \\ 5/33 \\ 21/33 \end{matrix}$$

Limitations of PageRank

- **Measures generic popularity of a page**
  - Ignore or miss topic-specific authorities
  - **Solution:** Topic-specific PageRank
- **Susceptible to link spam**
  - Artificial link topologies created in order to boost page rank
  - **Solution:** TrustRank
- **Uses a single measure of importance**
  - Other models of importance
  - **Solution:** Hubs-and-Authorities

Topic-Specific PageRank

- **Instead of generic popularity, can we measure popularity within a topic?**
- **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:** Is "Jaguar" an animal, the automobile, or a version of MAC OS?

Topic-Specific PageRank

- **Recall random walker has a small probability of teleporting at any step**
  - Standard PageRank: Any page with equal probability
  - Topic Specific PageRank: Teleport set is restricted to a topic-specific set of “relevant” pages
- **Idea: Bias the random walk**
  - When random walker teleports, pick a page from a set $\boldsymbol{S}$ of web pages
  - $\boldsymbol{S}$ contains only pages that are relevant to the topic
  - Get a different rank vector $\boldsymbol{r_S}$ for each teleport set $\boldsymbol{S}$

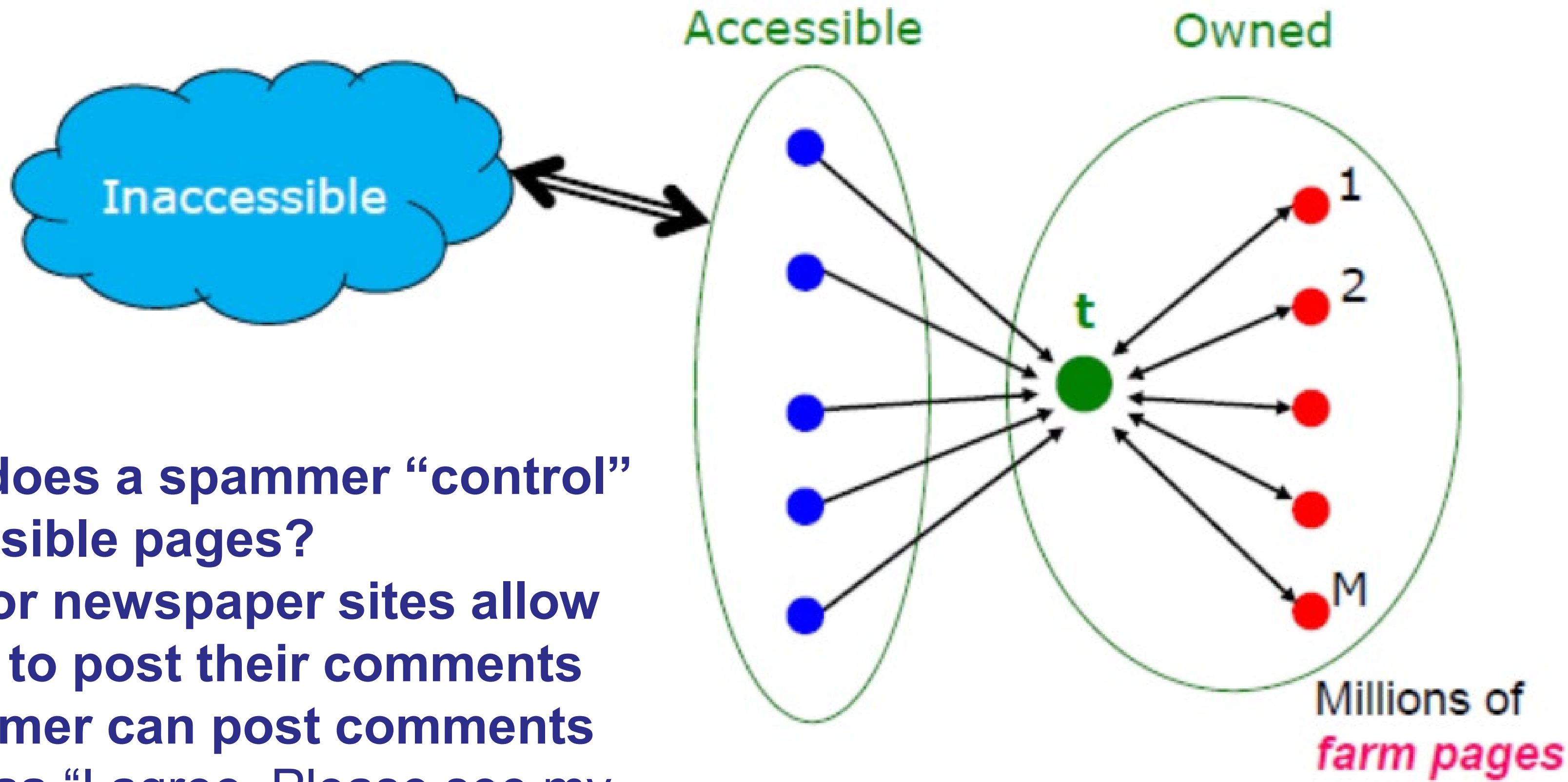
Topic-Specific PageRank

- **Decide on topics to create PageRank vectors**
  - Open Directory (DMOZ) (www.dmoz.org)
  - The 16 DMOZ top-level categories: arts, business, sports, …
- **Pick a teleport set for each of these topics, and compute the topic-sensitive PageRank vector for that topic**
- **Determine the topic that is most relevant for a query**
  - User picks from a menu
  - Query context e.g., query from a web page on a known topic
  - User context e.g., user's bookmarks
- **Use the PageRank vectors for that topic to order results to the search query**

TrustRank – Combating Web Spam

- **Spamming**
  - Any deliberate action to boost a web page's position in search engine results, incommensurate with page's real value
  - Approximately **10-15%** of web pages are spam
- **Link Spam**
  - Create link structures that boost the PageRank of a particular page

