Chapter 6: Link Analysis

Most slides courtesy Bing Liu

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: 10 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 (Scotch Plains, NJ) filed a hyperlink based search patent. The method uses words in anchor text of hyperlinks.
- Web pages on the other hand 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.
- **"Stanford University"** the great!
  - Google: Sergey Brin and Larry Page (PhD candidates in CS)
  - Yahoo!: Jerry Yang and David Filo (PhD candidates in EE)
  - HP, Sun, Cisco, …

Introduction (cont …)

- Apart from search ranking, hyperlinks are also useful for finding Web communities.
  - A Web community is a cluster of densely linked pages representing a group of people with a special interest.
- Beyond explicit hyperlinks on the Web, links in other contexts are useful too, e.g.,
  - for discovering communities of named entities (e.g., people and organizations) in free text documents, and
  - for analyzing social phenomena in emails..

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.
- The interactions and relationships can be represented with 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.

Social networks 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 analysis 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.

Centrality

- Important or prominent actors are those that are linked or involved with other actors extensively.
- A person with extensive contacts (links) or communications with many other people in the organization is considered more important than a person with relatively fewer contacts.
- The links can also be called **ties**. A central actor is one involved in many ties or connections.

Degree Centrality

Central actors are the most active actors that have most links or ties with other actors. Let the total number of actors in the network be n.

Undirected graph: In an undirected graph, the **degree centrality** of an actor i (denoted by $C_D(i)$) is simply the node degree (the number of edges) of the actor node, denoted by $d(i)$, normalized with the maximum degree, n-1.

$$C_D(i) = \frac{d(i)}{n-1} \qquad (1)$$

Directed graph: In this case, we need to distinguish **in-links** of actor i (links pointing to i), and **out-links** (links pointing out from i). The degree centrality is defined based on only the out-degree (the number of out-links or edges), $d_o(i)$.

$$C'_D(i) = \frac{d_o(i)}{n-1} \qquad (2)$$

Closeness Centrality

This view of centrality is based on the closeness or distance. The basic idea is that 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, we can use the shortest distance to compute this measure. Let the shortest distance from actor i to actor j be $d(i, j)$.

Undirected graph: The closeness centrality $C_C(i)$ of actor i is defined as

$$C_C(i) = \frac{n-1}{\sum_{j=1}^{n} d(i,j)} \tag{3}$$

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 shortest distances from i to all other actors. Note that this equation is only meaningful for a connected graph.

Directed graph: The same equation can be used for a directed graph. The distance computation needs to consider directions of links or edges.

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.

Betweenness Centrality (cont'd)

- **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.

$$\sum_{j<k} \frac{p_{jk}(i)}{p_{jk}} \qquad (4)$$

Betweenness Centrality (cont…)

Note that there may be multiple shortest paths between j and k. Some passes i and some do not. If we are to ensure the value range is between 0 and 1, we can normalize it with $(n\text{-}1)(n\text{-}2)/2$, which is the maximum value of the above quantity, i.e., the number of pairs of actors not including i. The final betweenness of i is defined as

$$C_B(i) = \frac{2\sum_{j<k} \frac{p_{jk}(i)}{p_{jk}}}{(n-1)(n-2)} \tag{5}$$

Unlike the closeness measure, the betweenness can be computed even if the graph is not connected.

Directed graph: The same equation can be used but must be multiplied by 2 because there are now $(n\text{-}1)(n\text{-}2)$ pairs considering a path from j to k is different from a path from k to j. Likewise, p_{jk} must consider paths from both directions.

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 is one who is object of extensive ties as a recipient.
  - To compute the prestige: we use only in-links.
- Difference between centrality and prestige:
  - centrality focuses on out-links
  - prestige focuses on in-links.
- We study three prestige measures. **Rank prestige** forms the basis of most Web page link analysis algorithms, including PageRank and HITS.

