

# How Does a Search Engine Work? Part 2

Frank McCown  
Intro to Web Science  
Harding University  
Spring 2011

# Link Analysis

- Content analysis is useful, but combining with link analysis allow us to rank pages much more successfully
- 2 popular methods
  - Sergey Brin and Larry Page's PageRank
  - Jon Kleinberg's Hyperlink-Induced Topic Search (HITS)

# What Does a Link Mean?



- A recommends B
- A specifically does **not** recommend B
- B is an authoritative reference for something in A
- A & B are about the same thing (topic locality)

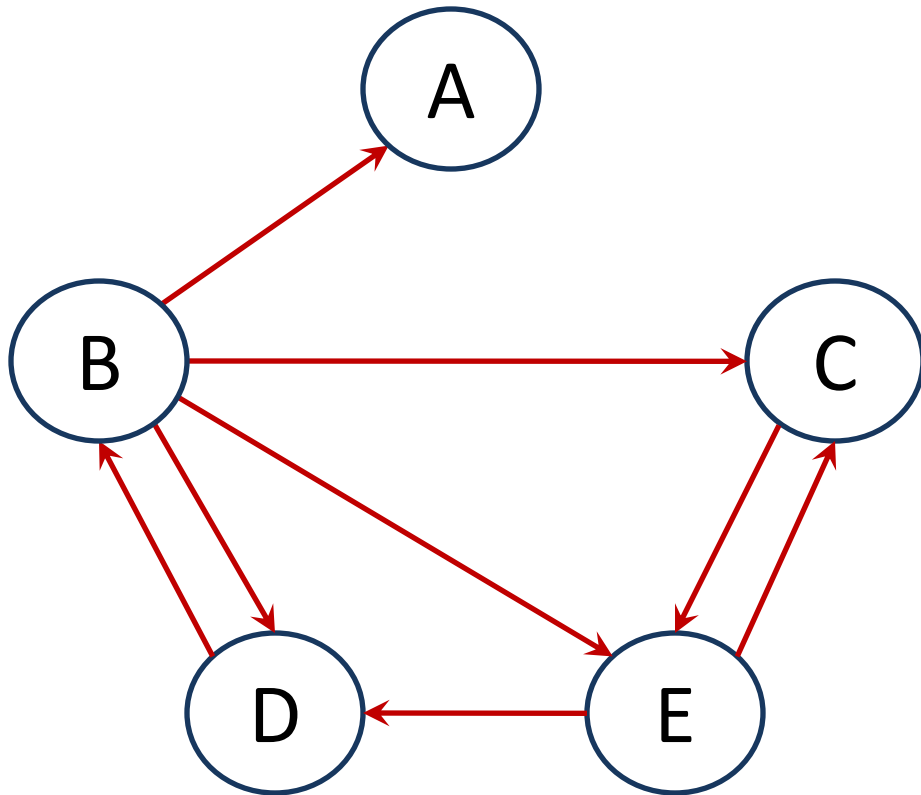
# PageRank

- Developed by Brin and Page (Google) while Ph.D. students at Stanford
- Links are a recommendation system
  - The more links that point to you, the more important you are
  - Inlinks from important pages are weightier than inlinks from unimportant pages
  - The more outlinks you have, the less weight your links carry

# Random Surfer Model

- Model helpful for understanding PageRank
- The Random Surfer starts at a randomly chosen page and selects a link at random to follow
- PageRank of a page reflects the probability that the surfer lands on that page

