INGENIERÍA INDUSTRIAL
UNIVERSIDAD DE CHILE

FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Chapter 4: Web Structure Mining

Professors: Juan D. Velásquez
Gaspar Pizarro V.

http://wi.dii.uchile.cl/
@juandvelasquez

Outline

- Introduction:
  - Web Structure Mining (WSM)
  - Examples & Applications
  - The business
- Web Crawling
- The PageRank Algorithm
- The Hits Algorithm
- Identifying Web Communities
- Further topics on WSM

Section 4.1

Introduction

Web Structure Mining

- It deals with the **mining of the web hyperlink structure** (inter document structure).
- A website is represented by a **graph of its links**, **within the site or between sites**.
- Facts like the popularity of a web page can be studied, for instance, if a **page is referred by a lot of other pages** in the web.
- The web link structure allows to develop a notion of hyperlinked communities.
- It can be used by **search engines**, like Google or Yahoo!, in order to get the **set of pages more cited for a particular subject**.

WSM (2)

- To **discover the link structure** of the hyperlinks
  - At the **inter-document level**
  - To generate **structural summary** about the Website and Web page.
- **Direction 1**
  - **based on the hyperlinks**, **categorizing the Web pages** and generated information.
- **Direction 2**
  - discovering the **structure of Web document itself**.
- **Direction 3**
  - discovering the nature of the hierarchy or network of hyperlinks in the Website of a **particular domain**.

WSM (3)

- Finding authoritative Web pages
  - Retrieving pages that are not only relevant, but also of high quality, or **authoritative on the topic**
- Hyperlinks can infer the **notion of authority**
  - The Web consists not only of pages, but also of **hyperlinks pointing** from one page to another
  - These hyperlinks contain an enormous amount of **latent human annotation**
  - A hyperlink pointing to another Web page, this can be considered as the author's **endorsement** of the other page

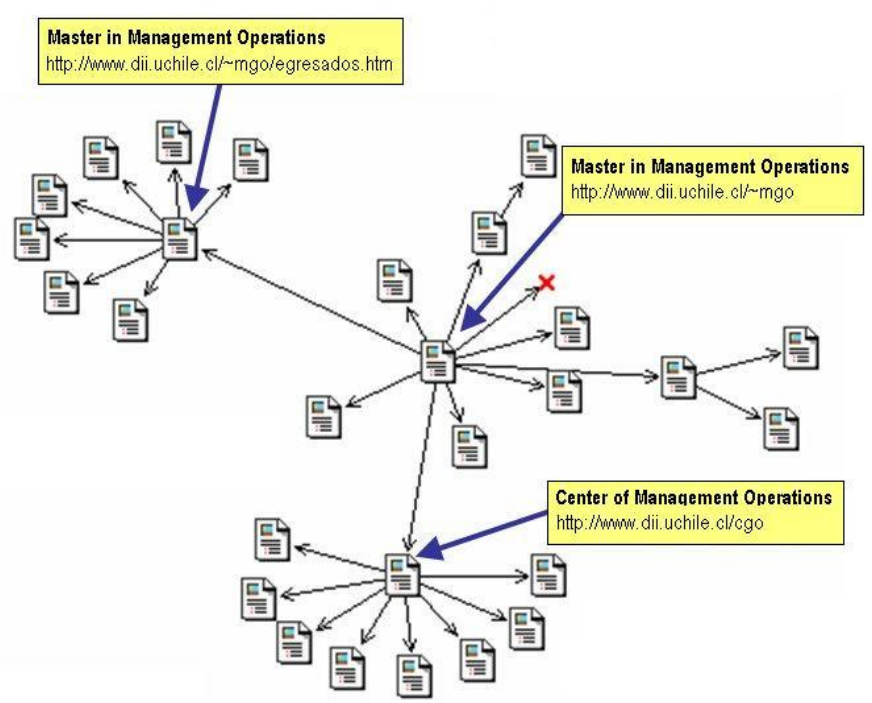
WSM (4)

- Web pages categorization
  - (Chakrabarti, et al., 1998)
- Discovering micro communities on the Web
  - Example: Clever system
    - (Chakrabarti, et al., 1999), Google (Brin and Page, 1998)