Degree prestige

Based on the definition of the prestige, it is clear that 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.

$$P_D(i) = \frac{d_I(i)}{n-1}, \tag{6}$$

where $d_I(i)$ is in-degree of i (the number of in-links of actor i_i) and n is the total number of actors in the network. As in the degree centrality, dividing $n - 1$ 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 degree index of 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*.
  - We consider every actor *j* that can reach *i*.
- Let I_i be the set of actors that can reach actor *i*.
- The **proximity** is defined as closeness or distance of other actors to *i*.
- Let $d(j, i)$ denote the distance from actor *j* to actor *i*.

Proximity prestige (cont …)

$$\frac{\sum_{j \in I_i} d(j,i)}{|I_i|}, \tag{7}$$

where $|I_i|$ is the size of the set I_i. If we look at the ratio or proportion of actors who can reach i to the average distance that these actors are from i, we obtain the following, which has the value range of [0, 1]:

$$P_P(i) = \frac{|I_i| / (n-1)}{\sum_{j \in I_i} d(j,i) / |I_i|}, \tag{8}$$

where $|I_i|/(n\text{-}1)$ is the proportion of actors that can reach actor i. In one extreme, every actor can reach actor i, which gives $|I_i|/(n\text{-}1) = 1$. The denominator is 1 if every actor is adjacent to i. Thus, $P_P(i) = 1$. On the other extreme, no actor can reach actor i. Then $|I_i| = 0$, and $P_P(i) = 0$. Each link has the unit distance.

Rank prestige

- In the previous two prestige measures, an important factor is not considered
  - the **prominence** of individual actors who do the “voting”
- In the real world, a person *i* chosen by an important person is more prestigious than chosen by a less important person.
  - For example, if a company CEO votes for a person is much more important than a worker votes for the person.
- If one’s circle of influence is full of prestigious actors, then one’s own prestige is also high.
  - Thus one’s prestige is affected by the ranks or statuses of the involved actors.

Rank prestige (cont …)

- Based on this intuition, the rank prestige $P_R(i)$ is defined as a linear combination of links that point to *i*:

$$P_R(i) = A_{1i}P_R(1) + A_{2i}P_R(2) + \ldots + A_{ni}P_R(n), \tag{9}$$

where $A_{ji} = 1$ if *j* points to *i*, and 0 otherwise. This equation says that an actor's rank prestige is a function of the ranks of the actors who vote or choose the actor, which makes perfect sense.

Since we have *n* equations for *n* actors, mathematically we can write them in the matrix notation. We use $\boldsymbol{P}$ to represent the vector that contains all the rank prestige values, i.e., $\boldsymbol{P} = (P_R(1), P_R(2), \ldots, P_R(n))^T$ (*T* means **matrix transpose**). $\boldsymbol{P}$ is represented as a column vector. We use matrix $\boldsymbol{A}$ (where $A_{ij} = 1$ if *i* points to *j*, and 0 otherwise) to represent the adjacency matrix of the network or graph. As a notational convention, we use bold italic letters to represent matrices. We then have

$$\boldsymbol{P} = \boldsymbol{A}^T\boldsymbol{P} \tag{10}$$

This equation is precisely the characteristic equation used for finding the **eigensystem** of the matrix $\boldsymbol{A}^T$. $\boldsymbol{P}$ is an **eigenvector** of $\boldsymbol{A}^T$.

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.

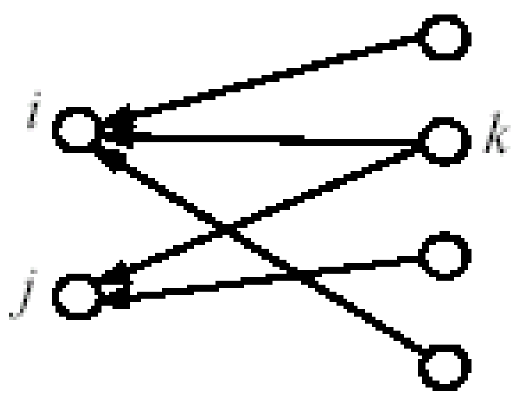


Fig. 2. Paper *i* and paper *j* are co-cited by paper *k*

Co-citation

- Let ***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*,

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

- C_{ii} is naturally the number of papers that cite *i*.
- A square matrix ***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.

