

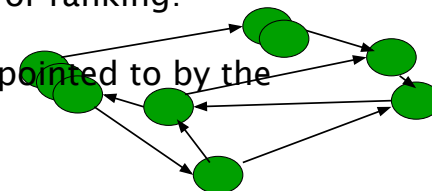
# Link Analysis and Page Rank

Information Retrieval  
Instructor: Dr. Swati Agrawal

## Hypertext and Links

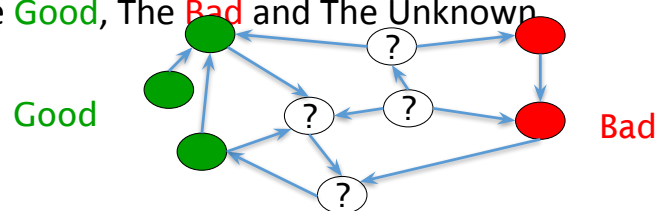
- We look beyond the *content* of documents
    - We begin to look at the *hyperlinks* between them
- Address questions like  
Do the links represent a conferral of authority to some pages? Is this useful for ranking?

How likely is it that a page pointed to by the CERN home page is about high energy physics



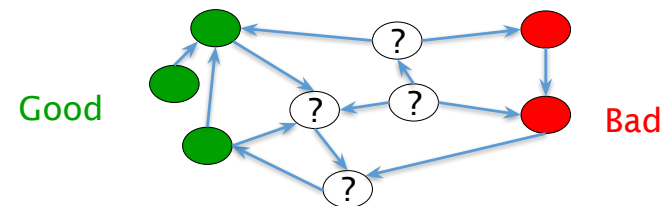
## Links are everywhere

- Powerful sources of authenticity and authority
  - Mail spam – which email accounts are spammers?
  - Host quality – which hosts are “bad”?
  - Phone call logs
- The *Good*, The *Bad* and The Unknown



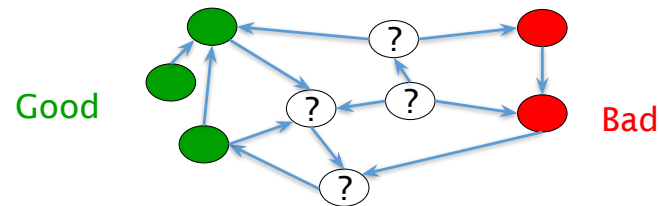
## Example 1: *Good*/*Bad*/*Unknown*

- The *Good*, The *Bad* and The Unknown
  - *Good* nodes won't point to *Bad* nodes
  - All other combinations plausible



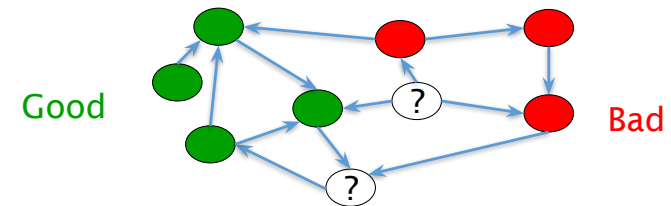
## Simple iterative logic

- **Good** nodes won't point to **Bad** nodes
  - If you point to a **Bad** node, you're **Bad**
  - If a **Good** node points to you, you're **Good**



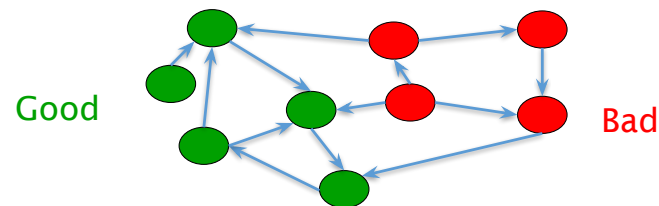
## Simple iterative logic

- **Good** nodes won't point to **Bad** nodes
  - If you point to a **Bad** node, you're **Bad**
  - If a **Good** node points to you, you're **Good**