# Example of Random Surfer



Start at: B

$\frac{1}{4}$  probability of going to A

$\frac{1}{4}$  probability of going to C

$\frac{1}{4}$  probability of going to D

$\frac{1}{4}$  probability of going to E

Choose: E

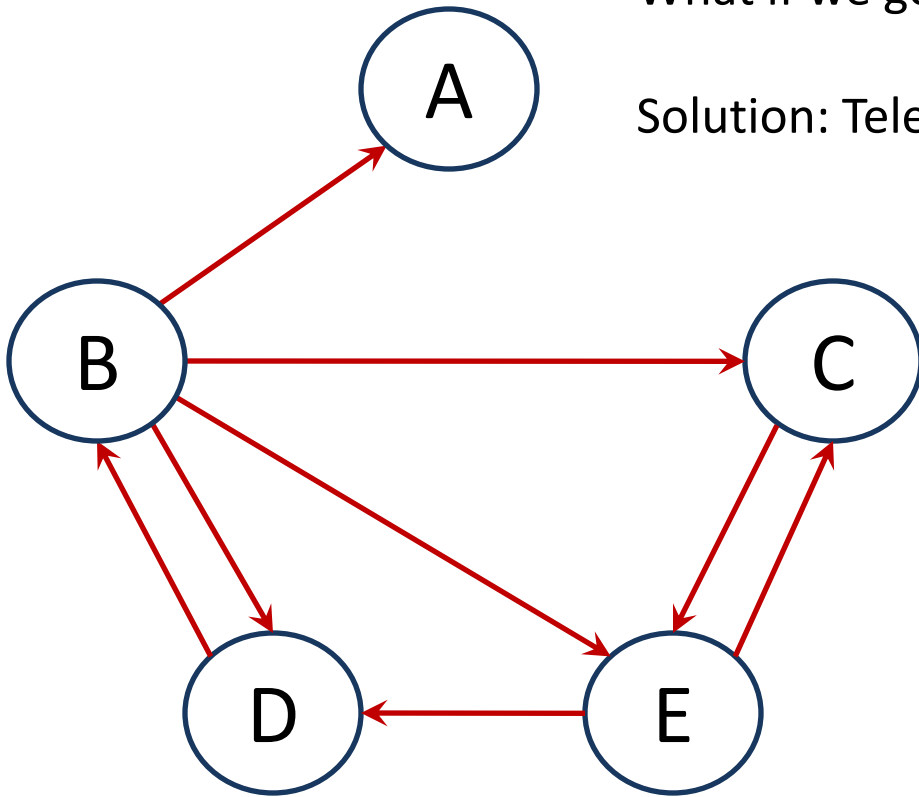
$\frac{1}{2}$  probability of going to C

$\frac{1}{2}$  probability of going to D

# Problem 1: Dangling Node

What if we go to A? We're stuck at a dead-end!

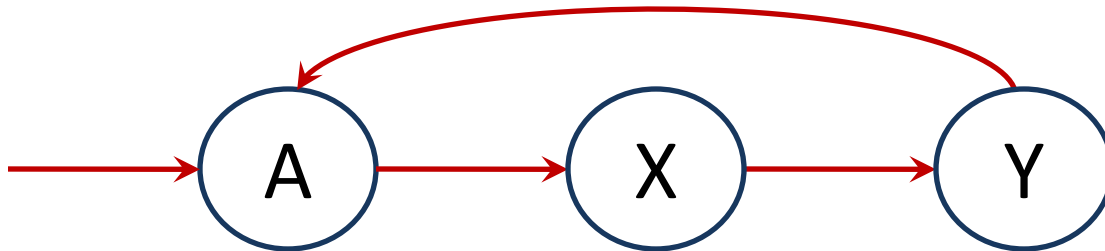
Solution: Teleport to any other page at random



# Problem 2: Infinite Loop

What if we get stuck in an a **cycle**?