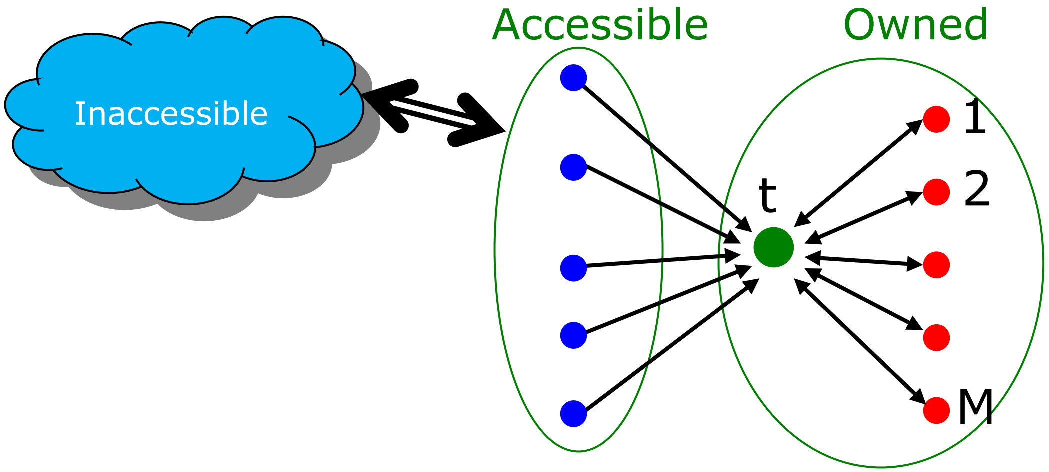
Spammer's View of the Web



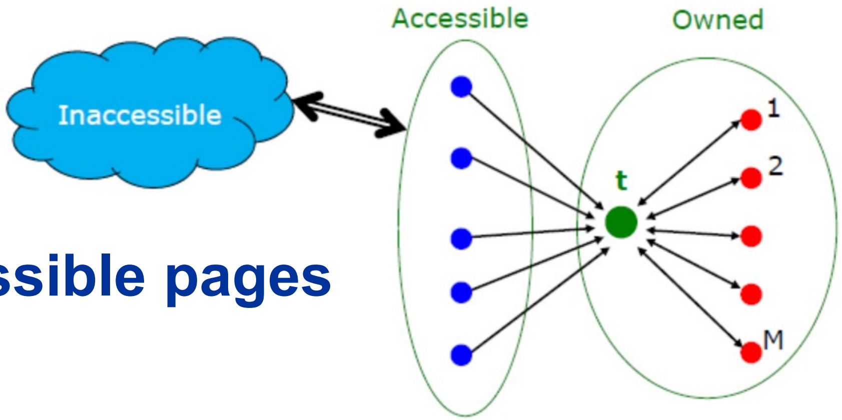
- **How does a spammer "control" accessible pages?**
- **Blog or newspaper sites allow users to post their comments**
- **Spammer can post comments such as** "I agree. Please see my article at www.mySpamFarm.com"

Link Farms

- **Spammer's goal is to maximize the PageRank of target page *t***
- **Get as many links from accessible pages as possible to target page *t***
- **Construct "link farm" to get PageRank multiplier effect**



Analysis



x: PageRank contributed by accessible pages

y: PageRank of target page t

N: Total number of web pages

Rank of each "farm" page $= \frac{\beta y}{M} + \frac{1-\beta}{N}$

$$y = x + \beta M \left[\frac{\beta y}{M} + \frac{1-\beta}{N}\right] + \frac{1-\beta}{N}$$

$$y = \frac{x}{1-\beta^2} + c\frac{M}{N} \quad \text{where } c = \frac{\beta}{1+\beta}$$

Let β = 0.85. Then $1/(1 - \beta^2)$ = 3.6, and c = 0.46

- External PageRank (x) increased by 360%!
- Obtain additional amount of PageRank that is 46% of the fraction of the Web, M/N, that is in the spam farm
- By making M large, we can make y as large as we want

How to Combat Link Spam?

- **Detect and blacklist structures that look like spam farms**
  - One page links to a very large number of pages, each of which links back to it
  - Leads to more sophisticated way of hiding spam farms, and detecting them…
- **TrustRank: Topic-specific PageRank with a teleport set of trusted pages**
  - e.g, **.edu** domains, **.gov** domains, etc
  - Lower the score of spam pages

TrustRank

- **Basic principle: Approximate isolation**
  - It is rare for a “good” (trustworthy) page to point to a “bad” (spam) page