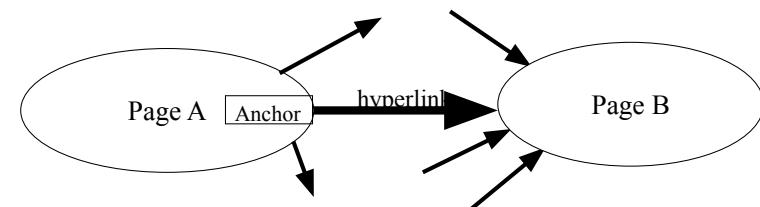


## Simple iterative logic

- **Good** nodes won't point to **Bad** nodes
  - If you point to a **Bad** node, you're **Bad**
  - If a **Good** node points to you, you're **Good**



## The Web as a Directed Graph

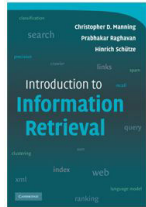


**Hypothesis 1:** A hyperlink between pages denotes a conferral of authority (quality signal)

**Hypothesis 2:** The text in the anchor of the hyperlink on page A describes the target page B

## Assumption 1: reputed sites

### Introduction to Information Retrieval



This is the companion website for the following book.

Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*

You can order this book at [CUP](#), at your local bookstore or on the internet. The best search

The book aims to provide a modern approach to information retrieval from a computer science University and at the University of Stuttgart

We'd be pleased to get feedback about how this book works out as a textbook, what is in comments to: [informationretrieval \(at\) yahoo \(dot\) com](mailto:informationretrieval@yahoo.com)

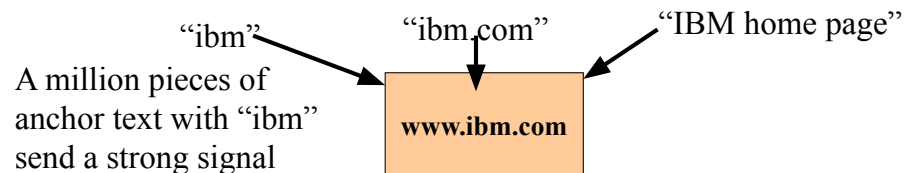
## Assumption 2: annotation of target



## Anchor Text *www Worm* - McBryan

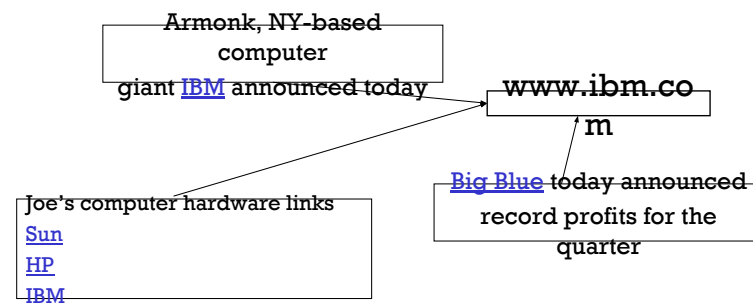
[Mcbr94]

- For **ibm** how to distinguish between:
  - IBM's home page (mostly graphical)
  - IBM's copyright page (high term freq. for 'ibm')
  - Rival's spam page (arbitrarily high term freq.)



## Indexing anchor text

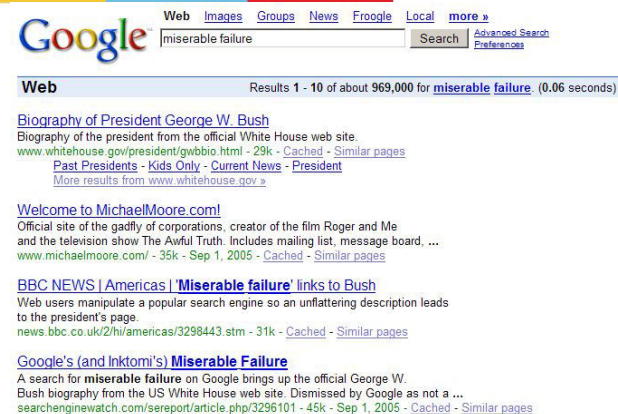
- When indexing a document  $D$ , include (with some weight) anchor text from links pointing to  $D$ .