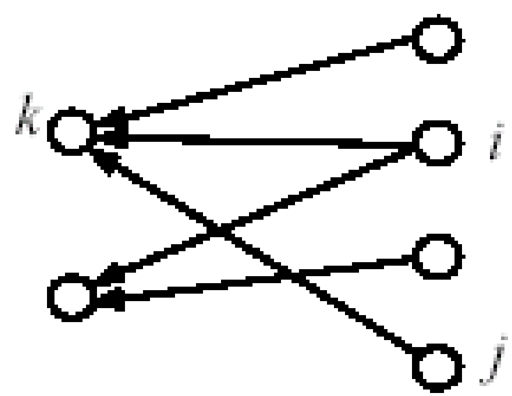


Fig. 3. Both paper *i* and paper *j* cite paper *k*

Bibliographic coupling (cont'd)

We use B_{ij} to represent the number of papers that are cited by both papers i and j.

$$B_{ij} = \sum_{k=1}^{n} L_{ik} L_{jk}. \tag{12}$$

B_{ii} is naturally the number of references (in the reference list) of paper i. A square matrix $\boldsymbol{B}$ can be formed with B_{ij}, and it is called the **bibliographic coupling matrix**. Bibliographic coupling is also symmetric and is regarded as a similarity measure of two papers in clustering.

Road map

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

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.

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

Matrix notation

- We have a system of n linear equations with n unknowns. We can use a matrix to represent them.
- Let $\boldsymbol{P}$ be a n-dimensional column vector of PageRank values, i.e., $\boldsymbol{P} = (P(1), P(2), \ldots, P(n))^T$.
- Let $\boldsymbol{A}$ be the adjacency matrix of our graph with

$$A_{ij} = \begin{cases} \dfrac{1}{O_i} & if\ (i, j) \in E \\ 0 & otherwise \end{cases} \qquad (14)$$

- We can write the n equations with (PageRank)

$$\boldsymbol{P} = \boldsymbol{A}^T \boldsymbol{P} \qquad (15)$$

Solve the PageRank equation

$$P = A^T P \qquad (15)$$

- This is the characteristic equation of the **eigensystem**, where the solution to ***P*** is an **eigenvector** with the corresponding **eigenvalue** of 1.
- It turns out that if ***some conditions*** are satisfied, 1 is the largest **eigenvalue** and the PageRank vector ***P*** is the **principal eigenvector**.
- A well known mathematical technique called **power iteration** can be used to find ***P***.
- **Problem**: the above Equation does not quite suffice because the Web graph does not meet the conditions.

Using Markov chain

- To introduce these conditions and the enhanced equation, let us derive the same Equation (15) based on the **Markov chain**.
  - In the Markov chain, each Web page or node in the Web graph is regarded as a state.
  - A hyperlink is a transition, which leads from one state to another state with a probability.
- This framework models Web surfing as a stochastic process.
- It models a Web surfer randomly surfing the Web as state transition.

Random surfing

- Recall we use O_i to denote the number of out-links of a node i.
- Each transition probability is $1/O_i$ if we assume the Web surfer will click the hyperlinks in the page i uniformly at random.
    - The "back" button on the browser is not used and
    - the surfer does not type in an URL.

Transition probability matrix

- Let ***A*** be the state transition probability matrix,,

$$\boldsymbol{A} = \begin{pmatrix} A_{11} & A_{12} & . & . & . & A_{1n} \\ A_{21} & A_{22} & . & . & . & A_{2n} \\ . & . & & & & . \\ . & . & & & & . \\ . & . & & & & . \\ A_{n1} & A_{n2} & . & . & . & A_{nn} \end{pmatrix}$$

- A_{ij} represents the transition probability that the surfer in state *i* (page *i*) will move to state *j* (page *j*). A_{ij} is defined exactly as in Equation (14).

