

# Web Structure Mining

## Link Analysis Algorithms

Li Xiaoli

# Road map

- **Introduction**
- **Social network analysis**
- **Co-citation and bibliographic coupling**
- **PageRank**
- **HITS**
- **Summary**

# Introduction

- Early search engines mainly compare content similarity of the query and the indexed pages. i.e.,
  - They use information retrieval methods, cosine, TF-IDF, ...
- From 1996, it became clear that **content similarity alone was no longer sufficient**.
  - The number of pages grew rapidly in the mid-late 1990's.
    - Try “classification technique”, Google estimates: 196 million relevant pages.
    - How to choose only 30-40 pages and rank them suitably to present to the user?
  - **Content similarity is easily spammed**.
    - A page owner can repeat some words and add many related words to boost the rankings of his pages and/or to make the pages relevant to a large number of queries.

# Introduction (cont ...)

- Starting around 1996, researchers began to work on the problem. They resort to **hyperlinks**.
  - In Feb, 1997, Yanhong Li (Robin Li), Scotch Plains, filed a hyperlink based search patent. The method uses words in anchor text of hyperlinks.
- Web pages are connected through hyperlinks, which carry important information.
  - Some **hyperlinks**: organize information at the same site.
  - Other **hyperlinks**: point to pages from other Web sites. Such **out-going hyperlinks** often indicate an **implicit conveyance of authority** to the pages being pointed to.
- Those pages that are pointed to by many other pages are likely to contain authoritative information.

# Introduction (cont ...)

- During 1997-1998, two most influential hyperlink based search algorithms **PageRank** and **HITS** were reported.
- Both algorithms are related to **social networks**. They exploit the hyperlinks of the Web to rank pages according to their levels of “prestige” or “authority”.
  - **HITS**: Jon Kleinberg (Cornel University), at *Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, January 1998
  - **PageRank**: Sergey Brin and Larry Page, PhD students from Stanford University, at *Seventh International World Wide Web Conference (WWW7)* in April, 1998.
- **PageRank powers the Google search engine.**

# Link Analysis Tasks

- Link-based Object Classification (LOC)
  - Assign class labels to entities based on their link characteristics
  - E.g. Enhance Web page classification by incorporating the anchor text and other pages, disease gene prediction
- Link-based Object Ranking (LOR)
  - Associate a relative quantitative assessment with each entity using link-based measures
  - E.g. PageRank, HITS, SimRank
- Link prediction
  - Extrapolating knowledge/pattern of links in a given network to deduce novel links that are plausible, and may occur in the future
  - E.g. Recommendation systems (friend recommendation)

# Road map

- Introduction
- **Social network analysis**
- Co-citation and bibliographic coupling
- PageRank
- HITS
- Summary

# Social network analysis

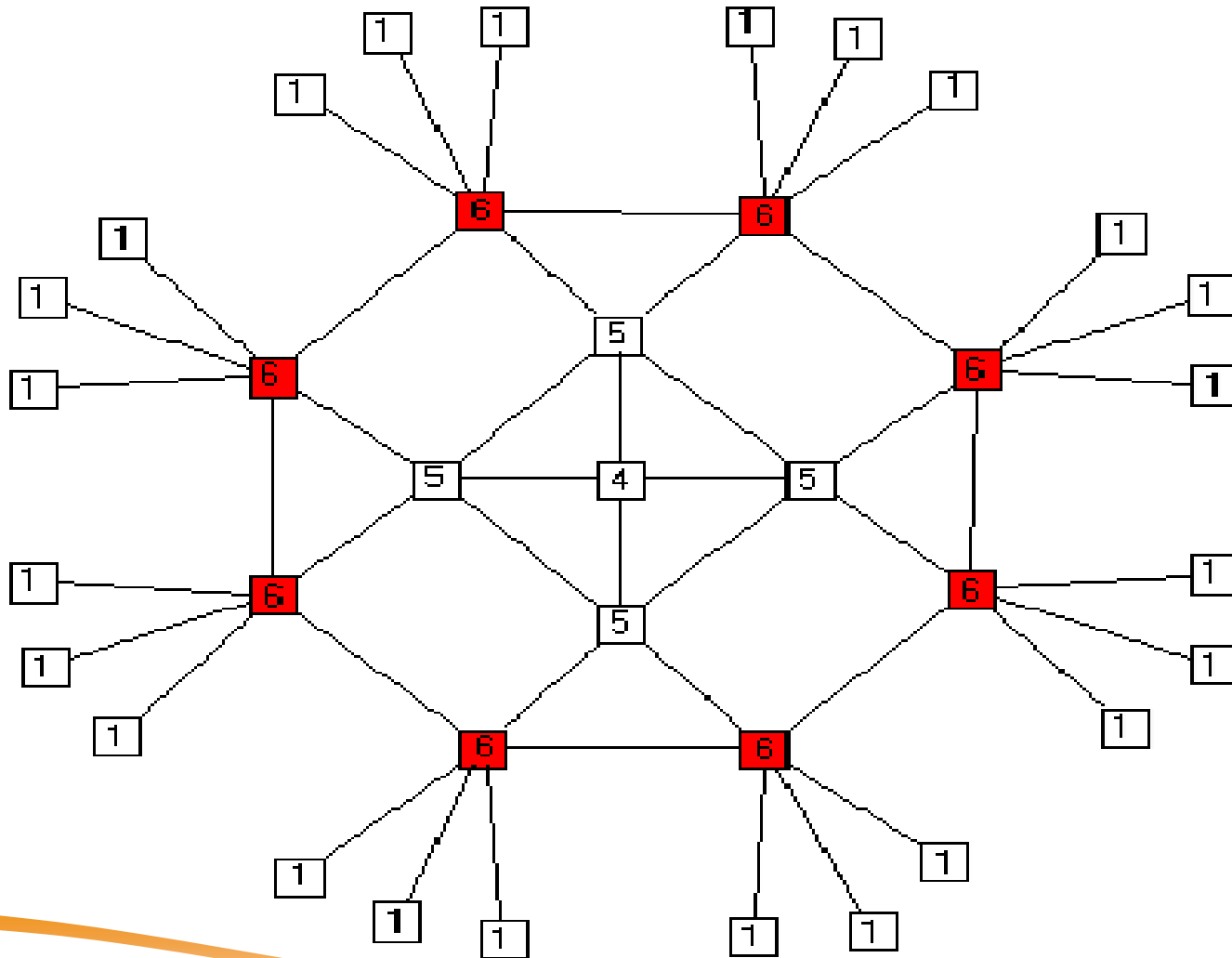
- Social network is the study of social entities (people in an organization, called **actors**), and their **interactions and relationships**, which can be represented as a **network or graph**,
  - each vertex (or node) represents an actor and
  - each link represents a relationship.
- From the network, we can study the properties of its structure, and **the role, position** and **prestige** of each social actor.
- We can also find various kinds of sub-graphs, e.g., **communities** formed by groups of actors.
- SNA is relevant to advertising, national security, medicine, geography, politics, social psychology, etc.



# Social network and the Web

- Social network analysis is useful for the Web because the Web is essentially a virtual society, and thus a virtual social network,
  - Each page: a social actor and
  - each hyperlink: a relationship.
- Many results from social network can be adapted and extended for use in the Web context.
- We study two types of social network analysis, **centrality** and **prestige**, which are closely related to hyperlink analysis and search on the Web.