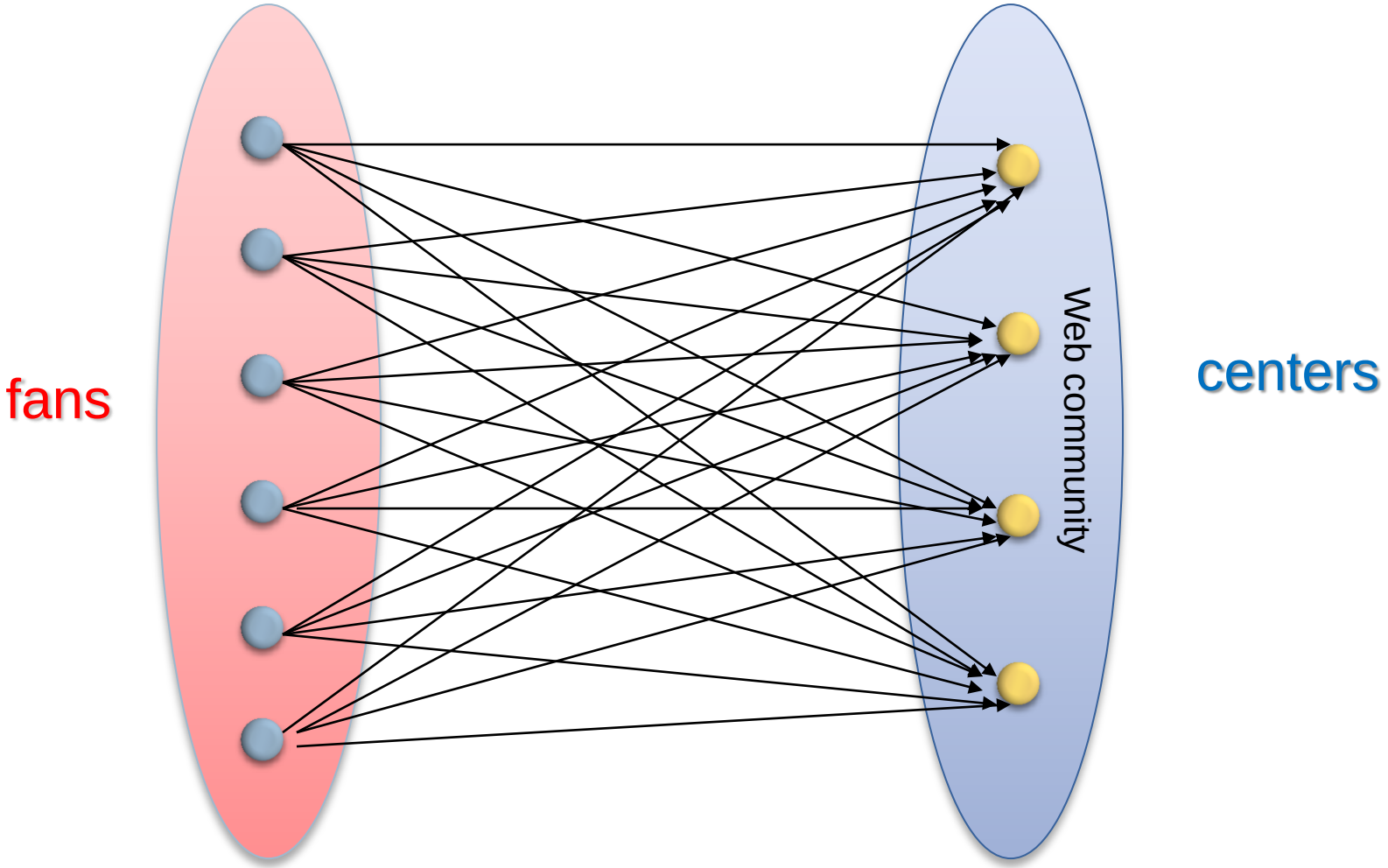
WSM (5)
THE MATHEMATICAL FRAMEWORK

- Graph theory
- We define the web as an **directed graph**
- $P=\{p_i\}$: Set of Pages
- $L=PxP$: Set of links
- Adjacency Matrix M_{ik}
- Stochastic Matrices

$$(H_{ik} ; \Sigma_k H_{ik}=1)$$



WSM: Example

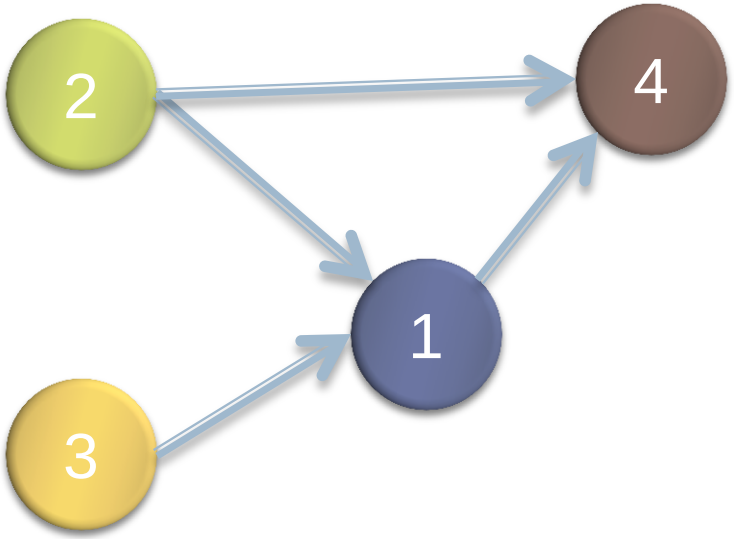


Web Community Centers: many web pages go there

HITS (Kleinberg99), PageRank (Google, Bring and Page98)

- **Assumptions:**
  - **Credible sources** will mostly **point to credible sources**
  - Names of hyperlinks suggest meaning
  - Ranking is a function of the ***query terms*** *and of the* ***hyperlink structure***
- An example of why this makes sense:
  - The official Lord of the Rings site will be linked to by most high-quality sites about movies, Lord of the Rings, etc.
  - The biggest LoTR fan clubs probably are also frequently linked
  - A *spammer* who adds "Lord of the Rings" to his/her web site *probably* won't have many links to it

A simple web graph for a web community



Let $P = \{p_1, \ldots, p_n\}$ be the **set of pages in the Web community** (in the example n=4)

The e-business of structure mining

- Google sessions
  - $5{,}28*10^8$ Visitor/month on March 2007.
- Google revenue
  - 3,66 Billions US$ quarter March 31 2007. An increase of 63% in comparison to March 2006.
- The business
  - **“Advertising Sales (Ad)”** and **“user click on them”** on the right side of the search result. They receive **$ for each click** on them.

Section 4.2

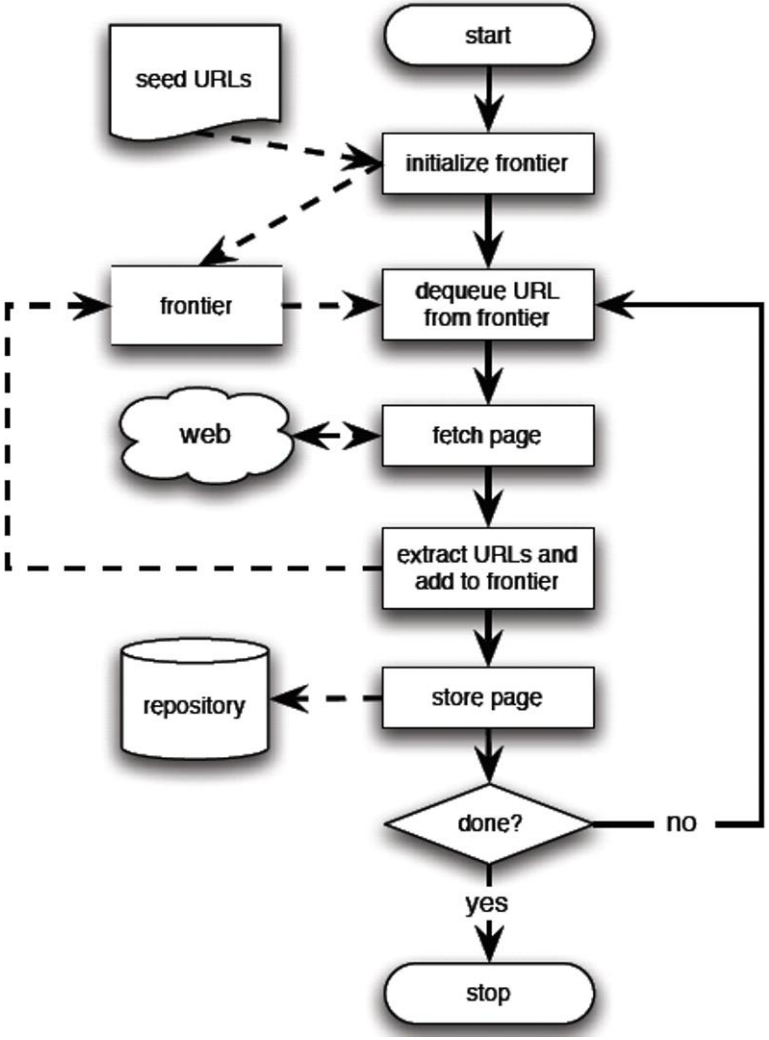
Web Crawling

Or How to Build Google

The Crawler

- A program also called **Robot** or **Spider.**
  - Decide which "kind of page to retrieve".
  - Decide which "kind of visit to page" has to followed (depth first, breath first).
  - Usually very **distributed computing** modules.
  - Very **memory consuming**
- The web is dynamic
  - **¿When to crawl, in order to maintain accurate information?**