Let us start

- Given an initial probability distribution vector that a surfer is at each state (or page)
  - $\boldsymbol{p}_0 = (p_0(1), p_0(2), \ldots, p_0(n))^T$ (a column vector) and
  - an $n \times n$ **transition probability matrix** $\boldsymbol{A}$,

  we have

$$\sum_{i=1}^{n} p_0(i) = 1 \qquad (16)$$

$$\sum_{j=1}^{n} A_{ij} = 1 \qquad (17)$$

- If the matrix $\boldsymbol{A}$ satisfies Equation (17), we say that $\boldsymbol{A}$ is the **stochastic matrix** of a Markov chain.

Back to the Markov chain

- In a Markov chain, a question of common interest is:
  - Given $\boldsymbol{p}_0$ at the beginning, what is the probability that *m* steps/transitions later the Markov chain will be at each state *j*?
- We determine the probability that the system (or the random surfer) is in state *j* after 1 step (1 transition) by using the following reasoning:

$$p_1(j) = \sum_{i=1}^{n} A_{ij}(1)\, p_0(i) \qquad (18)$$

State transition

$$p_1(j) = \sum_{i=1}^{n} A_{ij}(1) p_0(i), \tag{18}$$

where $A_{ij}(1)$ is the probability of going from i to j after 1 transition, and $A_{ij}(1) = A_{ij}$. We can write it with a matrix:

$$\boldsymbol{p}_1 = \boldsymbol{A}^{\boldsymbol{T}} \boldsymbol{p}_{\boldsymbol{0}} \tag{19}$$

In general, the probability distribution after k steps/transitions is:

$$\boldsymbol{p}_k = \boldsymbol{A}^{\boldsymbol{T}} \boldsymbol{p}_{\boldsymbol{k-1}} \tag{20}$$

Stationary probability distribution

- By a Theorem of the Markov chain,
  - a finite Markov chain defined by the **stochastic matrix *A*** has a unique **stationary probability distribution** if ***A*** is **irreducible** and **aperiodic**.
- The stationary probability distribution means that after a series of transitions $\boldsymbol{p}_k$ will converge to a steady-state probability vector $\boldsymbol{\pi}$ regardless of the choice of the initial probability vector $\boldsymbol{p}_0$, i.e.,

$$\lim_{k \to \infty} \boldsymbol{p}_k = \boldsymbol{\pi} \qquad (21)$$

PageRank again

- When we reach the steady-state, we have $\boldsymbol{p}_k = \boldsymbol{p}_{k+1} = \boldsymbol{\pi}$, and thus

$$\boldsymbol{\pi} = \boldsymbol{A}^T \boldsymbol{\pi}.$$

- $\boldsymbol{\pi}$ is the **principal eigenvector** of $\boldsymbol{A}^T$ with **eigenvalue** of 1.
- In PageRank, $\boldsymbol{\pi}$ is used as the PageRank vector $\boldsymbol{P}$. We again obtain Equation (15), which is re-produced here as Equation (22):

$$\boldsymbol{P} = \boldsymbol{A}^T \boldsymbol{P} \qquad (22)$$

Is $\boldsymbol{P} = \boldsymbol{\pi}$ justified?

- Using the stationary probability distribution $\boldsymbol{\pi}$ as the PageRank vector is reasonable and quite intuitive because
  - it reflects the long-run probabilities that a random surfer will visit the pages.
  - A page has a high prestige if the probability of visiting it is high.

Back to the Web graph

- Now let us come back to the real Web context and see whether the above conditions are satisfied, i.e.,
  - whether ***A*** is a stochastic matrix and
  - whether it is irreducible and aperiodic.
- None of them is satisfied.
- Hence, we need to extend the ideal-case Equation (22) to produce the “actual PageRank” model.