Solution: Teleport to any other page at random



# Rank Sinks

- Dangling nodes and cycles are called **rank sinks**
- Solution is to add a **teleportation probability  $\alpha$**  to every decision
- $\alpha\%$  chance of getting bored and jumping somewhere else,  $(1 - \alpha)\%$  chance of choosing one of the available links
- $\alpha = .15$  is typical

# PageRank Definition

Teleportation probability

Sum of PR of all pages pointing to  $P_i$

$$PR(P_i) = \frac{\alpha}{|P|} + (1 - \alpha) \cdot \sum_{P_j \in B_{P_i}} \frac{PR(P_j)}{|P_j|}$$

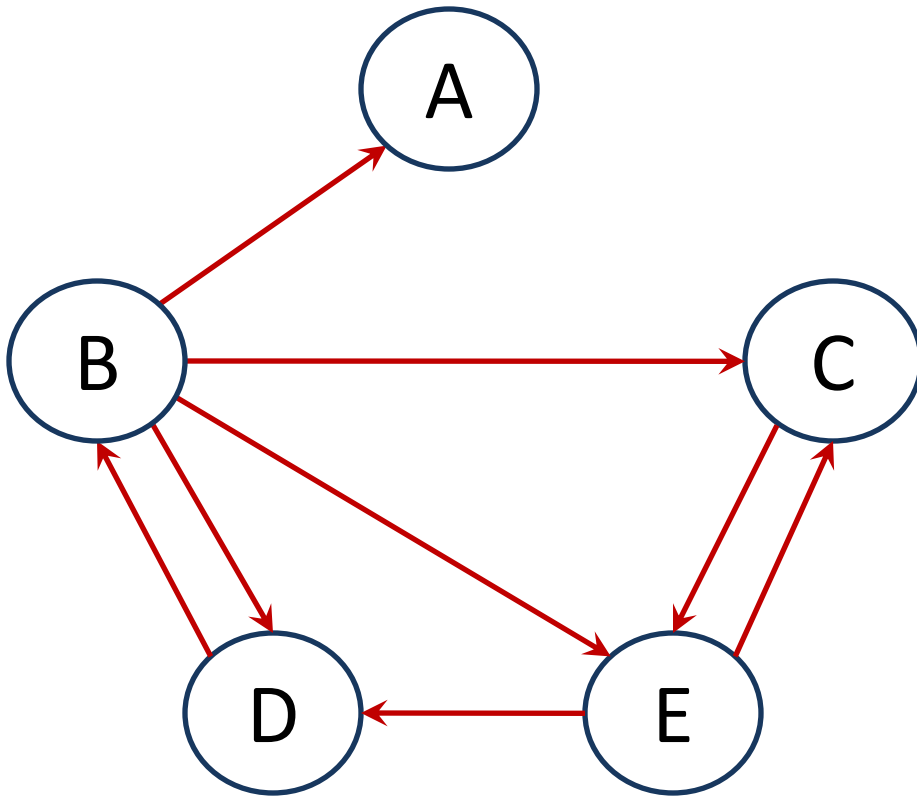
PageRank of page  $P_i$

Total num of pages

$B_{P_i}$  is set of all pages pointing to  $P_i$

Number of outlinks from  $P_j$

# PageRank Example

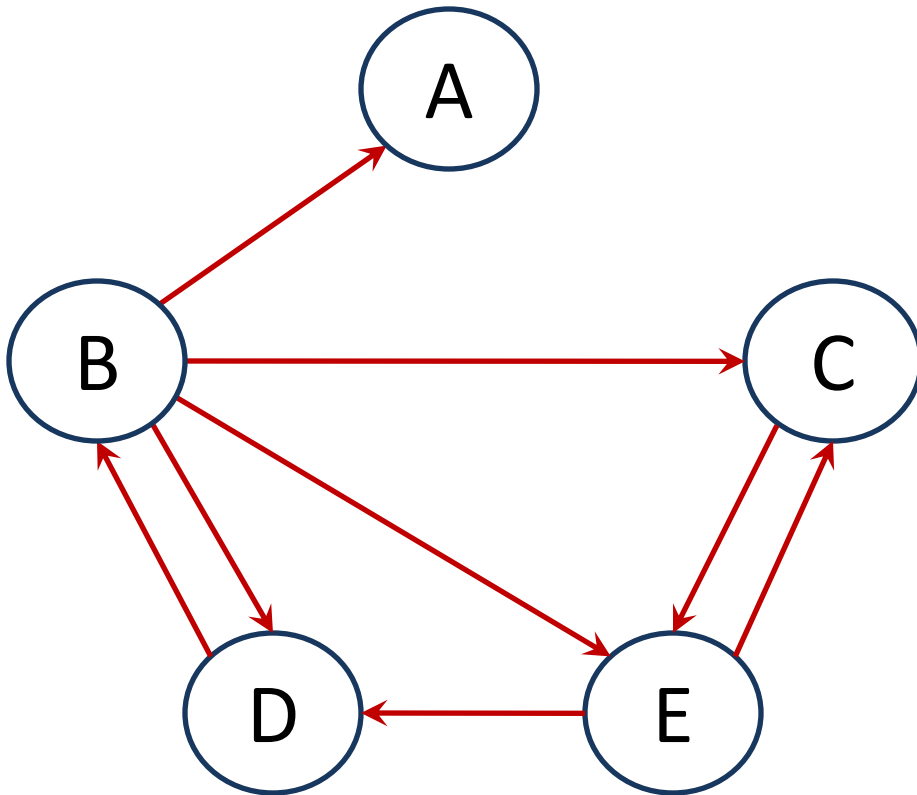


$$PR(C) = .15/5 + .85 \times (PR(B)/4 + PR(E)/2)$$

Problem: What is  $PR(B)$  and  $PR(E)$ ?