## 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
  - Miserable failure-> George Bush Wikipedia page
  - Bad writers, Dumb and Dumber-> Game of Thrones Season 08
- Google introduced a new weighting function in January 2007 that fixed many Google bombs

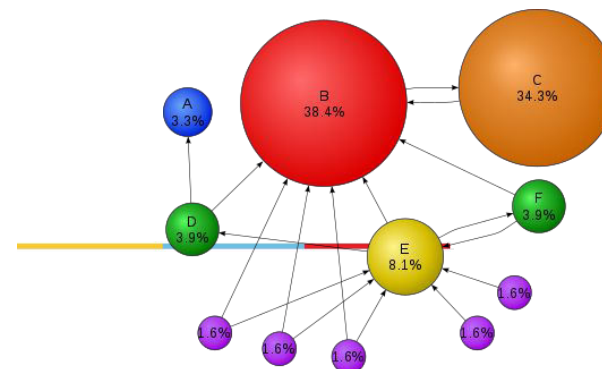
## Google bomb example



## Indexing anchor text

- Can score anchor text with weight depending on the authority of the anchor page's website
  - E.g., if we were to assume that content from cnn.com or yahoo.com is authoritative, then trust (more) the anchor text from them
  - Increase the weight of off-site anchors (non-nepotistic scoring)

## Link Analysis: PageRank



## Google Pagerank System

Google was developed by Sergey Brin and Larry Page

This is the method that Larry Page developed to rank and order the pages.

Hence, the **Pagerank**.

PageRank is a trademark of Google. The PageRank process has been patented.

## Citation Analysis

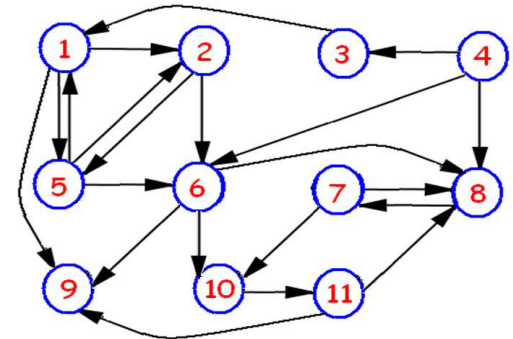
- Citation frequency
- **Bibliographic coupling frequency**
  - Articles that co-cite the same articles are related
- **Citation indexing**
  - Who is this author cited by? (Garfield 1972)
- Pagerank preview: Pinski and Narin '60s
  - Asked: which journals are authoritative?

## The web isn't scholarly citation

- Millions of participants, each with self interests
- Spamming is widespread
- Once search engines began to use links for ranking (roughly 1998), link spam grew
  - You can join a **link farm** – a group of websites that heavily link to one another

## How would you order these site?

- Suppose each of the nodes at right have the links shown in the directed graph. Which node is most important and should appear first?



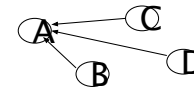
## The Basic Idea

1. How important a page is on the web
2. The more incoming edges to a page, the more important the page must be
3. How important the connecting page is

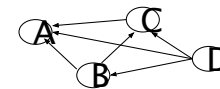
source: <http://www.webworkshop.net/pagerank.html>

## Simplified PageRank algorithm

- Assume four web pages: A, B, C and D. Let each page would begin with an estimated PageRank of 0.25.



$$PR(A) = PR(B) + PR(C) + PR(D).$$



$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}.$$

- $L(A)$  is defined as the number of links going out of page A. The PageRank of page A is given as follows:

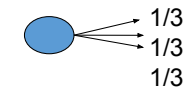
## PageRank algorithm including damping factor

Assume page A has pages B, C, D ..., which point to it. The parameter d is a damping factor which can be set between 0 and 1. Usually set d to 0.85. The PageRank of a page A is given as follows:

$$PR(A) = 1 - d + d \left( \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \dots \right)$$

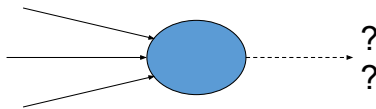
## Pagerank scoring

- Imagine a user doing a random walk on web pages:
  - 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 long run" each page has a long-term visit rate - use this as the page's score.