# Degree Centrality



# Degree Centrality –

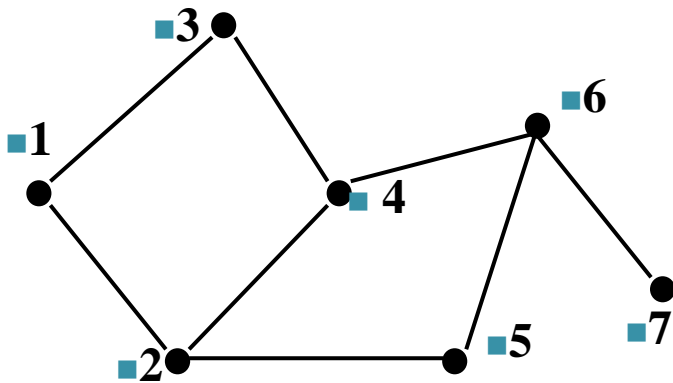
We already know this

- Central actors are the most active actors that have most links or tiers with other actors.
- The degree of a node is the number of edges incident on it, which is the simplest indicator of how connected a node is within a graph
- In a directed graph, in-degree is the no. of incoming edges, and out-degree the no. of outgoing one
- For directed graphs, total degree = in-degree + out-degree (used in some software)

# Closeness Centrality (**Example**)

- It is based on the closeness or distance. An actor  $x_i$  is *central* if it can easily interact with all other actors. That is, its distance to all other actors is short. Thus, the shortest distance is used to compute this measure. Let the shortest distance from actor  $i$  to actor  $j$  be  $d(i, j)$  (measured as the number of links in a shortest path).

$$C_c(i) = \frac{n-1}{\sum_{j=1}^n d(i, j)}$$



$$C_c(1) = (7-1)/(1+1+2+2+3+4) = 6/13$$

1-→2, 1-→3, 1-→4, 1-→5, 1-→6, 1-→7

$$C_c(4) = (7-1)/(2+1+1+2+1+2) = 6/9$$

4-→1, 4-→2, 4-→3, 4-→5, 4-→6, 4-→7

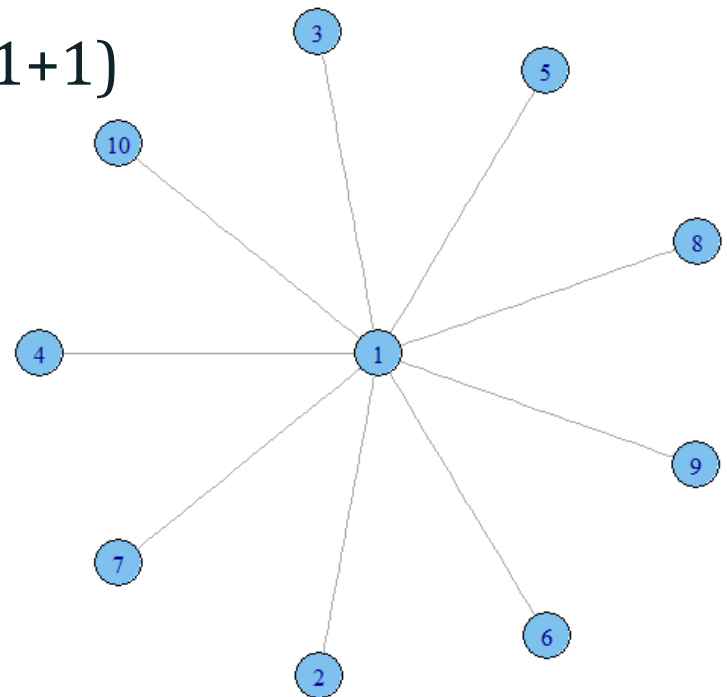
# Closeness Centrality (**Example**)

The value of this measure also ranges between 0 and 1 as  $n - 1$  is the *minimum* value of the denominator, which is the sum of the shortest distances from  $i$  to *all other actors*.

Example: Closeness Centrality gets the maximal value

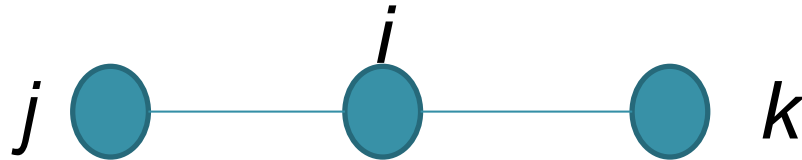
$$C_c(1) = (10-1)/(1+1+1+1+1+1+1+1+1+1) \\ = 1$$

$$C_c(i) = \frac{n-1}{\sum_{j=1}^n d(i,j)}$$



# Betweenness Centrality

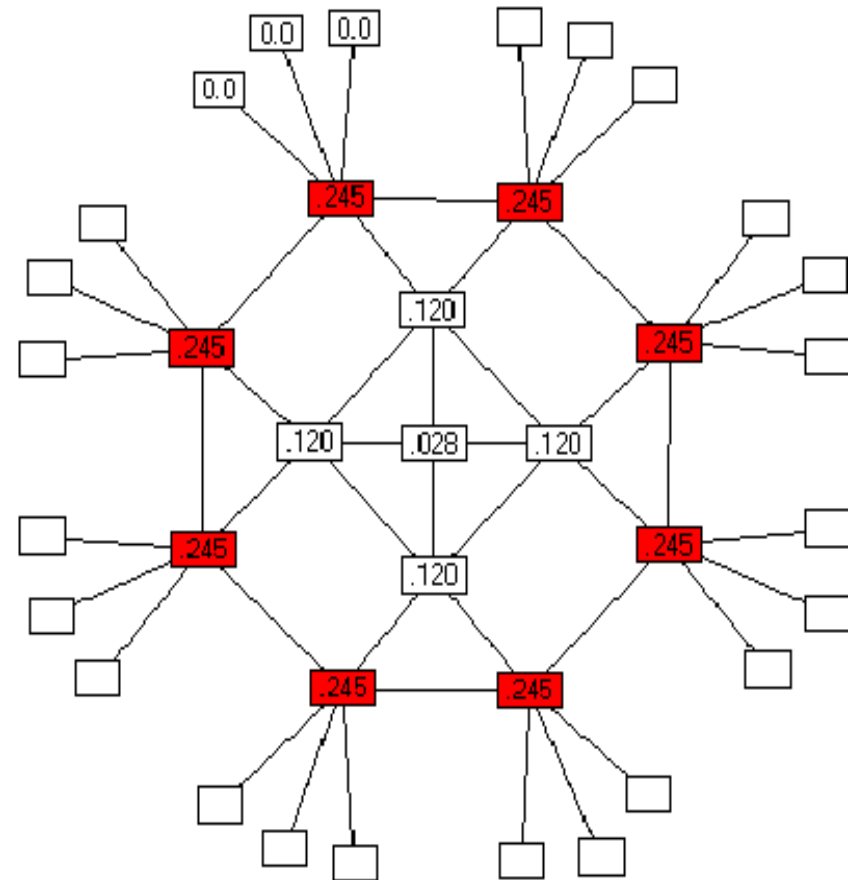
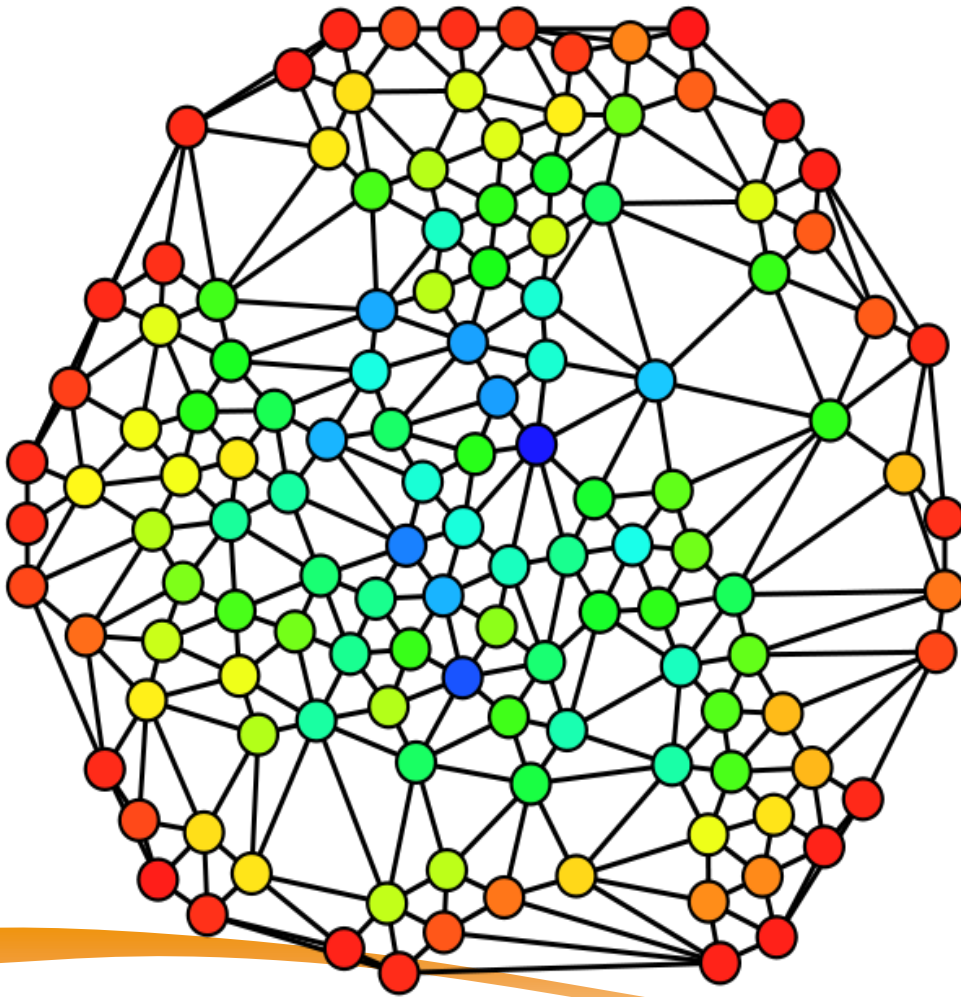
- If two non-adjacent actors  $j$  and  $k$  want to interact and actor  $i$  is on the path between  $j$  and  $k$ , then  $i$  may have some control over the interactions between  $j$  and  $k$ .