The Crawler: Basic algorithm



- Needs a set of seed URLs

Exploring the web graph

- Breadth-first strategy
  - Implemented with a FIFO frontier queue
  - The “default” strategy
  - Converges to popular pages
- Best-first crawler
  - Implemented with a priority queue
  - Priority defined by application

Stages: Fetching

- Crawler sends GET request to server, server responds
  - It needs to be ready for anything: Slow servers, redirects, huge pages

Stages: Parsing

- Server responds with code 200 and sends HTML
- Parser parses HTML to a computer-manageable structure
  - HTML is dirty, parser must be robust

Stages: Link extraction

- So we have (clean) HTML tree
- Links are extracted
- Links must be normalized
  - Set host part to lowercase
  - Turn relative URLs into absolute
  - Remove default port
  - Etc.
- All of this in order to avoid crawling the same page from two different urls

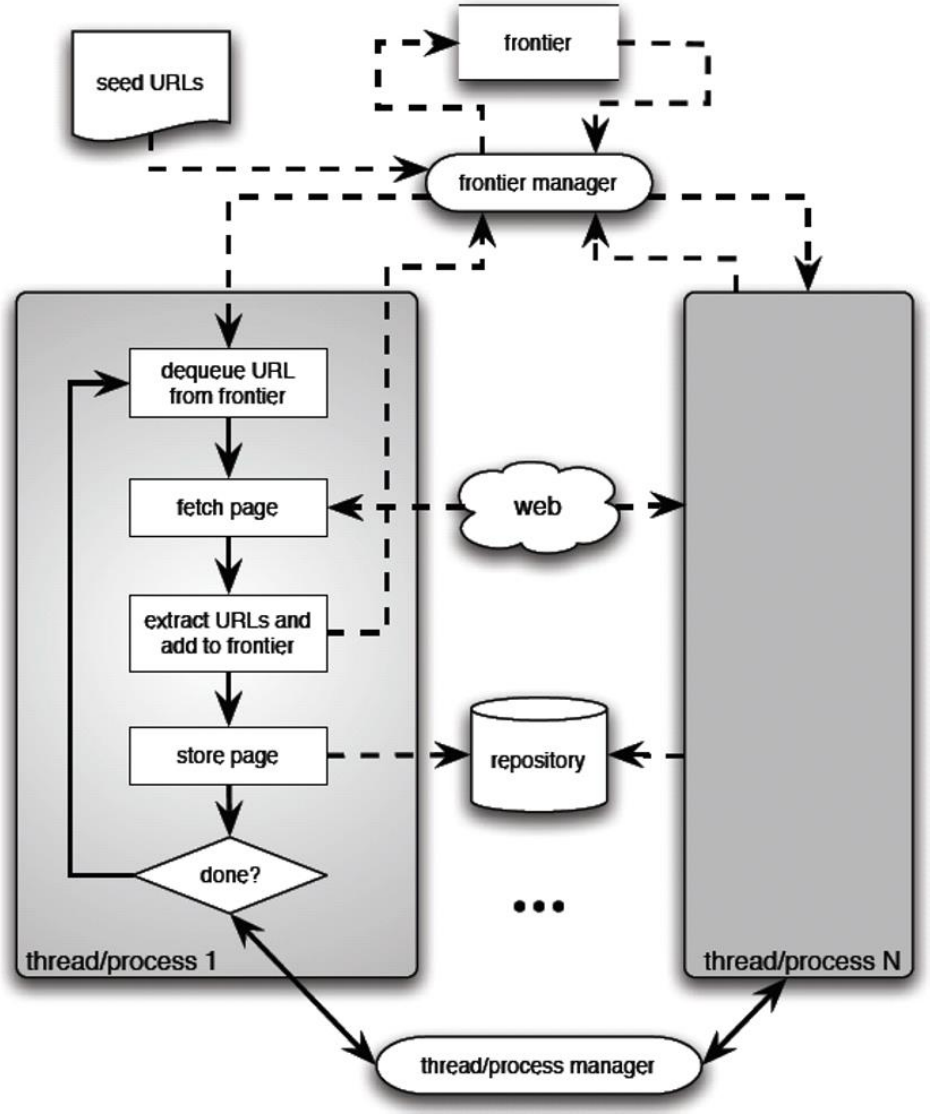
Stages: Storage

- Pages are stored
- Naïve strategy: On disk, one file per page
  - Big overhead
  - What can you do with a bunch of files?
- Better strategy: On database
  - Less overhead
  - More “minable”, with an inverted index

Performance: Still not Google

- First algorithm can be easily parallelized
  - The basic algorithm can be run on many threads
- Easily, but not immediate
  - There are points of communication that must be managed

Performance: Still not Google



But, still not Google

Spider traps

- Sites (intentionally or not) can trick crawlers to crawl a website infinitely
  - Infinitely deep url paths
  - Same page with different url
- If crawler falls in trap everybody lose
  - Crawler does not advance
  - Site wastes bandwidth and processing time

Crawler Ethics

- A crawler is a kind of “user” of a site
- It uses network bandwidth
- Misuse of a crawler can make a denial-of-service attack to a server, resulting in the site being unavailable to the users (the real users)

Crawler Ethics: Measures for politeness

- Throttle crawler requests for a given domain
  - Wait a fixed time between requests, let's say 15 seconds
  - Wait a time dependent on the response time, let's say 3 times the site response time
- Disclose the nature of the crawler
  - Set the User-Agent header
- Follow the Robot Exclusion Protocol
  - Defines who crawls and where to crawl in the site
  - In /robots.txt

Crawler Ethics: Crawler vs Site

- Crawler “unethical” measures
  - Change User-Agent to one of a browser
  - Randomize request rate in order to look “human”
- Server IP ban
  - Server can ban the IP of the crawler
  - Sometimes IPs can be changed
  - Sometimes users share IPs
- The CAPTCHA
  - The ultimate crawler ban
  - Hampers site usability

Crawler Ethics: Crawler vs Site

- Site “unethical” measures
  - Detect crawler, show different text
  - Can be detected and “banned” by a search engine

Section 4.3

The Page Rank Algorithm

The Page Rank Algorithm

- **Idea**
  - An Hyperlink that point to me works like a **"recommendation"** from other page.
  - The more **recommendations**, the more **"important"** is my page.
  - The more important my **recommenders**, the more **"important"** is my page.
- **Rank**:
  - A **numeric value** to represent the importance of a page.
- We are not inspecting the content

The mathematical foundation of PageRank

- $G = (V, E)$
- $V = \{1, \cdots, n\}$
- $E \in V \times V$

- Pagerank of node i is dependent on the pagerank of its recommenders and their out-degrees

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

- O_j: Out-degree of node j

The mathematical foundation of PageRank

- This can be expressed in matricial form as

$$P = A^T P$$

where

$$A_{ij} = \begin{cases} \frac{1}{O_j}, & (i, j) \in E \\ 0, & \text{otherwise} \end{cases}$$

Thus, A is the adjacency matrix of G.