## 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

- At a dead end, jump to a random web page.
- Probability to jump =  $1/N$
- Can jump to current position as well
- Two ways to teleport
  - When at a node with no out-links, surfer invokes the teleport operation
  - When at a node with out-links, teleport operation with probability  $0 < \alpha < 1$
- At any non-dead end, with probability 10%, jump to a random web page.
  - With remaining probability (90%), go out on a random link.
  - 10% -  $\alpha$  parameter.

## Result of teleporting

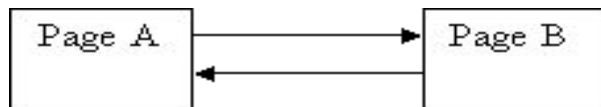
- Now cannot get stuck locally.
- There is a long-term rate at which any page is visited
- How do we compute this visit rate?

## Intuitive Justification

- A "random surfer" who is given a web page at random and keeps clicking on links, never hitting "back", but eventually gets bored and starts on another random page.
  - The probability that the random surfer visits a page is its PageRank.
  - The damping factor is the probability at each page the "random surfer" will get bored and request another random page.
- A page can have a high PageRank
  - If there are many pages that point to it
  - Or if there are some pages that point to it, and have a high PageRank.

## Compute the Page Rank of A and B

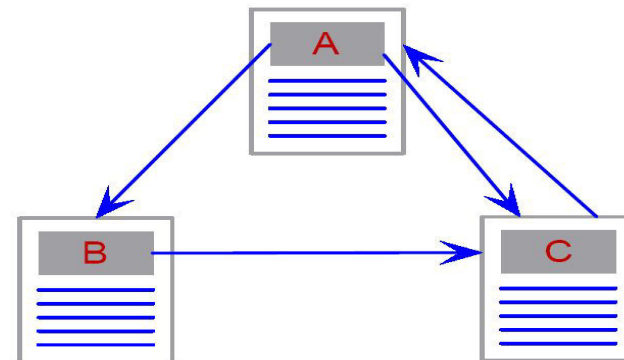
$d=0.5$



- Each page has one outgoing link. So that means  $L(A) = 1$  and  $L(B) = 1$ .
- $PR(A) = PR(B) = 1$

## What is the PageRank of A, B, C

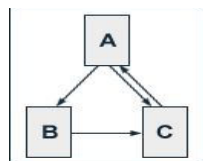
$d=0.5$



## The Characteristics of PageRank™

$D=0.5$

- $PR(A) = 0.5 + 0.5 PR(C)$
- $PR(B) = 0.5 + 0.5 (PR(A) / 2)$
- $PR(C) = 0.5 + 0.5 (PR(A) / 2 + PR(B))$



We get the following PageRank™ values for the single pages:

- $PR(A) = 14/13 = 1.07692308$
- $PR(B) = 10/13 = 0.76923077$
- $PR(C) = 15/13 = 1.15384615$

## Topic Specific PageRank

- aka personalized pagerank
- Previous pagerank computation: teleporting to a random web page chosen uniformly at random
- Now, teleporting to a web page chosen non-uniformly
- Computing the pagerank values tailored to particular interests