- **Sample a set of seed pages from the web**
- **An oracle (human) identifies the good pages and the spam pages in the seed set**
  - Expensive task, so keep seed set small
  - Subset of pages in the seed set that are identified as good are called the **trusted pages**

Trust Propagation

- **Perform a topic-sensitive PageRank with the trusted pages as the teleport set**
- **Propagate trust through links**
  - Each page gets a trust value between 0 and 1
- **Use a threshold value and mark all pages below the trust threshold as spam**

Trust Propagation (Simple Model)

- Set trust of each trusted page to 1
- Suppose trust of page p is t_p
  - p has a set of out-links o_p
- For each $q \in o_p$, p confers the trust to q
  - $\beta t_p / |o_p|$ for $0 < \beta < 1$
- Trust is additive
  - Trust of p is the sum of the trust conferred on p by all its in-linked pages
- Trust attenuation
  - Degree of trust conferred by a trusted page decreases with the distance in the graph
- Trust splitting
  - The larger the number of out-links, the less scrutiny the page author gives each out-link; trust is split across out-links

Picking the Seed Set

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

1. Use PageRank to pick the top-k pages

- Theory is a bad page cannot have very high rank

2. Use trusted domains with controlled membership

- E.g. university pages (.edu)
  or government pages (.gov)

Summary

- Link analysis in social network graphs to find communities
- Girvan-Newman algorithm use edge betweenness measure to separate nodes into communities

- Content of web pages and hyperlinks are important in web search
- Page Rank algorithm determine importance of web pages
- Trust Rank algorithm to overcome link spams