The mathematical foundation of PageRank

- We can use A as a transition matrix of a Markov Chain where
  - Webpages are nodes
  - Hyperlinks as edges

- Imagine a random surfer that follows links and, for a page, picks randomly one out-link to follow.

- Pagerank would be the stationary distribution of the page probabilities.

The mathematical foundation of PageRank

- By construction
- Problem: In order to assure convergence to a stationary probability, A must be:
  - Irreducible -> G is strongly connected
  - Aperiodic -> There are no «obligatory» cycles in the graph
- We have to complement A in order to get a stationary distribution

PageRank key definition

- The random surfer model is modified
- With probability d, random surfer follows an out-link (chosen randomly between them)
- With probability 1-d random surfer resets and picks a random page (any page)

PageRank key definition

- Original definition

$$\mathrm{P} = \mathrm{A}^{\mathrm{T}}\mathrm{P}$$

- Pagerank'd defintion

$$\mathrm{P} = \left(\mathrm{d}\overline{\mathrm{A}}^{\mathrm{T}} + (1-\mathrm{d})\frac{\mathrm{E}}{\mathrm{n}}\right)\mathrm{P}$$

where

- d: Damping factor
- $\overline{A}$: Normalized adjacency matrix, where sink nodes are connected to all nodes
- E: Matrix of ones

PageRank key definition

$$P = \left(d\overline{A}^{T} + (1-d)\frac{E}{n} \right) P$$

$$P = G \cdot P$$

$$G_{ij} = \begin{cases} \dfrac{(1-d)}{n} + dA_{ji}, & [A]_j \neq \vec{0} \\ \dfrac{1}{n}, & [A]_j = \vec{0} \end{cases}$$

PageRank key definition

- So now we can replicate the pagerank basic idea with this equation

$$P(i) = (1 - d) + d \sum_{j=1}^{n} A_{ji} P(j)$$

PageRank Algorithm

- Intuitive Justification:
  - Thought as a **model of user behavior**.
  - **Random surfer** that keeps clicking on links **starting from one random page**, and **some time get bored** and **continue** from other random page.
  - **Page Rank** value correspond to the **probability that the random surfer visit the page**.
    - The stationary probability.
  - Then the **d parameter** correspond to the ***probability to start again from another page***.
- Other:
  - Based on **Academic citation** literature.

PageRank Algorithm

1. Initialize

$$p_o = [\frac{1}{n}, \frac{1}{n}, \cdots]$$

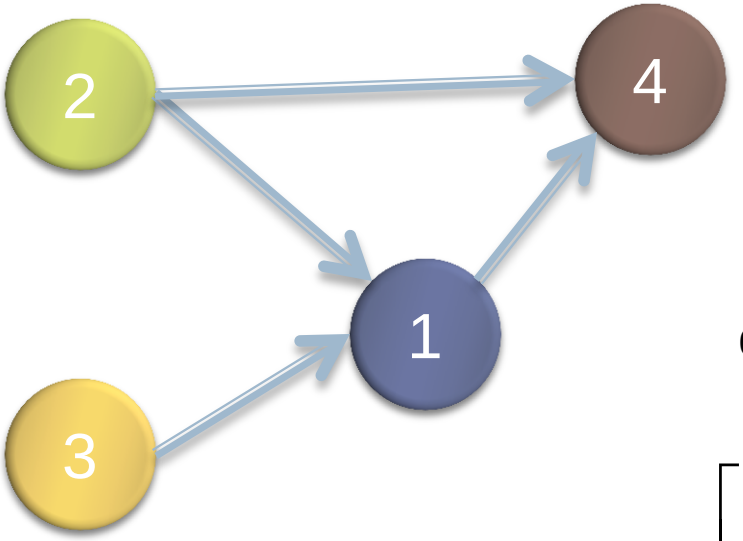
2. Set d, usually d = 0,85.

3. Repeat

$$p_{k+1} = G p_k$$

3. Until $\|p_k - p_{k-1}\| < \varepsilon$

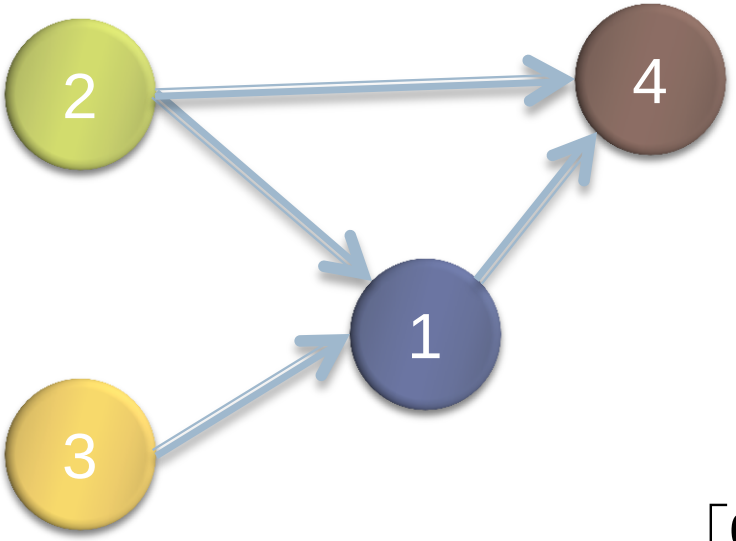
PageRank Algorithm



d = 0.9

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0.5 & 0 & 0 & 0.5 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad p_0 = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$$