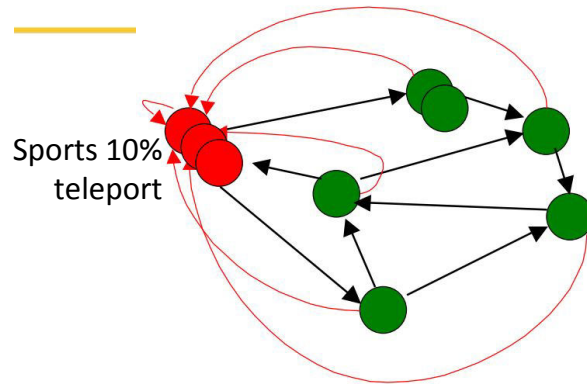


## Example

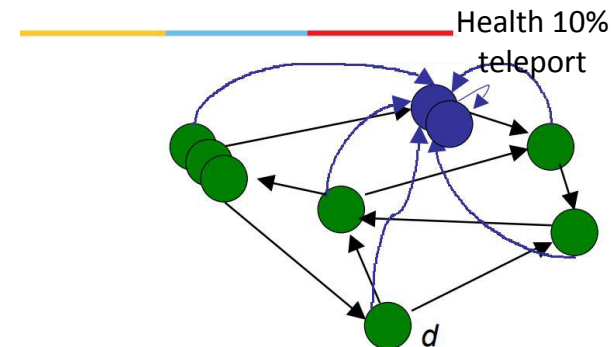
- A person having interest in sports might wish that pages on sports be ranked higher than non-sports page
- For a random surfer, in case of teleport operation as before, teleports to *a random web page on sports topic* instead of teleporting to a uniformly chosen random web page
- Requirement: a non-zero subset  $s$  of pages on sports topic to make the teleport operation feasible
- Topic specific pagerank

## Topic Specific PageRank

- The distribution over teleporting  $s$  need not to be uniformly chosen but could be arbitrary
- Similar topic specific pagerank distribution can be done for other domains as well
- Advantage: gives the potential of considering settings in which the search engine knows what topic a user is interested in
- Challenge: how to know users' interests
  - users explicitly register their interests
  - the system learns by observing each user's behavior over time



Non- dead nodes:  
10% teleport, 90% random walk



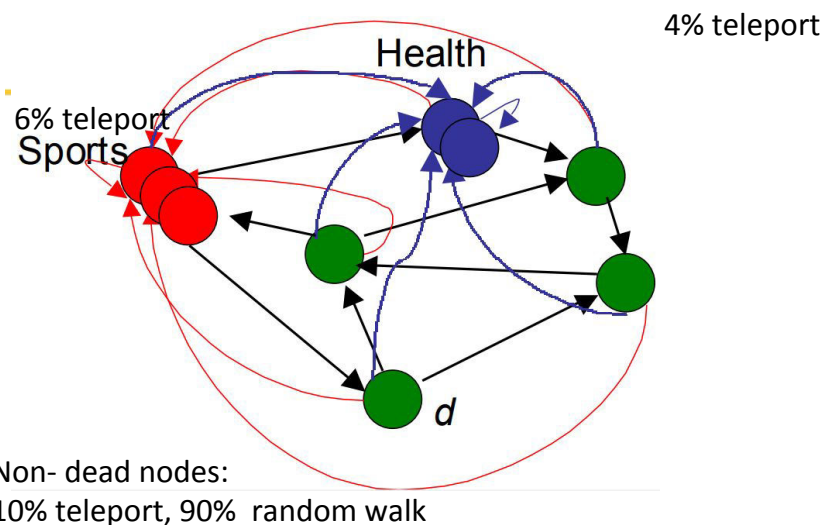
Non- dead nodes:  
10% teleport, 90% random walk

## Multiple Topics

- what if a user is known to have a mixture of interests from multiple topics?
- 60% sports and 40% health
- can we compute a *personalized PageRank* for this user and How?
- Assumption:
  - an individual's interests can be well-approximated as a linear combination of a small number of topic page distributions

## Multiple Topics

- A user with this mixture of interests could teleport as follows:
  - determine first whether to teleport to the set  $s$  of known sports pages, or to the set of known health pages.
  - This choice is made at random, choosing sports pages 60% of the time and health pages 40% of the time
  - Once we choose that a particular teleport step is to (say) a random sports page, we choose a web page in  $s$  uniformly at random to teleport to.