- **Betweenness** measures this control of  $i$  over other pairs of actors. Thus,
  - if  $i$  is on the paths of *many* such interactions, then  $i$  is an important actor.

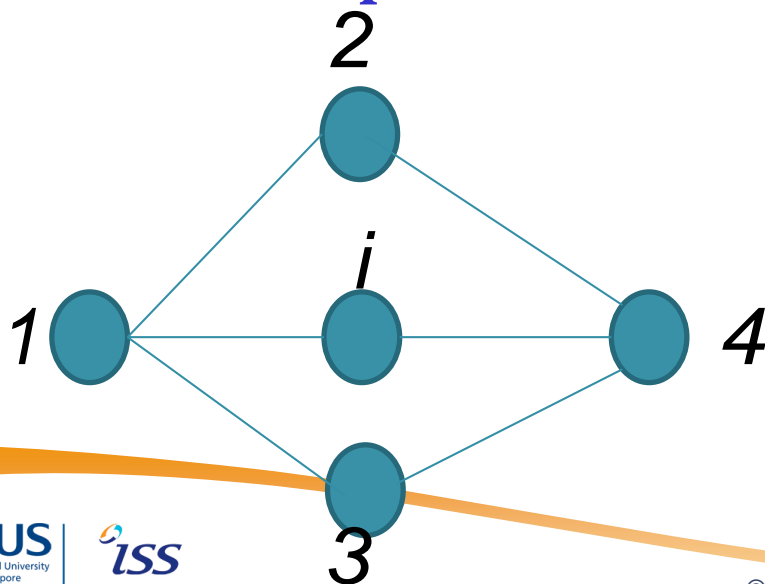
# Node Betweenness

from red=0 to blue=max



# Example: Betweenness Centrality

- **Undirected graph:** Let  $p_{jk}$  be the number of shortest paths between actor  $j$  and actor  $k$ .
- The betweenness of an actor  $i$  is defined as the number of shortest paths that pass  $i$   $p_{jk}(i)$  normalized by the total number of shortest paths.



$$C_B(i) = \sum_{j < k} \frac{p_{jk}(i)}{p_{jk}}$$

$$C_c(i) = 0/1 + 0/1 + 1/3 + 0/2 + 0/1 + 0/1 = 1/3$$

(1,2), (1,3), (1, 4), (2, 3), (2, 4), (3, 4)



# Prestige

- Prestige is a more **refined** measure of prominence of an actor than centrality.
  - Distinguish: ties sent (**out-links**) and ties received (**in-links**).
- A prestigious actor has **extensive ties as a recipient**.
  - To compute the prestige: we use **only in-links**.
- We study two prestige measures. **Rank prestige** forms the basis of most Web page link analysis algorithms, including **PageRank** and **HITS**.

# Prestige (cont ...)

- An actor is prestigious if it receives many in-links or nominations. Thus, the simplest measure of prestige of an actor  $i$  (denoted by  $P_D(i)$ ) is its in-degree.

$$P_D(i) = \frac{d_I(i)}{n - 1}$$

where  $d_I(i)$  is the in-degree of  $i$  (the number of in-links of  $i$ ). As in the degree centrality, dividing by  $n - 1$  standardizes the prestige value to the range from 0 and 1. The maximum prestige value is 1 when every other actor links to or chooses actor  $i$ .

# Proximity prestige

- The standard prestige of an actor  $i$  only considers the actors that are adjacent to  $i$ .
- The **proximity prestige** generalizes it by considering both the actors directly and *indirectly* linked to actor  $i$ .
- Let  $I_i$  be the set of actors that can **reach** actor  $i$ .
- The **proximity** is defined as distance of other actors to  $i$ .

# Proximity prestige (cont ...)

- Let  $d(j, i)$  denote the distance from actor  $j$  to actor  $i$ .
- We consider every actor  $j$  that can reach  $i$ , i.e., there is a directed path from  $j$  to  $i$ .

$$\frac{\sum_{j \in I_i} d(j, i)}{|I_i|}$$

- Let  $|I_i|$  be the set of actors that can reach actor  $i$ . The proximity is defined as average distance of other actors to  $i$ .

# Road map

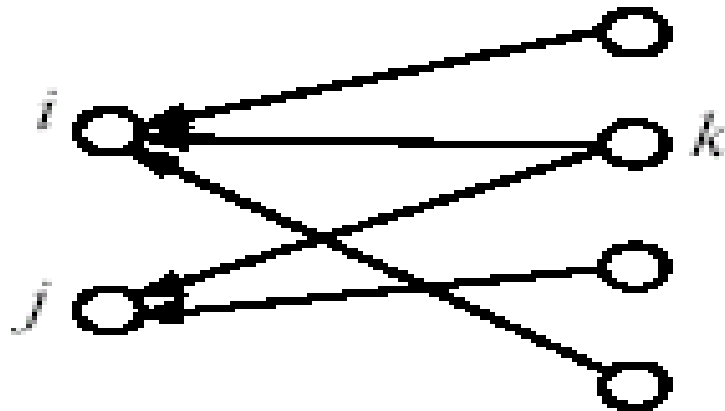
- Introduction
- Social network analysis
- **Co-citation and bibliographic coupling**
- PageRank
- HITS
- Summary

# Co-citation and Bibliographic Coupling

- Another area of research concerned with links is **citation analysis** of scholarly publications.
  - A scholarly publication cites related prior work to acknowledge the origins of some ideas and to compare the new proposal with existing work.
- When a paper cites another paper, a relationship is established between the publications.
  - Citation analysis uses these relationships (links) to perform various types of analysis.
- We discuss two types of citation analysis, **co-citation** and **bibliographic coupling**. The HITS algorithm is related to these two types of analysis.