PageRank Algorithm

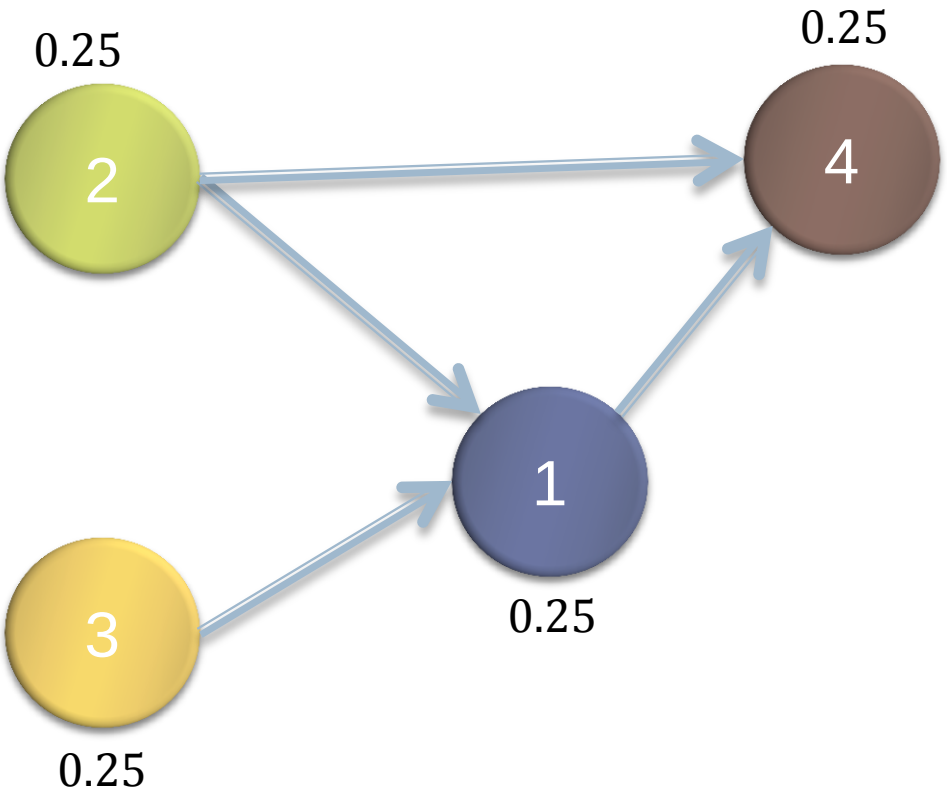


d = 0.9

$$G = \begin{bmatrix} 0.025 & 0.475 & 0.925 & 0.25 \\ 0.025 & 0.025 & 0.025 & 0.25 \\ 0.025 & 0.025 & 0.025 & 0.25 \\ 0.025 & 0.475 & 0.025 & 0.25 \end{bmatrix} \qquad p_0 = \begin{pmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{pmatrix}$$

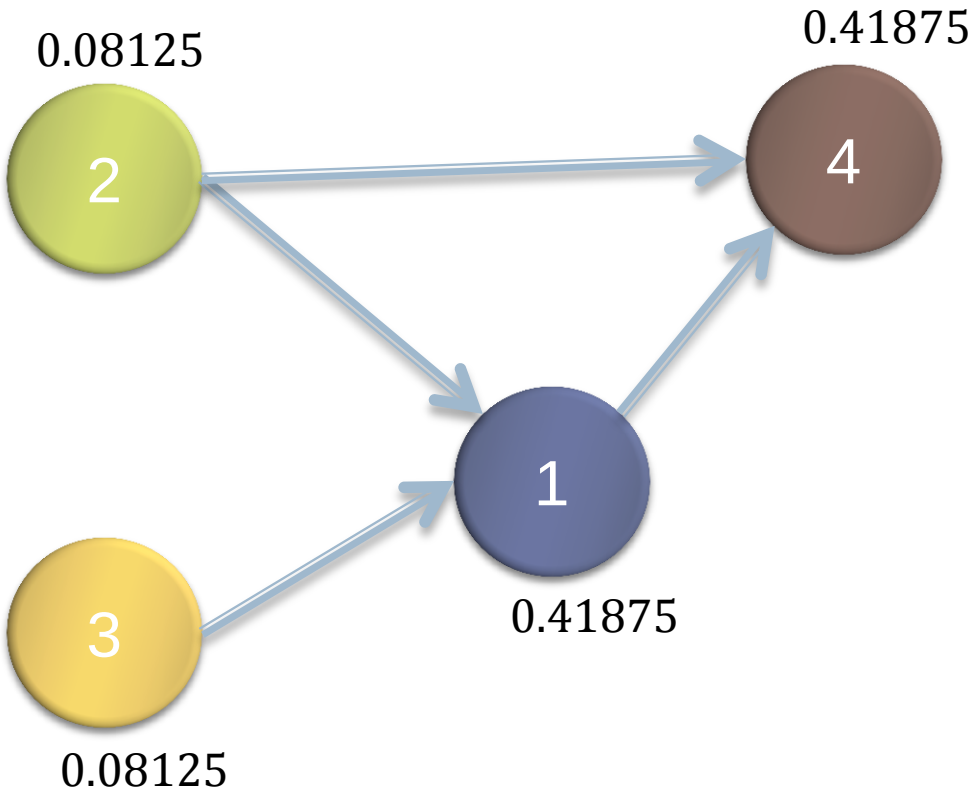
PageRank Algorithm

Start



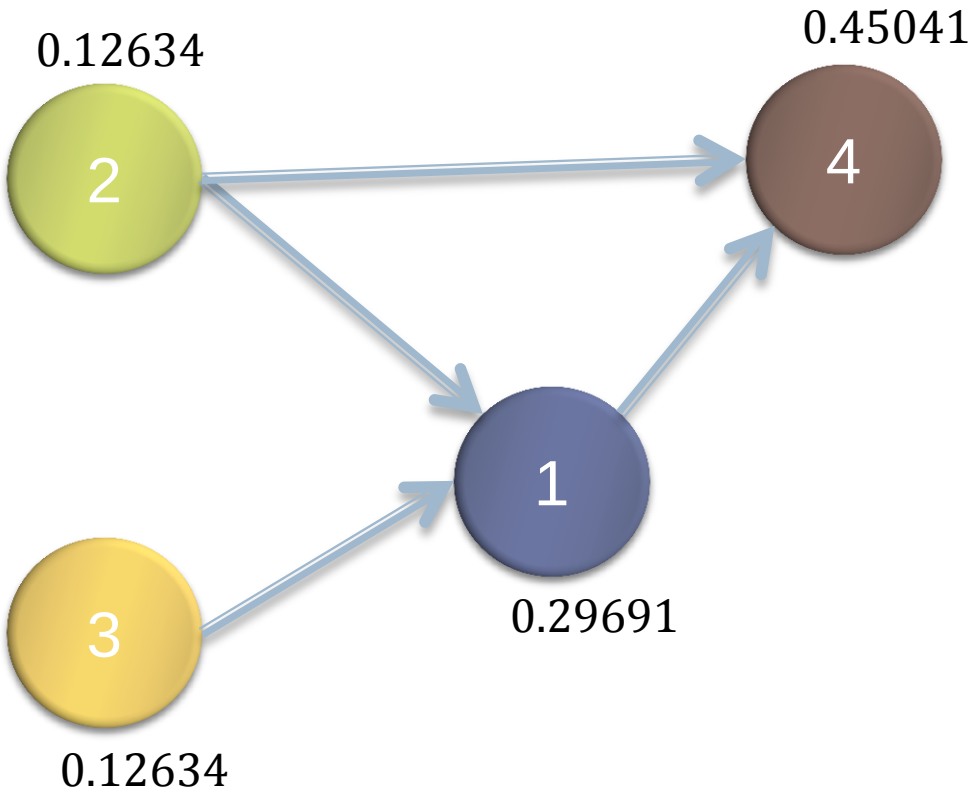
PageRank Algorithm

First iteration



PageRank Algorithm

After 100 iterations



PageRank Algorithm

- **Strengths**
  - Global measure: Every page can be page-ranked
  - Offline computation: Crawl everything, then page-rank everything
  - Hard to spam: A spammer has to have a link farm to distort the PageRank considerably
- **Weaknesses**
  - Global measure: Every page is page-ranked against all. So importance is distorted inside communities
  - Time-invariant: PageRank is biased towards old pages