A is a not stochastic matrix

- ***A*** is the transition matrix of the Web graph

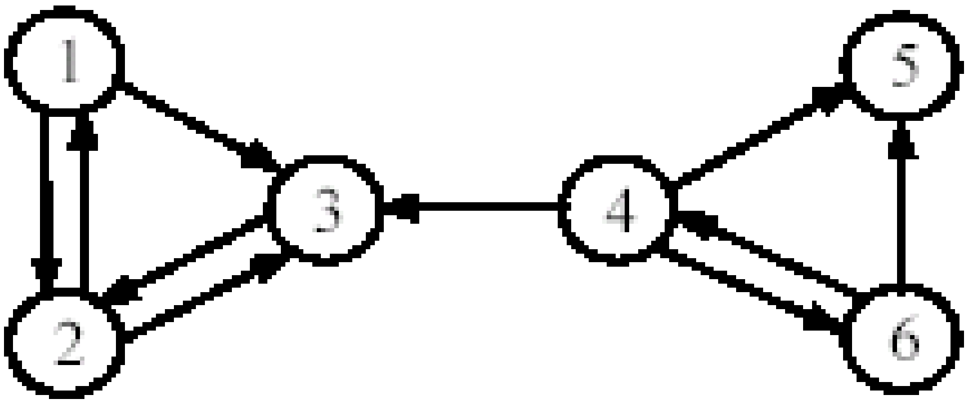
$$A_{ij} = \begin{cases} \dfrac{1}{O_i} & if\,(i, j) \in E \\ 0 & otherwise \end{cases}$$

- It does not satisfy equation (17)

$$\sum_{j=1}^{n} A_{ij} = 1$$

- because many Web pages have no out-links, which are reflected in transition matrix ***A*** by some rows of complete 0's.
  - Such pages are called the **dangling pages** (nodes).

An example Web hyperlink graph



$$A = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}$$

Fix the problem: two solutions

1. Remove those pages with no out-links during the PageRank computation as these pages do not affect the ranking of any other page directly.
2. Add a complete set of outgoing links from each such page *i* to all the pages on the Web.

Let us use the second way

$$\bar{A} = \begin{pmatrix} 0 & 1/2 & 1/2 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 1/3 \\ 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 0 & 0 & 0 & 1/2 & 1/2 & 0 \end{pmatrix}$$

A is a not irreducible

- Irreducible means that the Web graph G is strongly connected.

Definition: A directed graph $G = (V, E)$ is strongly connected if and only if, for each pair of nodes $u, v \in V$, there is a path from u to v.

- A general Web graph represented by ***A*** is not irreducible because
  - for some pair of nodes u and v, there is no path from u to v.
  - In our example, there is no directed path from nodes 3 to 4.

A is a not aperiodic

- A state *i* in a Markov chain being periodic means that there exists a directed cycle that the chain has to traverse.

Definition: A state *i* is **periodic** with period k > 1 if k is the smallest number such that all paths leading from state *i* back to state *i* have a length that is a multiple of k.

  - If a state is not periodic (i.e., k = 1), it is **aperiodic**.
  - A Markov chain is **aperiodic** if all states are aperiodic.

An example: periodic

- Fig. 5 shows a periodic Markov chain with $k = 3$. Eg, if we begin from state 1, to come back to state 1 the only path is 1-2-3-1 for some number of times, say h. Thus any return to state 1 will take $3h$ transitions.

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

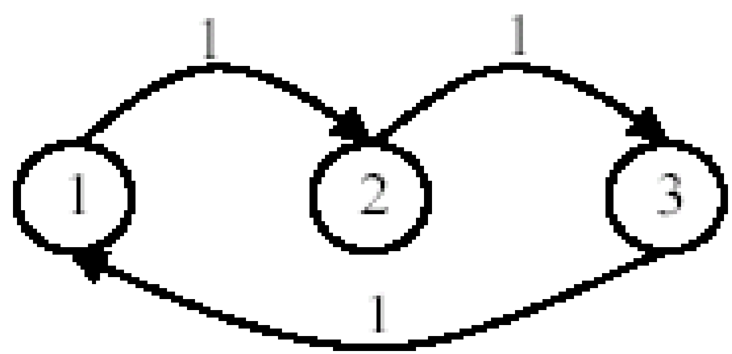


Fig. 5. A Periodic Markov chain with $k = 3$.