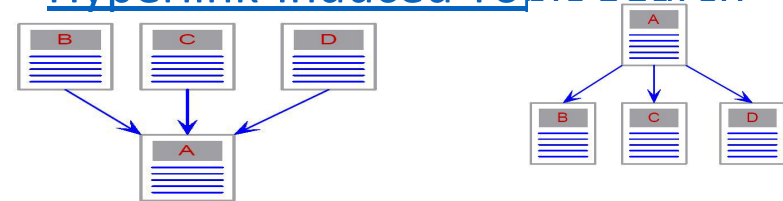
## Multiple Topics

- Idea is intuitive but difficult to implement or bring in practice
- Why?
  - for each user, we compute a transition probability matrix and compute its steady-state distribution
  - Pagerank computation for every distinct combination of user interests over topics

## Hubs and Authorities

- Given a query, every web page is assigned two scores
  - Hub score
  - Authority score
- Two ranked lists of results are computed
- Useful for broad search topics
- Query: I wish to learn “PageRank”
- Results:
  - authoritative source of information on pagerank
  - Hand-compiled lists (not authoritative in themselves) of links to authoritative web pages on specific topic
- Good hub page: pointing to many good authorities
- Good authority page: pointed to by many good hub pages

## HITS – Hubs and Authorities - Hyperlink-Induced Topic Search



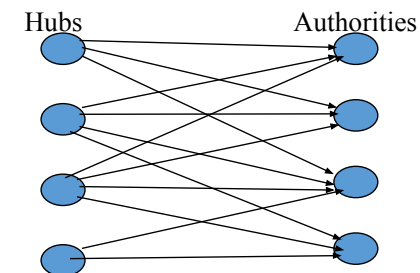
- **A** on the left is an **authority**
- **A** on the right is a **hub**

## 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.

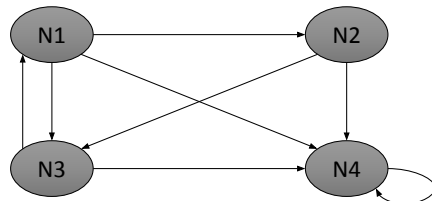
## Hubs and Authorities

- Together they tend to form a bipartite graph:



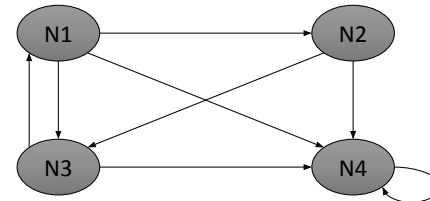
## Example

	N1	N2	N3	N4
N1	0	1	1	1
N2	0	0	1	1
N3	1	0	0	1
N4	0	0	0	1



Graph with the nodes

## Example



Nodes	Hub (out-degree)	Authority (in-degree)
N1	3	1
N2	2	1
N3	2	2
N4	1	4

Ranks using out-degree and in-degree

Hub: N1, N2, N3 (tie), N4  
Authority: N4, N3, N2, N1 (tie)

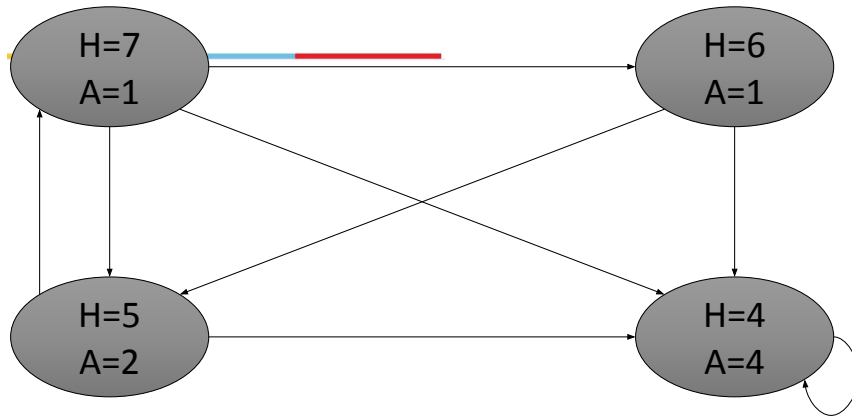
## HITS algorithm

- **Input:** an adjacency matrix A, initial hub weight vector u
- Authority weight vector,  $v = A^T * u$
- Updated hub weight vector,  $u = A * v$
- Recreate the node graph with new hub and authority values (k=1)
- Recreate the rank table using new hub and authority scores