# Co-citation

- If papers  $i$  and  $j$  are both cited by paper  $k$ , then they may be related in some sense to one another.
- The more papers they are cited by, the stronger their relationship is.




1 to multiple

Paper  $i$  and paper  $j$  are co-cited by paper  $k$

# Co-citation

- Let  $\mathbf{L}$  be the citation matrix. Each cell of the matrix is defined as follows:
  - $L_{ij} = 1$  if paper  $i$  cites paper  $j$ , and 0 otherwise.
- Co-citation** (denoted by  $C_{ij}$ ) is a similarity measure defined as the number of papers that co-cite  $i$  and  $j$ ,

$$C_{ij} = \sum_{k=1}^n L_{ki} L_{kj},$$

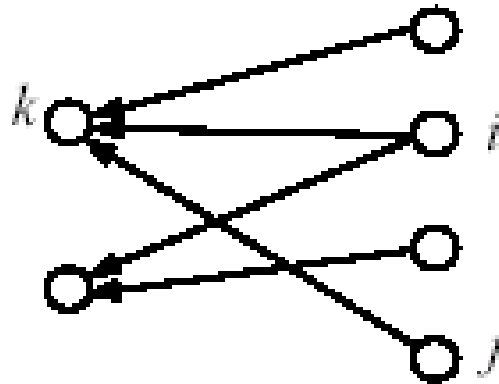

$$\begin{bmatrix} L_{1i} & L_{1j} \\ L_{2i} & L_{2j} \\ \dots & \dots \\ L_{ni} & L_{nj} \end{bmatrix}$$

- $C_{ii}$  is naturally the number of papers that cite  $i$ .
- A square matrix  $\mathbf{C}$  can be formed with  $C_{ij}$ , and it is called the **co-citation matrix**.



# Bibliographic coupling

- Bibliographic coupling operates on a similar principle.
- Bibliographic coupling links papers that cite the same articles
  - if papers  $i$  and  $j$  both cite paper  $k$ , they may be related.
- The more papers they both cite, the stronger their similarity is.



Multiple to 1

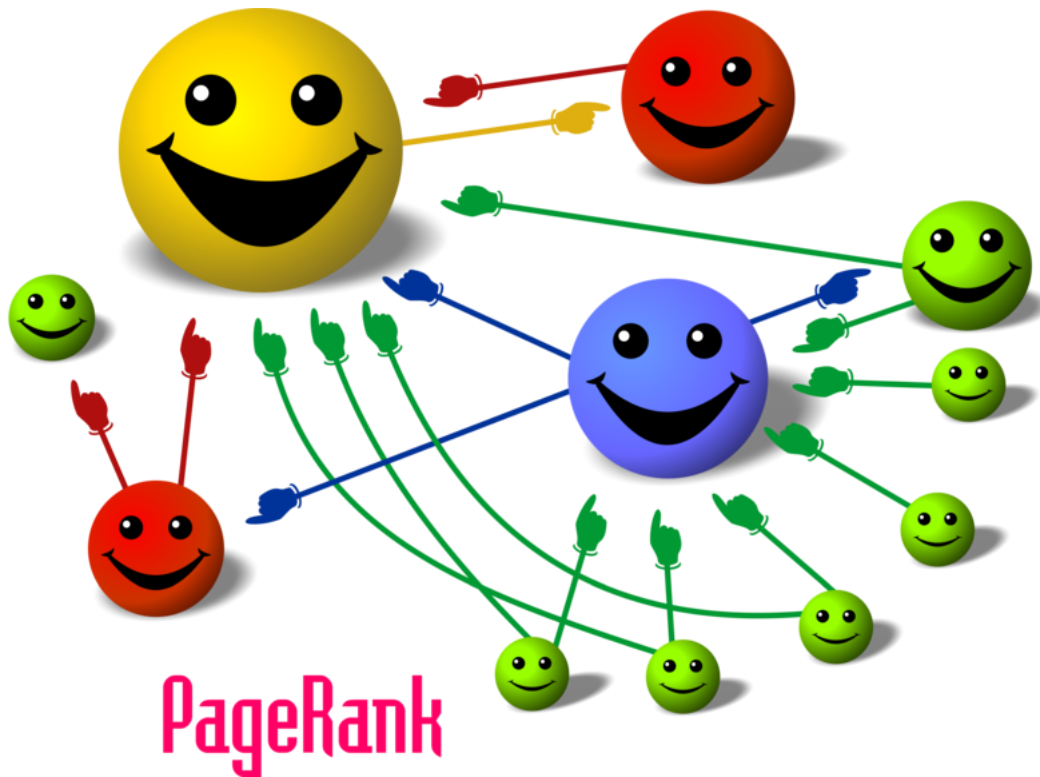
Both Paper  $i$  and paper  $j$  cite paper  $k$

# Road map

- Introduction
- Social network analysis
- Co-citation and bibliographic coupling
- PageRank
- HITS
- Summary

# PageRank

- How does Google® rank web pages in order to provide meaningful search results?



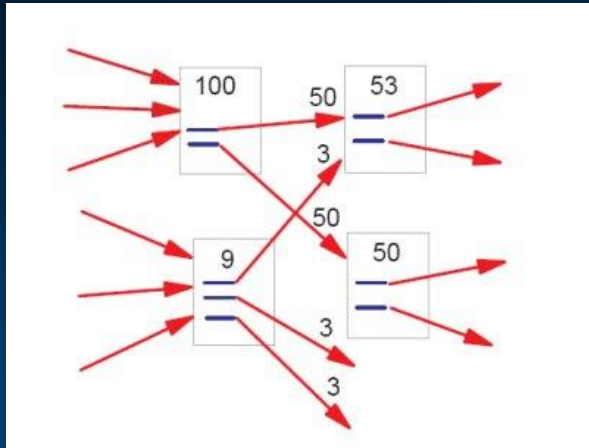
Google



Larry Page and Sergey Brin

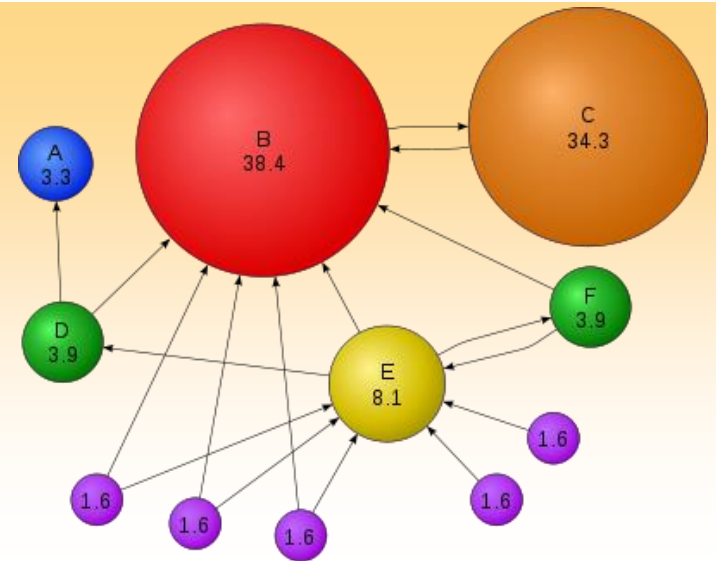
■ [www.validdomainauctions.com](http://www.validdomainauctions.com)

# The PageRank Algorithm



PageRank assigns numerical ranks to pages based on backlink counts and ranks of pages providing those backlinks.