PageRank Algorithm: An improvement

- **Timed Pagerank**
  - Time is considered by making the damping factor d a function of the last time a page was updated, so that newer pages have a smaller probability of random jump

PageRank Algorithm

- **Not the real Google algorithm**
  - It is a very **carefully hidden secret**.
- The original PageRank doesn't consider text content (**keywords**) on the page.
  - But the real one **does**.
  - The real algorithm also considers **n-grams** (list of words)
- The real algorithm also considers user behavior.
  - They capture it with:
    - Click on links
    - Google toolbar
    - Google web-accelerator (a Google proxy)
    - Gmail and YouTube

Google: the world's largest matrix computation

- N = 8 Billions of pages
- H matrix very sparse of NxN
- **Overnight** computations
- When the calculation are restarted over a site. They only look at the restricted index and doesn't calculate again the values.
- **50 Terabyte on page result**
  - 4,9 Gb per second
- 8000 server only for
  - www.google.* pages
- 70000 server for
  - data storage and proccesing
- 425 Mega Watt hour per day

Section 4.4

The HITS Algorithm

The HITS Algorithm

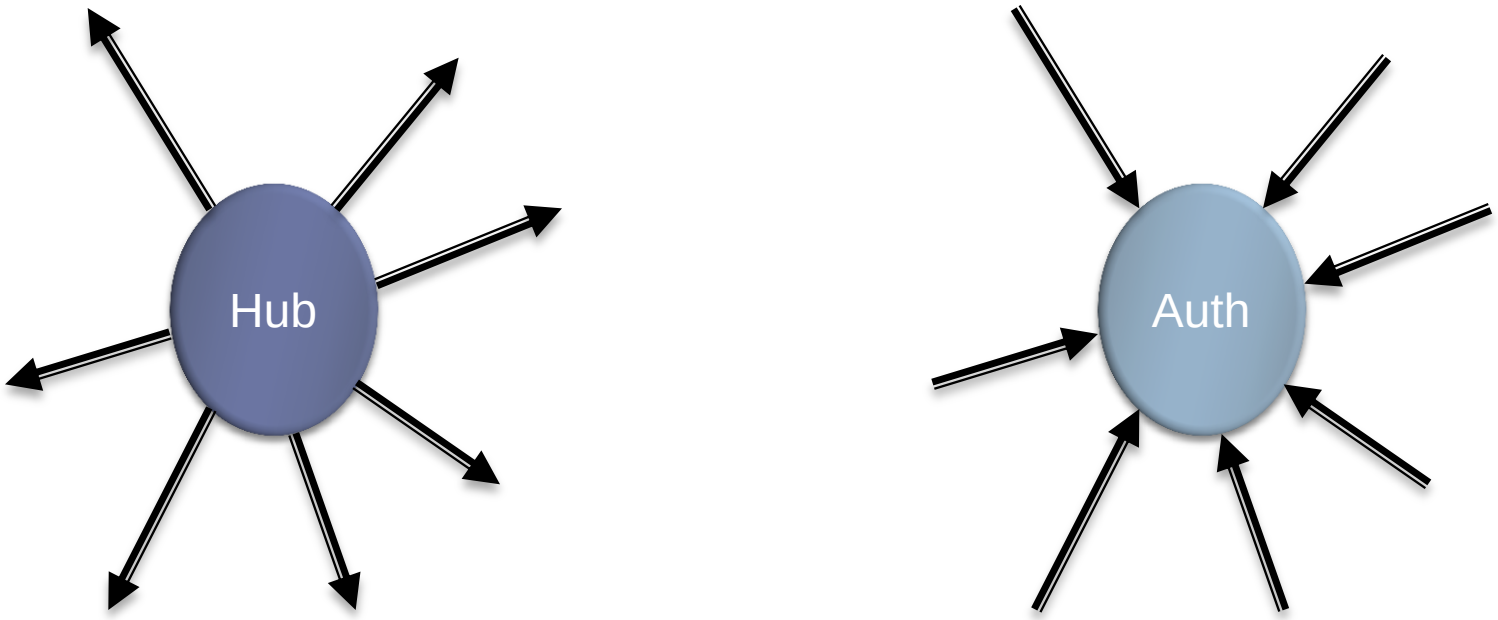
- The web page ranking is given by
  - **“linked” popularity**.
- Teoma (http://www.teoma.com) and Ask (http://es.ask.com) use it.
- Proposed by John Kleinberg in1998

Assumptions

- A **credible page** will **point to credible pages**.
- **Credible pages** are **pointed by others**.

The page ranking depends on the user query and the hyperlink structure that follows from paths of the most credible pages.

Hubs and Authority Pages



HITS Algorithm

- Assumptions
  - The **authority level (or rank) came from in-edges**.
  - A **good authority come from good hubs** and **a good hub contains links that point to good authorities**.
- A **simple method** to differentiate the page's relevance is
  - First **assigning non-negative weights**, depending if the page is hub or authoritative. Well, finally the page have both of them.
  - Next, the weights are **adjusted by an iterative process** and the **relative page's importance** in the community is calculated.

HITS Algorithm (2)

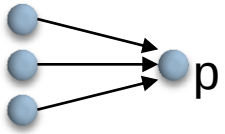
- Identifying **which pages contain more relevant information** that others [Kleinberg99]:
  - **Authorities:** A natural information repository for the community, because are pointed by many Hub.
  - **Hubs:** These concentrate links to authorities web pages, for instance, **"my favorite sites".** They point to authorities.
- ***If q*** *it is good hub* ***Then q*** *point to many good authority pages p.*
- ***If p*** *it is a good authority* ***Then p*** *is pointed by many good hub pages q.*
- *We can give a measure of the quality of "goodness" for authority and hubs.* We call it:
  - **Authority and hub weight (a_p,h_q).**

HITS Algorithm (3)

- Let a_p and h_p be the weights associate to authority and hub pages, with $p \in P$.
- We relate these weight to each other by:

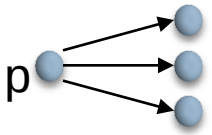
$$a_p = \sum_{\forall q, p \in P / q \to p} h_q$$