Solution: Give all pages same PR to start ( $1/|P|$ ) & iteratively calculate new PR

# PageRank Example



$$\begin{aligned} \text{PR}(A) &= .03 + .85 \times \text{PR}(B)/4 \\ &= .0725 \end{aligned}$$

$$\text{PR}(B) = .03 + .85 \times \text{PR}(D)/1 = .2$$

$$\begin{aligned} \text{PR}(C) &= .03 + .85(\text{PR}(B)/4 + \\ &\quad \text{PR}(E)/2) \\ &= .03 + .85(.2/4 + .2/2) \\ &= .1575 \end{aligned}$$

$$\begin{aligned} \text{PR}(D) &= .03 + .85(\text{PR}(B)/4 + \\ &\quad \text{PR}(E)/2) \\ &= .1575 \end{aligned}$$

$$\begin{aligned} \text{PR}(E) &= .03 + .85(\text{PR}(B)/4 + \\ &\quad \text{PR}(C)/1) \\ &= .2425 \end{aligned}$$

# Calculating PageRank

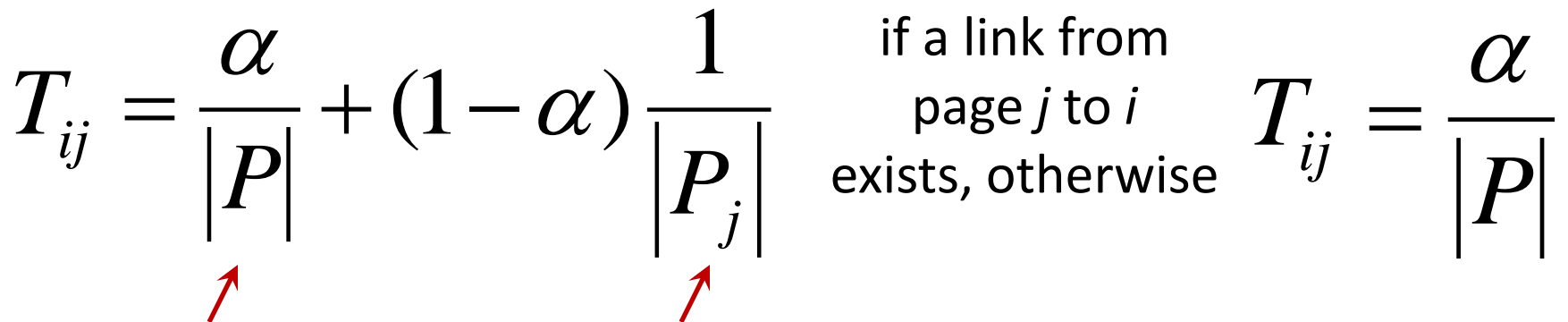
- PageRank is computed over and over until it converges, around 20 times
- Can also be calculated efficiently using matrix multiplication