The algorithm considers a model in which a user starts at a webpage and performs a “random walk” by following links from the page he is currently in. To start another such walk, a new webpage may be opened occasionally. PageRank of a webpage is the probability of that webpage being visited on a particular random walk.



<http://hamletbatista.com/2007/10/29/pagerank-caught-in-the-paid-link-crossfire/>

<http://www.prlog.org/10235329-use-twitter-social-networking-for-your-business-build-google-pagerank.html/>

# PageRank

- The year 1998 was an eventful year for Web link analysis models. Both the **PageRank** and **HITS** algorithms were reported in that year.
- The connections between PageRank and HITS are quite striking.
- Since that eventful year, PageRank has emerged as the dominant link analysis model,
  - due to its query-independence,
  - its ability to combat spamming, and
  - Google's huge business success.

# PageRank: the intuitive idea

- PageRank relies on the democratic nature of the Web by using its vast link structure as an indicator of an individual page's value or quality.
- PageRank interprets a hyperlink from page  $x$  to page  $y$  as a vote, by page  $x$ , for page  $y$ .
- However, PageRank looks at more than the sheer number of votes; it also analyzes the page that casts the vote.
  - Votes casted by “important” pages weigh more heavily and help to make other pages more “important.”
- This is exactly the idea of **rank prestige** in social network.

# More specifically

- A hyperlink from a page to another page is an implicit conveyance of authority to the target page.
  - The more in-links that a page  $i$  receives, the more prestige the page  $i$  has.
- Pages that point to page  $i$  also have their own prestige scores.
  - A page of a higher prestige pointing to  $i$  is more important than a page of a lower prestige pointing to  $i$ .
  - In other words, a page is important if it is pointed to by other important pages.

When you try to find a job, who should you ask to write a recommendation letter? **BIG NAME**

# PageRank algorithm

- According to **rank prestige**, the importance of page  $i$  ( $i$ 's PageRank score) is the sum of the PageRank scores of all pages that point to  $i$ .
- Since a page may point to many other pages, its prestige score should be **shared**.
- The Web as a directed graph  $G = (V, E)$ . Let the total number of pages be  $n$ . The PageRank score of the page  $i$  (denoted by  $P(i)$ ) is defined by:

$$P(i) = \sum_{(j,i) \in E} \frac{P(j)}{O_j},$$

$O_j$  is the number of out-link of  $j$  (or  $j$ 's out -degree)



# The PageRank Algorithm

$$PR(P_u) = (1 - d) + d \left( \frac{PR(P_1)}{\deg(P_1)^+} + \frac{PR(P_2)}{\deg(P_2)^+} + \dots + \frac{PR(P_n)}{\deg(P_n)^+} \right)$$

PageRank of a page  $u$  is defined as the sum of ratios of PageRank of all webpages ( $v_1, v_2, \dots, v_n$  providing backlinks to  $u$ ) to the backlink count of all such pages.

- Damping factor ' $d$ ', to take into account the probability of a user beginning a new random walk. By default  $d$  is set to 0.15
- For every page  $P_v$  providing a backlink to  $P_u$ , find the number of outlinks of  $P_v$  [ $\deg(P_v)^+$ ] and the PageRank [ $PR(P_v)$ ].
- For each  $P_v$ , find the ratio of the PageRank to the outlink count of  $P_v$ .
- Compute the sum over all such pages providing backlinks to  $P_u$ .

## PageRank Notation

Symbol	Meaning
$P_u$	A webpage ' $u$ '
$d$	Damping factor- The Probability that the user opens a new webpage to begin a new random walk
$PR(P_u)$	PageRank of the page ' $u$ '
$\deg(P_u)^-$	The number of links coming in to a page $P_u$ (in-degree of $P_u$ )
$\deg(P_u)^+$	The number of links going out of a page $P_u$ (out-degree of $P_u$ )
$N(P_u)^-$	Set of pages that point to $P_u$ (the in-neighborhood of $P_u$ )
$N(P_u)^+$	Set of pages a webpage $P_u$ points to (the out-neighborhood of $P_u$ )
$W$	A hyperlink matrix representing the network, whose entries constitute the fractional PageRank contributions
$x$	Eigen vector containing the ranks for each vertex in the network.

# Advantages of PageRank

- **Fighting spam.** A page is important if the pages pointing to it are important.
  - Since it is not easy for Web page owner to add in-links into his/her page from other *important* pages, it is thus not easy to influence PageRank.
- **PageRank is a global measure and is query independent.**
  - PageRank values of all the pages are computed and saved off-line rather than at the query time.
- **Criticism:** Query-independence. It could not distinguish between pages that are authoritative in general and pages that are authoritative on the query topic.

# Road Map

- Introduction
- Social network analysis
- Co-citation and bibliographic coupling
- PageRank
- **HITS**
- Summary

# HITS: Introduction

- Hyperlink-Induced Topic Search
- Developed by Jon Kleinberg (1999)
- “Runtime” algorithm
  - Applied only when a user submits a query
- Models linked web pages as a directed graph



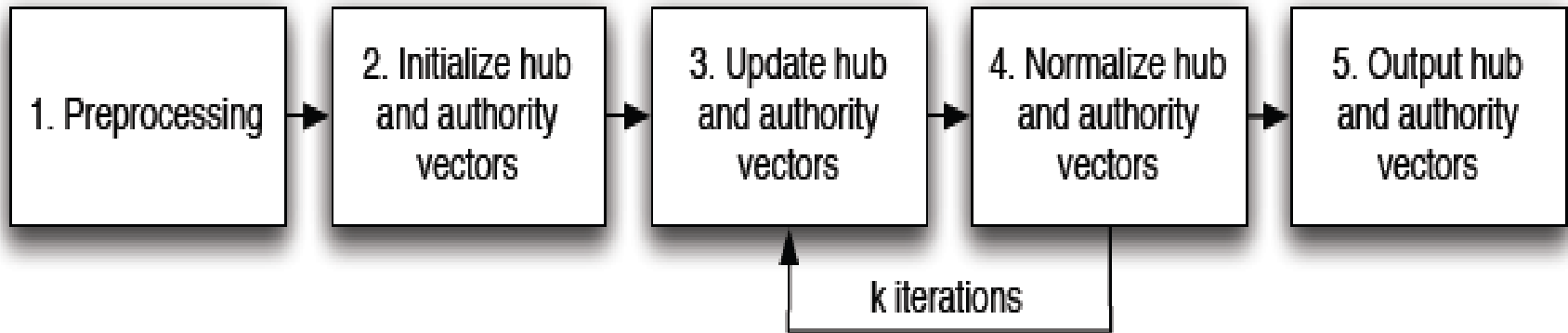
# HITS

- HITS stands for **Hypertext Induced Topic Search**.
- Unlike PageRank which is a static ranking algorithm, *HITS is search query dependent*.
- When the user issues a search query,
  - HITS first **expands** the list of relevant pages returned by a search engine and
  - then produces **two rankings** of the expanded set of pages, **authority ranking** and **hub ranking**.