The **measure of the authority p** is the Sum of all hub measure **pointing** to p.



$$h_p = \sum_{\forall q, p \in P / p \to q} a_q$$

The **measure of the hub** is the Sum of all authority weight **pointed to by** p



We can consider normalized

$$1 = \sum_{\forall q \in P} h_q \qquad 1 = \sum_{\forall q \in P} a_q$$

HITS Algorithm (4)

- First consider the query and then build the related graph
- Let $A = (a_1, ..., a_n)$ and $H = (h_1, ..., h_n)$ be the vectors with the weights for authorities and hubs pages in the community.
- Then $A = M^T H \quad H = MA$
- With: $M = (m_{ij}) = \begin{cases} 1 & i \to j \\ 0 & otherwise \end{cases}$

$$A^{(k+1)} \leftarrow M^T H^{(k)} = (M^T M) A^{(k)}$$

$$H^{(k+1)} \leftarrow MA^{(k)} = (MM^T) H^{(k)}$$

HITS Algorithm (5)

1. Initialize $A = (1,\ldots,1), H = (1,\ldots,1)$

2. Calculate

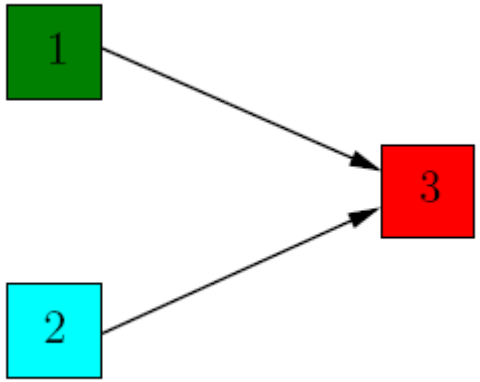
$$A^{(k+1)} \leftarrow M^T H^{(k)} = (M^T M) A^{(k)}$$

$$H^{(k+1)} \leftarrow M A^{(k)} = (M M^T) H^{(k)}$$

3. Normalize A and H.

4. If $A^{(k+1)} \approx A^{(k)}, H^{(k+1)} \approx H^{(k)}$, stop.
   Else $A^{(k)} = A^{(k+1)}, H^{(k)} = H^{(k+1)}$, go to point 2.

HITS Algorithm (6)



$$M = ?$$

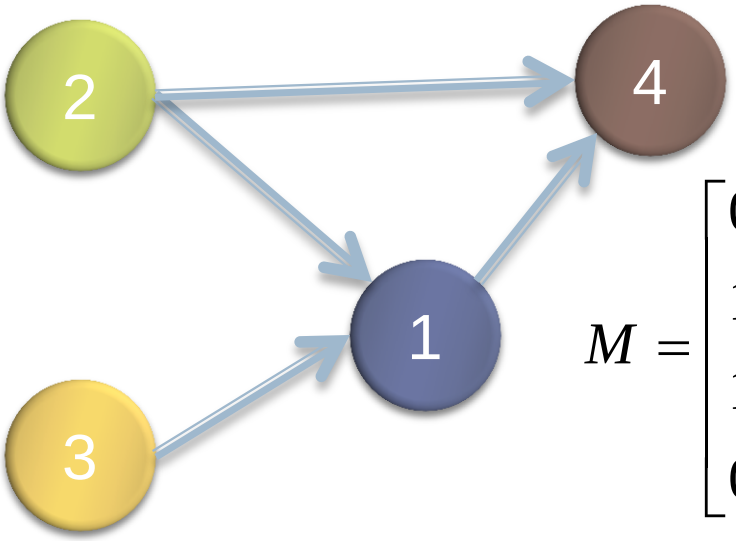
$$M^t = ?$$

$$A^1 = H^1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix},$$

$$M^t M = ?$$

$$M M^t = ?$$

HITS Algorithm (7)



$$M = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad and \quad M^T = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

$$M^T M = \begin{bmatrix} 2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 2 \end{bmatrix} \quad and \quad MM^T = \begin{bmatrix} 1 & 2 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

HITS Algorithm (8)

$$A^{(0)} = H^{(0)} = (1,1,1,1)$$

$$A^{(1)} = \begin{pmatrix} 0.5 \\ 0 \\ 0 \\ 0.5 \end{pmatrix}, A^{(2)} = \begin{pmatrix} 0.5 \\ 0 \\ 0 \\ 0.5 \end{pmatrix}, A^{(3)} = \begin{pmatrix} 0.5 \\ 0 \\ 0 \\ 0.5 \end{pmatrix}$$

$$H^{(1)} = \begin{pmatrix} 0.33 \\ 0.44 \\ 0.22 \\ 0 \end{pmatrix}, H^{(2)} = \begin{pmatrix} 0.37 \\ 0.43 \\ 0.2 \\ 0 \end{pmatrix}, H^{(3)} = \begin{pmatrix} 0.37 \\ 0.43 \\ 0.19 \\ 0 \end{pmatrix}$$

In general, the algorithm will converge in few iterations (around five [Chakrabarti98]).

HITS Algorithm (10)

- The result are **2 rankings for each page**:
  - The most auth pages and the most hub pages.
- The algorithm does not converge always
  - We need to have an **irreducible** and **aperiodic** matrix
  - The **original algorithm** consider to select **depending on the query** a suitable graph.
- A **better implementation algorithm** is available in the same way that PageRank does and **does not depend on the query**

$$A^{(k+1)} = ((1-d)D + dM^{\mathrm{T}}M)A^{(k)}$$

$$H^{(k+1)} = ((1-d)D + dMM^{\mathrm{T}})H^{(k)}$$

Hits: Bibliometrics Connection

- **Bibliometrics Concept**:
  - **co-citation**:
    - two document are both cited by the same third document.
  - **co-reference**:
    - two document both refer to the same third document.

- $M^tM=D_{in}+C_{cit}$
  - C_{cit} the co-citation value
  - D_{in} diagonal matrix with in-degree of each node.

- $MM^t=D_{out}+C_{ref}$
  - C_{ref} the co-reference value
  - D_{out} diagonal matrix with out-degree of each node.

Hits: evaluation

- Advantages
  - Double ranking
  - Bibliographic utility
- Disadvantages
  - Original algorithm was **query dependent**
    - but not the **new one**
  - **Spam sensibility**
    - more than page rank

Other Algorithms

- SALSA [2000 Lempel & Moran]:
  - **Based on HITS** but instead of matrices M^tM and M^tM they use a **stochastic matrix U and V** that are constructed in the same way that **page rank**.
    - (**too expensive in data preparation**)
- HotRank [2003 Tomlin]:
  - Based on an **Optimization Problem**.
    - Minimizing the entropy of the transition probabilities restricted that the total probability flux in each nodes is conserved.
    - Then he solve the stationary probability.
      - (**too expensive in data preparation**)

Section 4.5

Identifying Web Communities

Network Theory

- **World Wide Web** and *hyperlink structure*
- **The Internet** and *router connectivity*
- Collaborations among…
  - Movie actors
  - Scientists and mathematicians
- Sexual interaction
- Cellular networks in biology
- Food webs in ecology
- Phone call patterns
- Word **co-occurrence** in text
- Neural network connectivity of flatworms
- Conformational states in **protein folding**

Social Networks & Communities

- People are represented as **nodes**
  - **edges are relationship** between them.
- Acquaintanceship, friendship, co-authorship, etc.
- Very early studies related to the sociology research field.
  - **Analogue context** can be found in the web hyperlink structure.
  - **Web pages communities** are **cluster of pages** that points to it.