$$\begin{array}{c}
 \begin{array}{c|cccc}
 & N1 & N2 & N3 & N4 \\
 \hline
 N1 & 0 & 1 & 1 & 1 \\
 N2 & 0 & 0 & 1 & 1 \\
 N3 & 1 & 0 & 0 & 1 \\
 N4 & 0 & 0 & 0 & 1
 \end{array}
 \xrightarrow{\text{transpose}}
 \begin{array}{c|cccc}
 & N1 & N2 & N3 & N4 \\
 \hline
 N1 & 0 & 0 & 1 & 0 \\
 N2 & 1 & 0 & 0 & 0 \\
 N3 & 1 & 1 & 0 & 0 \\
 N4 & 1 & 1 & 1 & 1
 \end{array}
 \begin{array}{c}
 \\
 \\
 A^T \\
 u \quad v
 \end{array}
 \begin{array}{c}
 \\
 \\
 * \\
 \\
 \end{array}
 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
 =
 \begin{bmatrix} 1 \\ 1 \\ 2 \\ 4 \end{bmatrix}
 \end{array}$$
  

$$\begin{array}{c|cccc}
 & N1 & N2 & N3 & N4 \\
 \hline
 N1 & 0 & 1 & 1 & 1 \\
 N2 & 0 & 0 & 1 & 1 \\
 N3 & 1 & 0 & 0 & 1 \\
 N4 & 0 & 0 & 0 & 1
 \end{array}
 \begin{array}{c}
 \\
 \\
 A \\
 v \quad \text{hub } u
 \end{array}
 \begin{array}{c}
 \\
 \\
 * \\
 \\
 \end{array}
 \begin{bmatrix} 1 \\ 1 \\ 2 \\ 4 \end{bmatrix}
 =
 \begin{bmatrix} 7 \\ 6 \\ 5 \\ 4 \end{bmatrix}$$

K=1



### Ranks using new hub and authority

values (k=1)

Nodes	Hub (u)	Authority (v)
N1	7	1
N2	6	1
N3	5	2
N4	4	4



K=2

$$\text{Authority } v_i' = \frac{v_i}{\sqrt{\sum_{i=1}^n v_i^2}}$$

$$\text{Hub } u_i' = \frac{u_i}{\sqrt{\sum_{i=1}^n u_i^2}}$$

Hub: N1, N2, N3, N4

Authority: N4, N3, N2, N1 (tie)

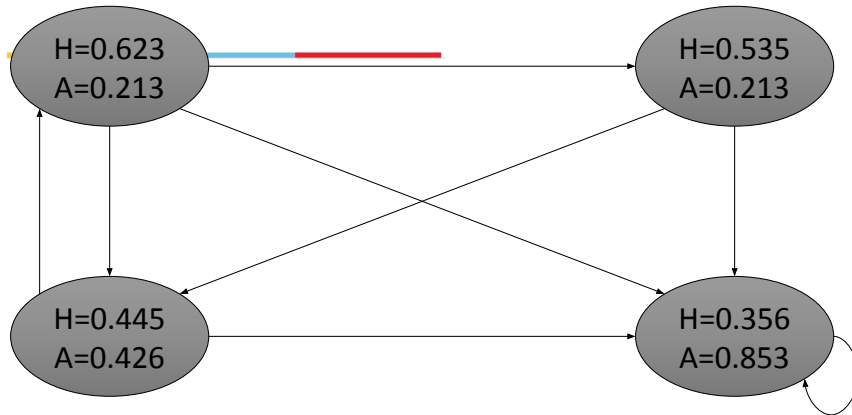
$$V' = \left[ \frac{1}{\sqrt{1^2+1^2+1^2+4^2}}, \frac{1}{\sqrt{1^2+1^2+1^2+4^2}}, \frac{2}{\sqrt{1^2+1^2+1^2+4^2}}, \frac{4}{\sqrt{1^2+1^2+1^2+4^2}} \right]$$