# The HITS algorithm: Grab pages

- Given a broad search query,  $q$ , HITS collects a set of pages as follows:
  - It sends the query  $q$  to a search engine.
  - It then collects  $t$  ( $t = 200$  is used in the HITS paper) highest ranked pages. This set is called the **root set**  $W$ .
  - It then grows  $W$  by including any page pointed by a page in  $W$  and any page that points to a page in  $W$  (How?). This gives a larger set  $S$ , **base set**.

# HITS: Algorithm Overview



- **Inputs:**

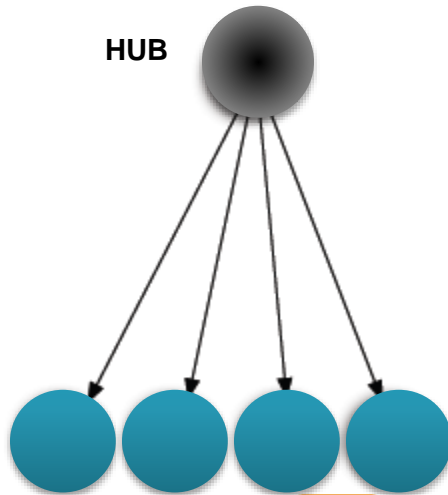
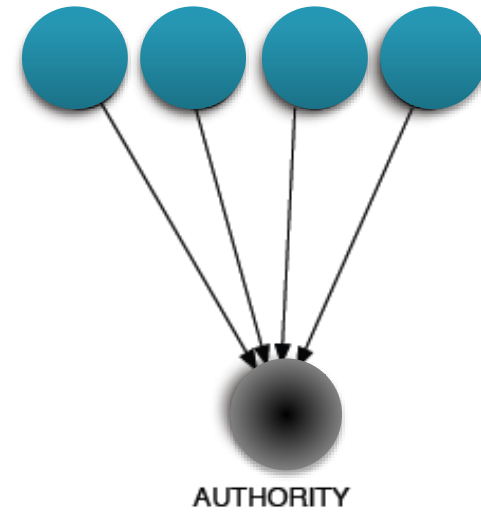
- An *adjacency matrix* representing a collection of items
- A value *k* defining the number of *iterations* to perform

- **Outputs:**

- Hub and Authority score vectors

# Authority and Hub

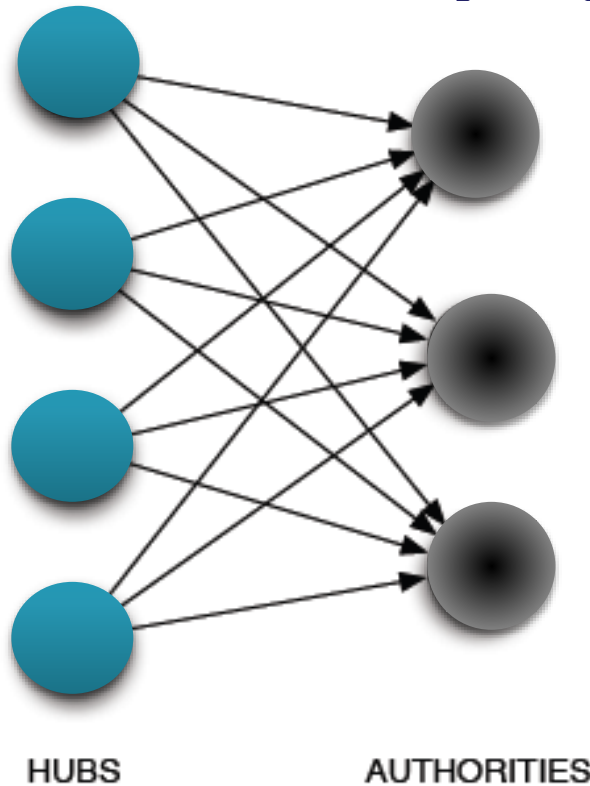
- **Authority** – A vertex is considered an authority if it has many pages linking to it (**High In-degree**)



- **Hub** – A vertex is considered a hub if it points to many other vertices (**High Out-degree**)



# Identifying the Most Relevant Pages

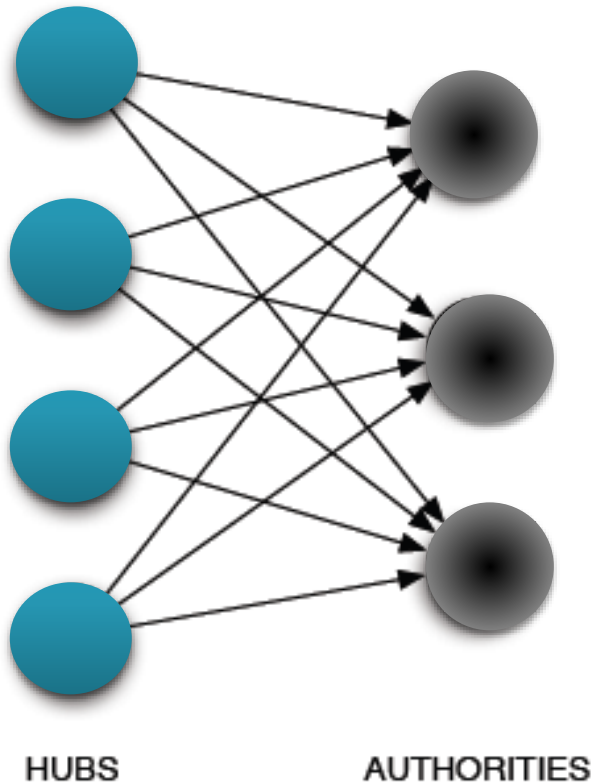


- Generally the pages considered *authoritative* on the subject are most relevant
- Finding the most relevant results is commonly found in **dense** subgraphs, primarily **bipartite graphs**

**Why authority pages are relevant?** Co-citation: If papers  $i$  and  $j$  (authority pages) are both cited by same paper  $k$  (hub), then they may be related in some sense to one another.

**Why hub pages are relevant?** Bibliographic coupling links papers that cite the same articles. if papers  $i$  and  $j$  (hub pages) both cite paper  $k$ , they may be related.

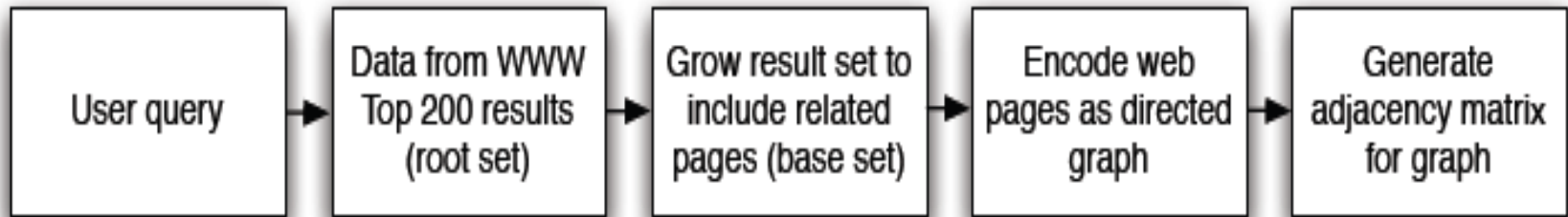
# The key idea of HITS



- A good hub points to many good authorities, and
- A good authority is pointed by many good hubs.
- Authorities and hubs have a **mutual reinforcement relationship**. Some **densely linked** authorities and hubs (a **bipartite sub-graph**) should be highly ranked.