Handle irreducible and aperiodic

- It is easy to deal with the above two problems with a single strategy.
- Add a link from each page to every page and give each link a small transition probability controlled by a parameter *d*.
- Obviously, the augmented transition matrix becomes **irreducible** and **aperiodic**

Improved PageRank

- After this augmentation, at a page, the random surfer has two options
  - With probability *d*, he randomly chooses an out-link to follow.
  - With probability 1-*d*, he jumps to a random page
- Equation (25) gives the improved model,

$$\boldsymbol{P} = ((1-d)\frac{\boldsymbol{E}}{n} + dA^T)\boldsymbol{P} \qquad (25)$$

where $\boldsymbol{E}$ is $\boldsymbol{ee}^T$ ($\boldsymbol{e}$ is a column vector of all 1's) and thus $\boldsymbol{E}$ is a $n \times n$ square matrix of all 1's.

Follow our example

$$(1-d)\frac{\boldsymbol{E}}{n}+d\boldsymbol{A}^T=\begin{pmatrix} 1/60 & 7/15 & 1/60 & 1/60 & 1/6 & 1/60 \\ 7/15 & 1/60 & 11/12 & 1/60 & 1/6 & 1/60 \\ 7/15 & 7/15 & 1/60 & 19/60 & 1/6 & 1/60 \\ 1/60 & 1/60 & 1/60 & 1/60 & 1/6 & 7/15 \\ 1/60 & 1/60 & 1/60 & 19/60 & 1/6 & 7/15 \\ 1/60 & 1/60 & 1/60 & 19/60 & 1/6 & 1/60 \end{pmatrix}$$

The final PageRank algorithm

- (1-*d*)***E***/*n* + *d****A****^{T}* is a **stochastic matrix** (transposed). It is also **irreducible** and **aperiodic**
- If we scale Equation (25) so that $\mathbf{e}^T\boldsymbol{P} = n$,

$$\boldsymbol{P} = (1-d)\mathbf{e} + dA^T\boldsymbol{P} \tag{27}$$

- PageRank for each page *i* is

$$P(i) = (1-d) + d\sum_{j=1}^{n} A_{ji}P(j) \tag{28}$$

The final PageRank (cont …)

- (28) is equivalent to the formula given in the PageRank paper

$$P(i) = (1-d) + d \sum_{(j,i)\in E} \frac{P(j)}{O_j}$$

- The parameter *d* is called the **damping factor** which can be set to between 0 and 1. *d* = 0.85 was used in the PageRank paper.

Compute PageRank

- Use the power iteration method

PageRank-Iterate(G)

$P_0 \leftarrow e/n$

$k = 1$

repeat

$P_{k+1} \leftarrow (1-d)e + dA^T P_k$;

$k = k + 1$;

until $||P_{k+1} - P_k||_1 < \varepsilon$

return P_{k+1}

Fig. 6. The power iteration method for PageRank

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 cannot 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

- 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**.

Authorities and Hubs

Authority: Roughly, an authority is a page with many in-links.

- The idea is that the page may have good or authoritative content on some topic and
- thus many people trust it and link to it.

Hub: A hub is a page with many out-links.

- The page serves as an organizer of the information on a particular topic and
- points to many good authority pages on the topic.