Web Applications of Social Networks

- Analyzing page importance
  - Page Rank
    - Related to recursive in-degree computation
  - Authorities/Hubs
- Discovering Communities
  - Finding near-cliques
- Analyzing Trust
  - Propagating Trust
  - Using **propagated trust to fight spam**
    - In Email
    - In Web page ranking

Identifying web communities

- [Flake02]
  - "Set of sites that have more links (in either direction) to members of the community than non-members"
- [Staab05].
  - The web community identification have several practical applications, like:
    - focalized search engine
    - content filters
    - complement of text-based searches
- [Kumar02,Mika04]
  - The most important continue being the analysis of the entire Web for studying the relationship within and between communities, like scientific, research and in general Social Networks

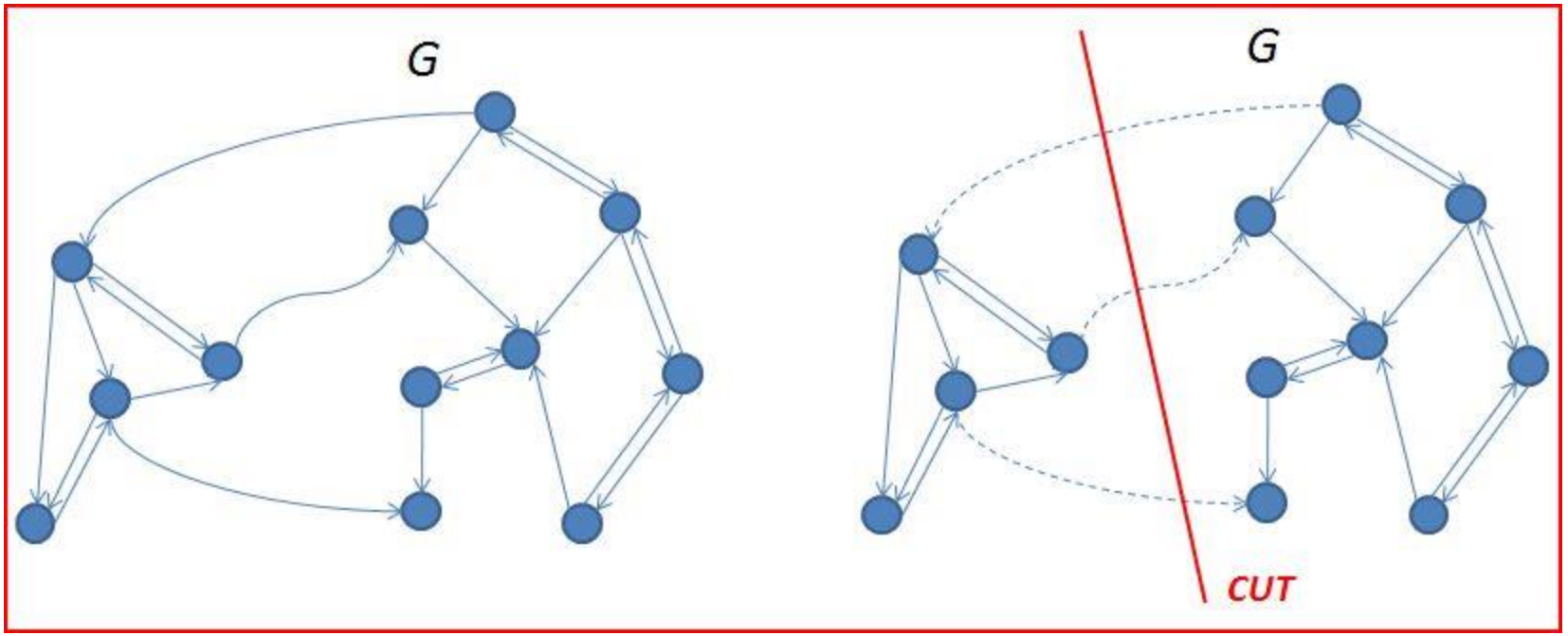
Identifying web communities (2)

- If the Web is represented as a directed graph or web-graph, then the “Max Flow-Min Cut" method [Ford56] can be used to identify web communities [Flake02].
- Let *G=(V,E)* be the directed web-graph with edge capacities $c(u_i, u_j) \in Z^+$ and two vertices:
- A web community is defined as a set a vertex

$$U \subset V \,/\, \forall u_i \in U, \exists u_j \in U \,/\, u_i \rightarrow u_j \quad with \quad i \neq j \quad and \quad u_j \notin (V - U)$$

- Find the maximum flow that it is possible to route from *s* to the sink *t*, maintaining the capacity constrains.

Identifying web communities (3)



Identifying web communities (4)

- For **large number of N** node then
  - it is a **larger number of sub-graph** (worst case 2N)
- These algorithm are **related to clustering** approach of data mining:
  - We are grouping things together.
- A **bunch of others algorithms** appears recently.
- They are grouped in:
  - **Partition**
    - (based on cutting the graph)
  - **Hierarchical**
    - (based on cluster hierarchy)

Section 4.6

Further topics on WSM

SPAM

- **Creating spam resistant algorithm** is a current research topics.
- Other way are creating **algorithm to identify web page spammer**.
  - Finding pages that reciprocal percentage links (links that go and after point back), if this **number goes over a threshold (80%) it is a spammer**.
- Other way is to build a **ranking of bad pages** (http://pr.efactory.de/) almost the same formula but with in-link.

$$r_p^{(k+1)} = (1-d)\frac{1}{n} r_q^{(k)} + d \sum_{\forall q, p \in P / q \to p} \frac{1}{K_q} r_q^{(k)}$$

- In this case, we potentate the inverse.

Personalization

- Web searcher with **personal settings**
- **Interest** of users
- Recommendations
- Personal statistics

Summary

- HITS and PageRank use back-links as a **means of adjusting** the **"worthiness"** or **"importance"** of a page
- Both use **iterative process** over **matrix/vector values** to reach a convergence point
- **HITS is query-dependent** (thus too expensive to compute in general)
- **PageRank is query-independent** and considered **more stable**
- Just **one piece of the overall picture**…