# PageRank Definition as Matrix

Stated as a matrix equation where  $\mathbf{R}$  is the vector of PageRank values and  $\mathbf{T}$  the matrix for transition probabilities:

$$\mathbf{R} = \mathbf{T}\mathbf{R}$$

where  $T_{ij}$  is the probability of going from page  $j$  to  $i$ :

$$T_{ij} = \frac{\alpha}{|P|} + (1 - \alpha) \frac{1}{|P_j|} \quad \begin{array}{l} \text{if a link from} \\ \text{page } j \text{ to } i \\ \text{exists, otherwise} \end{array} \quad T_{ij} = \frac{\alpha}{|P|}$$


Total num of pages

Total outlinks from page  $j$

# PageRank Matrix Example

No link from C to A so value =  $0.15/5$

$$\begin{bmatrix} PR_A \\ PR_B \\ PR_C \\ PR_D \\ PR_E \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & .03 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & .243 & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} .2 \\ .2 \\ .2 \\ .2 \\ .2 \end{bmatrix}$$

$$T_{EB} = \frac{\alpha}{|P|} + (1 - \alpha) \frac{1}{|P_B|}$$

$$= 0.15/5 + (1 - 0.15)/4 = 0.243$$

Init PR values  
 $1/|P|$

# PageRank Issues

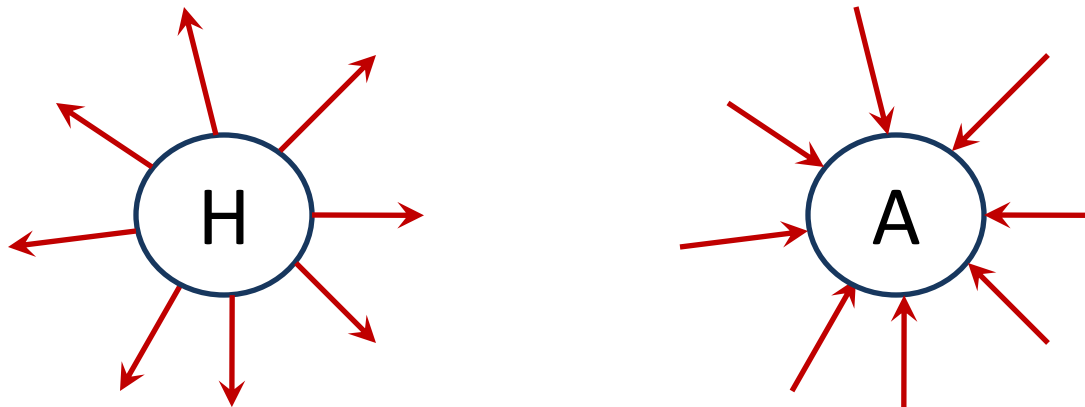
- Richer-get-richer phenomenon
  - May be difficult for new pages with few inlinks to compete with older, highly linked pages with high PageRank
  - Could promote small fraction of new pages at random<sup>1</sup> or add decay factor to links
- Study<sup>2</sup> showed just counting number of inlinks gives similar ranking as PageRank
  - Study was on small scale and pages were not necessarily “typical”
  - Counting inlinks more susceptible to spamming

<sup>1</sup>Pandey et al., Shuffling a stacked deck, VLDB 2005

<sup>2</sup>Amento et al., Does “authority” mean quality?