biclique

# HITS Preprocessor



- HITS algorithm must preprocess to limit the set of web pages taken into consideration
- Root Set – Set of pages most relevant to user’s query
- Base Set – “Grown” set of pages related to query
- Encodes the adjacency matrix to be used by the algorithm

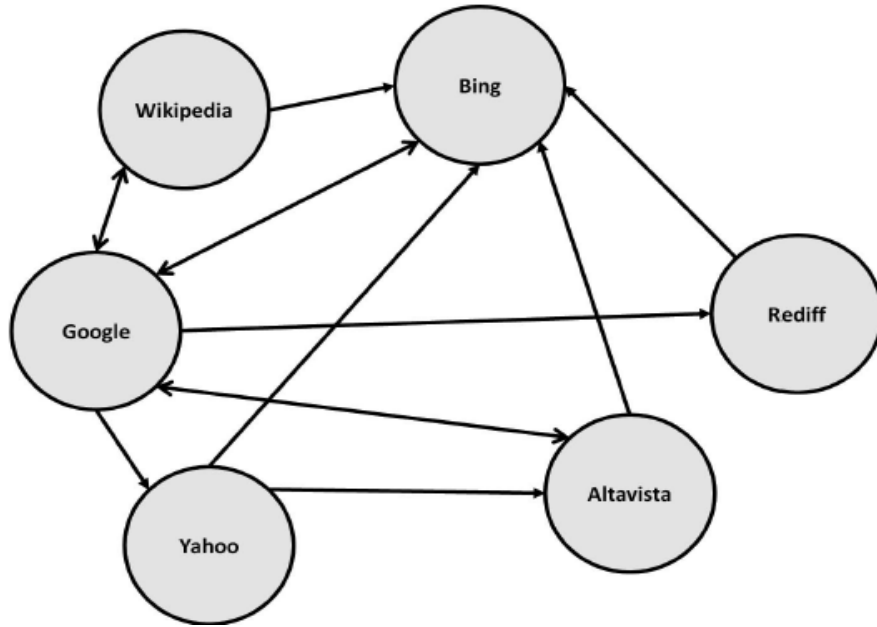
# Constructing the Adjacency Matrix

- An adjacency matrix is defined such that:

$$A_{\{ij\}} = \begin{cases} 1, & \text{if } e_{\{ij\}} \in E \\ 0, & \text{otherwise} \end{cases}$$

- For each position in the adjacency matrix:
  - Check if there is a directed edge between the 2 vertexes
  - If there is then place a 1 in that position of the matrix
  - Otherwise place a 0 in that position of the matrix

# Adjacency Matrix (Example)



■ A graph for a query, “search engine”, is displayed to the left. The adjacency matrix associated with the graph can be found below.

■  $A_{\{\text{Yahoo}, \text{Google}\}} = 0$

■  $A_{\{\text{Google}, \text{Yahoo}\}} = 1$

While there is a hyperlink from Google to Yahoo, there is not one from Yahoo to Google

	Wiki	Google	Bing	Yahoo	Altavista	Rediff
Wiki	0	1	1	0	0	0
Google	1	0	1	1	1	1
Bing	0	1	0	0	0	0
Yahoo	0	0	1	0	1	0
Altavista	0	1	1	0	0	0
Rediff	0	0	1	0	0	0

# Initialize Hub and Authority vectors

- For each web page the hub and authority scores are initially set to 1

$$\mathbf{X} = (x_1, x_2, \dots, x_n)^T$$

$$\mathbf{X}_0 = (1, 1, 1, 1, 1, 1)^T$$

- **X: Authority** Score Initialization

- For each iteration of the algorithm the hub and authority scores ( $\mathbf{Y}_1, \mathbf{X}_1$ ;  $\mathbf{Y}_2, \mathbf{X}_2$ ;  $\mathbf{Y}_3, \mathbf{X}_3$ ; ..... ) are updated

$$\mathbf{Y} = (y_1, y_2, \dots, y_n)^T$$

$$\mathbf{Y}_0 = (1, 1, 1, 1, 1, 1)^T$$

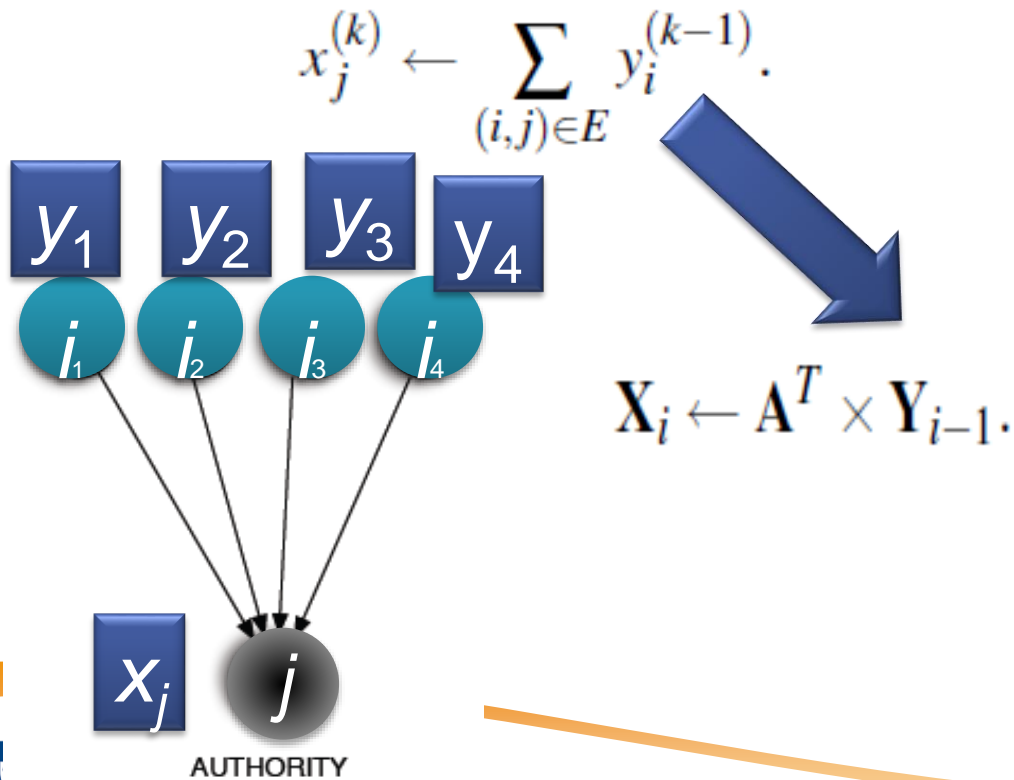
- **Y: Hub** Score Initialization

# Each Iteration: Updating Hub and Authority

(we calculate two scores for every node)