$$V' = \left[ \frac{1}{\sqrt{22}}, \frac{1}{\sqrt{22}}, \frac{2}{\sqrt{22}}, \frac{4}{\sqrt{22}} \right] = [0.213, 0.213, 0.426, 0.853]$$

Similarly,

$$U' = \left[ \frac{7}{\sqrt{126}}, \frac{6}{\sqrt{126}}, \frac{5}{\sqrt{126}}, \frac{4}{\sqrt{126}} \right] = [0.623, 0.535, 0.445, 0.356]$$

K=2



### Ranks using new hub and authority

values (k=2)

Nodes	Hub (u')	Authority (v')
N1	0.623	0.213
N2	0.535	0.213
N3	0.445	0.426
N4	0.356	0.853

Hub: N1, N2, N3, N4

Authority: N4, N3, N2, N1 (tie)

For k=3

$$\text{For } v, 0.213^2 + 0.213^2 + 0.426^2 + 0.853^2 = 0.999$$

$$\text{For } u, 0.623^2 + 0.535^2 + 0.445^2 + 0.356^2 = 0.999$$

## 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

## Recap

- Anchor text and hyperlinks
- Web as a graph
- Pagerank basic algorithm
- Random Surfing
- Teleporting
- Topic specific pagerank
  - Single topic
  - Multiple topics
- HITS algorithm (hubs and authorities)

## Google PageRank

- The original formula

$$PR(A) = (1-d) + d \left( \frac{PR(T_1)}{C(T_1)} + \frac{PR(T_2)}{C(T_2)} + \dots + \frac{PR(T_n)}{C(T_n)} \right)$$

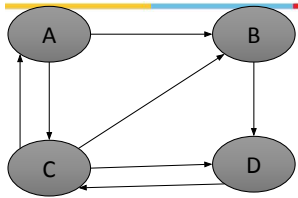
- a given **A** website page rank depends on other websites' page rank that are linking to **A**.
- **C(T)** parameter is the number of outgoing links from website **T**
- **d** is the damping factor: a value in the range **0** and **1**

## Google Pagerank

- We have to initialize the page ranks at the beginning
- The usual approach is to initialize every page rank to be **1/n**
- We make several iterations until convergence

$$PR_{t+1}(P_i) = \sum_{P_j} \frac{PR_t(P_j)}{C(P_j)}$$

## Example

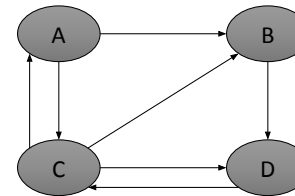


	I0	I1	I2	PR
A	1/4	1/12	1.5/12	1
B	1/4	2.5/12	2/12	2
C	1/4	4.5/12	4.5/12	4
D	1/4	4/12	4/12	3

$$PR_{t+1}(P_i) = \sum_{P_j} \frac{PR_t(P_j)}{C(P_j)}$$

## Matrix Representation

- In iterative approach, we update the values one-by-one
- We can use matrix multiplication to do multiple calculations at a time



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

Column stochastic matrix

## Matrix Representation

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

Power method

$$PR_{t+1} = H * PR_t$$

	I0	I1	I2	PR
A	1/4	1/12	1.5/12	1
B	1/4	2.5/12	2/12	2
C	1/4	4.5/12	4.5/12	4
D	1/4	4/12	4/12	3

$$V = \begin{bmatrix} 1/4 \\ 1/4 \\ 1/4 \\ 1/4 \end{bmatrix}$$

$$v_2 = H v$$

$$v_3 = H v_2 = H (H v) = H^2 v$$

$$v_4 = H v_3 = H^3 v$$