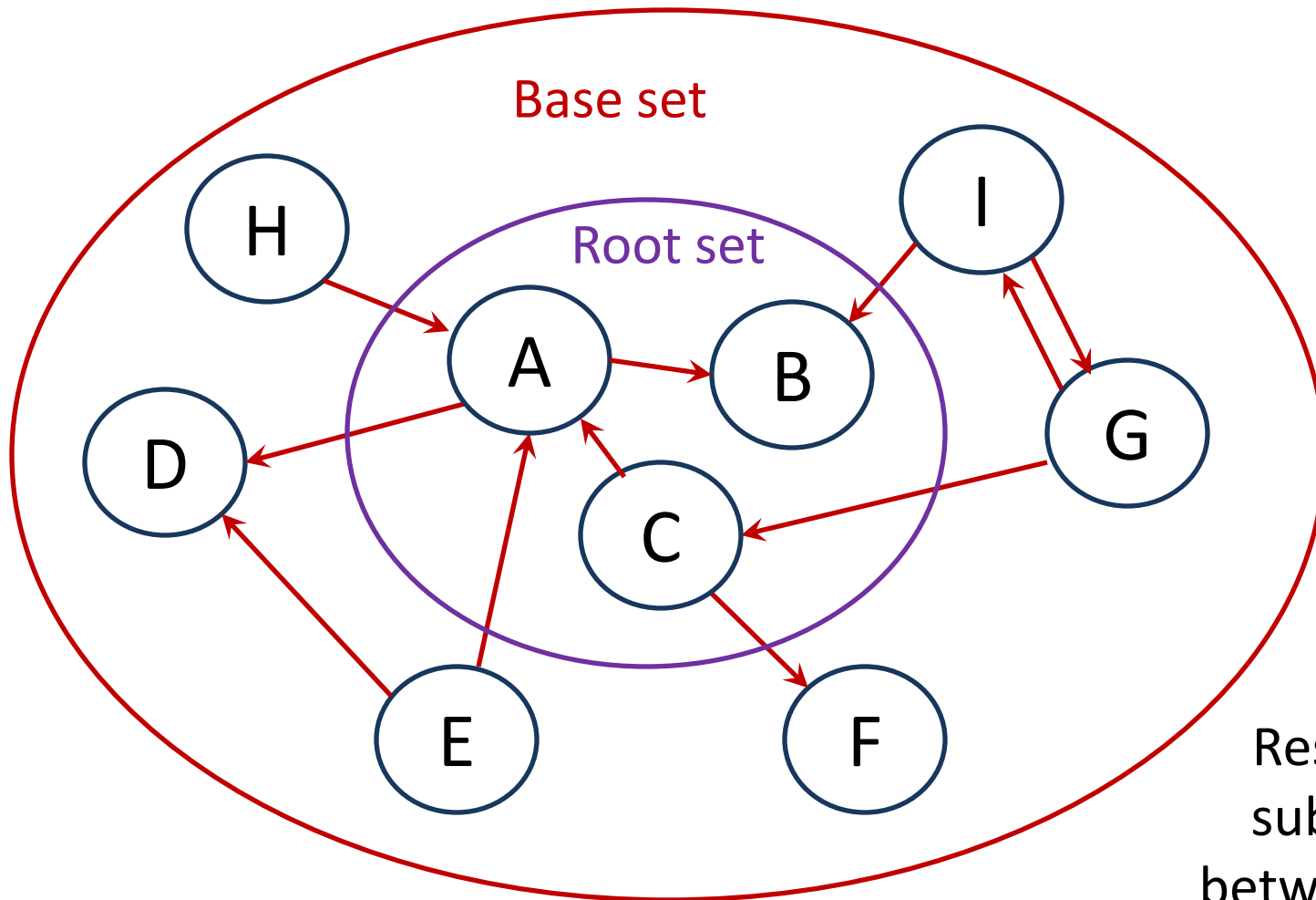
# HITS

- Hyperlink-induced topic search (HITS) by Kleinberg<sup>1</sup>
- **Hub**: page with outlinks to informative web pages
- **Authority**: informative/authoritative page with many inlinks
- Recursive definition:
  - Good hubs point to good authorities
  - Good authorities are pointed to by good hubs