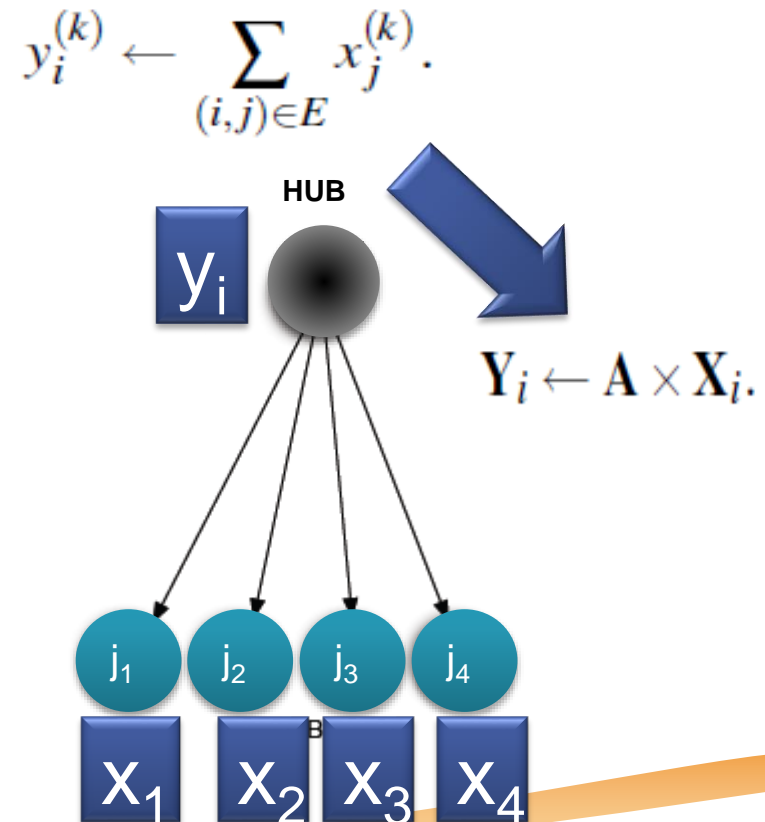
- Update **Authority** Score

- The **previous** iteration's **hub score** is used to calculate the current **authority score**



- Update **Hub** Score

- The **current** iteration's **authority score** is used to calculate the current **hub score**



# Each Iteration: Normalizing Hub and Authority

- The weights are normalized to ensure that **the sum of their squares** is 1
- The normalization process for Hub and Authority are practically identical

$$\sum_{x \in X} x^2 = 1.$$

$$x' = \frac{x}{\sqrt{\sum_{x \in X} (x^2)}},$$



$$X' = \{x' | x \in X\}.$$

- Example:

$$X = (1, 2, 3, 4)$$

$$X' = \left( \frac{1}{\sqrt{1^2 + 2^2 + 3^2 + 4^2}}, \frac{2}{\sqrt{1^2 + 2^2 + 3^2 + 4^2}}, \frac{3}{\sqrt{1^2 + 2^2 + 3^2 + 4^2}}, \frac{4}{\sqrt{1^2 + 2^2 + 3^2 + 4^2}} \right)$$

■ Normalization of Hub/Authority score



# Updating and Normalizing Authority (Example)

$$\begin{aligned}
 \mathbf{X}_1 &= \mathbf{A}^T \times \mathbf{Y}_0 \\
 &= \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}^T \times (1, 1, 1, 1, 1, 1)^T \quad \mathbf{X}'_1 = \begin{pmatrix} \frac{1}{\sqrt{1^2+3^2+5^2+1^2+2^2+1^2}} \\ \frac{3}{\sqrt{1^2+3^2+5^2+1^2+2^2+1^2}} \\ \frac{5}{\sqrt{1^2+3^2+5^2+1^2+2^2+1^2}} \\ \frac{1}{\sqrt{1^2+3^2+5^2+1^2+2^2+1^2}} \\ \frac{2}{\sqrt{1^2+3^2+5^2+1^2+2^2+1^2}} \\ \frac{1}{\sqrt{1^2+3^2+5^2+1^2+2^2+1^2}} \end{pmatrix} \\
 &= \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} \times \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad = \begin{pmatrix} \frac{1}{\sqrt{41}} \\ \frac{3}{\sqrt{41}} \\ \frac{5}{\sqrt{41}} \\ \frac{1}{\sqrt{41}} \\ \frac{2}{\sqrt{41}} \\ \frac{1}{\sqrt{41}} \end{pmatrix} \\
 &= \begin{pmatrix} 1 \\ 3 \\ 5 \\ 1 \\ 2 \\ 1 \end{pmatrix} \quad = \begin{pmatrix} 0.15617 \\ 0.46852 \\ 0.78087 \\ 0.15617 \\ 0.312348 \\ 0.15617 \end{pmatrix}
 \end{aligned}$$

# Convergence of HITS

- There is no formal convergence criteria
- Generally the upper bound for  $k$  is 20

Even after just 6 iterations of the “search engine” example the HITS algorithm on **Authority Score** you can begin to see convergence.

Iteration	Wiki	Google	Bing	Yahoo	Altavista	Rediff
0	1	1	1	1	1	1
1	0.156	0.469	0.781	0.156	0.312	0.156
2	0.204	0.388	0.777	0.204	0.347	0.204
3	0.224	0.350	0.769	0.224	0.369	0.224
4	0.232	0.332	0.765	0.232	0.378	0.232
5	0.236	0.324	0.762	0.236	0.383	0.236
6	0.238	0.320	0.761	0.238	0.385	0.238

Changes  
are very  
small



# Pseudocode

No need to build  
a bipartite graph

**Input:**  $A$ : an adjacency matrix representing a collection of items (e.g. web pages)  
**Input:**  $k$ : a natural number (number of iterations)

**Output:**  $X_k, Y_k$ : vectors of hub and authority scores for each vertex in the graph

1:  $X_0 \leftarrow (1, 1, 1, \dots, 1) \in \mathbb{R}^n$

2:  $Y_0 \leftarrow (1, 1, 1, \dots, 1) \in \mathbb{R}^n$

3: for  $i = 1$  to  $k$  do

4: Apply the  $\mathcal{J}$  operation to  $(X_{i-1}, Y_{i-1})$ , to obtain new authority scores,  $X'_i$ .

5: Apply the  $\mathcal{O}$  operation to  $(X'_i, Y_{i-1})$ , to obtain new hub scores,  $Y'_i$ .

6: Normalize  $X_i$ , obtaining  $X'_i$ .

7: Normalize  $Y_i$ , obtaining  $Y'_i$ .

8: end for

9: return  $(X_k, Y_k)$ .

$$X_i \leftarrow A^T \times Y_{i-1}$$

Step 6 should be run in step 4  
Step 7 should be run in step 5

$$Y_i \leftarrow A \times X_i$$

# Strengths

- Two vectors (hub and authority) allow application to decide which vector is most interesting
- Algorithm is efficient
- Its ability to rank pages according to the query topic, which may be able to provide *more relevant* authority and hub pages.

# Weaknesses

- “Topic Drift”: some pages in the expanded set may not be on topic.
- It is easily spammed. Manipulation of algorithm through “spam”. It is in fact quite easy to influence HITS since adding out-links in one’s own page is so easy.
- Inefficiency at query time: The query time evaluation is slow. Collecting the root set, expanding it are all expensive operations

# Road map

- Introduction
- Social network analysis
- Co-citation and bibliographic coupling
- PageRank
- HITS
- **Summary**

# Summary

- We have introduced
  - Social network analysis, centrality and prestige
  - Co-citation and bibliographic coupling
  - PageRank, which powers Google
  - HITS
- Yahoo! and MSN have their own link-based algorithms as well, but not published.
- **Important to note:** Hyperlink based ranking is not the only algorithm used in search engines. In fact, it is combined with many **content based factors** to produce the final ranking presented to the user.

# Summary

- Links can also be used to find **communities**, which are groups of content-creators or people sharing some common interests.
  - Web communities
  - Email communities
- Focused crawling: combining contents and links to crawl Web pages of a specific topic.
  - Follow links and
  - Use learning/classification to determine whether a page is on topic.



# ■ References

1. <http://www.cs.uic.edu/~liub/>
2. Nagiza F. Samatova, William Hendrix,  
John Jenkins, Kanchana  
Padmanabhan, Arpan Chakraborty,  
Practical Graph Mining With R