<sup>1</sup>Kleinberg, Authoritative sources in a hyperlinked environment, *J. ACM*, 1999

# Root and Base Sets




Resulting  
subgraph  
between 1K –  
5K pages


# Calculate H and A

$E$  is set of all directed edges in subgraph  
 $e_{qp}$  is edge from page  $q$  to  $p$

$$A(p) = \sum_{q:e_{qp} \in E} H(q)$$

 inlinks

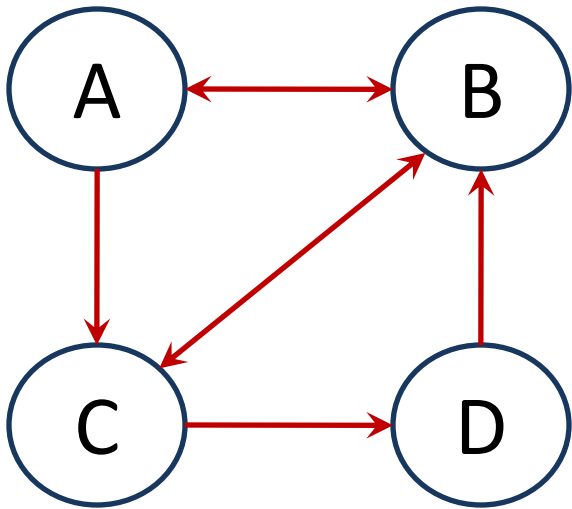
$$H(p) = \sum_{q:e_{pq} \in E} A(q)$$

 outlinks

# Calculating H and A

- H and A scores computed repetitively until they converge, about 10-15 iterations
- Can also be calculated efficiently using matrix multiplication

# Example Subgraph



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

Adjacency matrix

# Auth Scores

$$\begin{bmatrix} A_A \\ A_B \\ A_C \\ A_D \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

Best authority

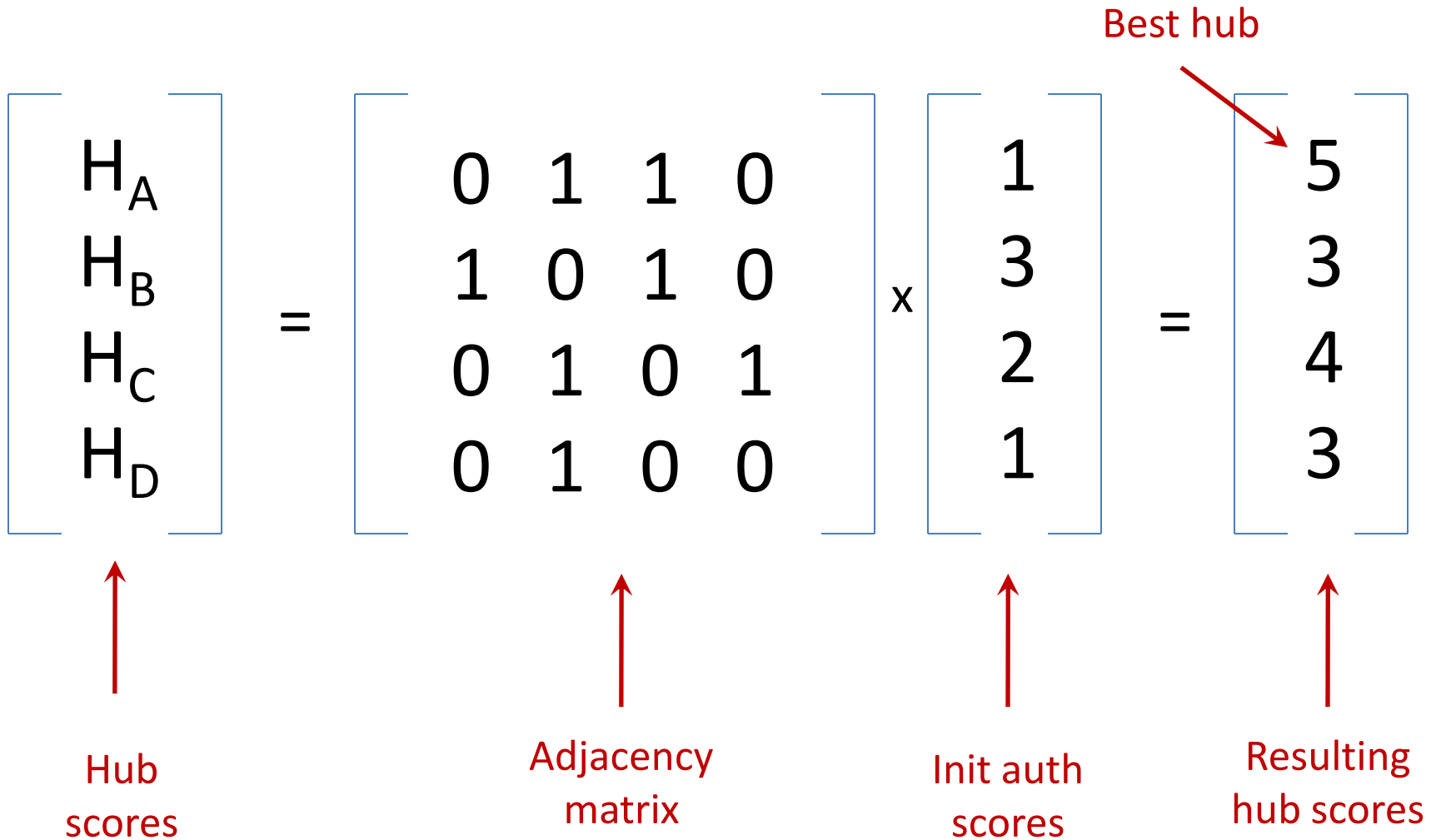
Authority scores

Transposed adjacency matrix

Init hub scores

Resulting auth scores

# Hub Scores



# Problems with HITS

- Has not been widely used
  - IBM holds patent
- Query dependence
  - Later implementations have made query independent
- Topic drift
  - Pages in expanded base set may not be on same topic as root pages
  - Solution is to examine link text when expanding

# Link Spam

- There is a strong economic incentive to rank highly in a SERP
- White hat SEO firms follow published guidelines to improve customer rankings<sup>1</sup>
- To boost PageRank, black hat SEO practices include:
  - Building elaborate link farms
  - Exchanging reciprocal links
  - Posting links on blogs and forums

<sup>1</sup>Google's Webmaster Guidelines

<http://www.google.com/support/webmasters/bin/answer.py?answer=35769>

# Combating Link Spam

- Sites like Wikipedia can discourage links than only promote PageRank by using “nofollow”  
`<a href="http://somesite.com/" rel="nofollow">Go here!</a>`
- Davison<sup>1</sup> identified 75 features for comparing source and destination pages
  - Overlap, identical page titles, same links, etc.
- TrustRank<sup>2</sup>
  - Bias teleportation in PageRank to set of trusted web pages

<sup>1</sup>Davison, Recognizing nepotistic links on the web, 2000

<sup>2</sup>Gyöngyi et al., Combating web spam with TrustRank, VLDB 2004

# Combating Link Spam cont.

- SpamRank<sup>1</sup>
  - PageRank for whole Web has power-law distribution
  - Penalize pages whose supporting pages do not approximate power-law dist
- Anti-TrustRank<sup>2</sup>
  - Give high weight to known spam pages and propagate values using PageRank
  - New pages can be classified spam if large contribution of PageRank from known spam pages or if high Anti-TrustRank

<sup>1</sup>Benczúr et al., SpamRank – fully automated link spam detection, AIRWeb 2005

<sup>2</sup>Krishnan & Raj, Web spam detection with Anti-TrustRank, AIRWeb 2006