Examples

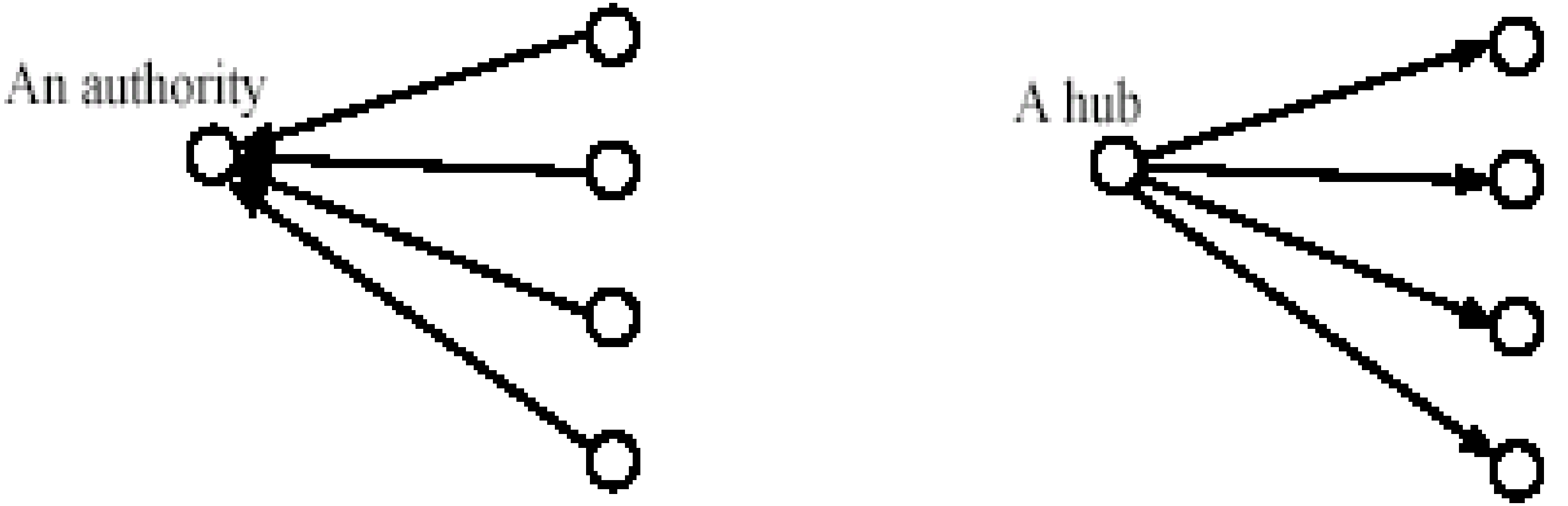


Fig. 7. An authority page and a hub page

The key idea of HITS

- A good hub points to many good authorities, and
- A good authority is pointed to by many good hubs.
- Authorities and hubs have a **mutual reinforcement relationship**. Fig. 8 shows some densely linked authorities and hubs (a **bipartite sub-graph**).

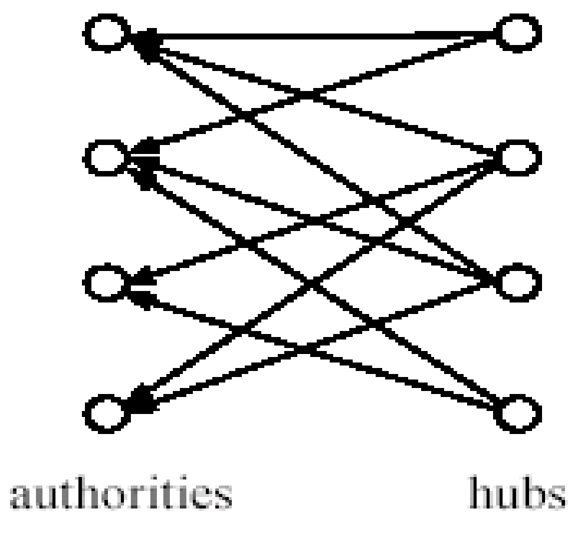


Fig. 8. A densely linked set of authorities and hubs

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 to by a page in *W* and any page that points to a page in *W*. This gives a larger set *S*, **base set**.

The link graph G

- HITS works on the pages in *S*, and assigns every page in *S* an **authority score** and a **hub score**.
- Let the number of pages in *S* be *n*.
- We again use *G* = (*V*, *E*) to denote the hyperlink graph of *S*.
- We use ***L*** to denote the adjacency matrix of the graph.

$$L_{ij} = \begin{cases} 1 & if\,(i,j) \in E \\ 0 & otherwise \end{cases}$$

The HITS algorithm

- Let the authority score of the page *i* be *a*(*i*), and the hub score of page *i* be *h*(*i*).
- The mutual reinforcing relationship of the two scores is represented as follows:

$$a(i) = \sum_{(j,i)\in E} h(j) \qquad (31)$$

$$h(i) = \sum_{(i,j)\in E} a(j) \qquad (32)$$

HITS in matrix form

- We use $\boldsymbol{a}$ to denote the column vector with all the authority scores,

  $\boldsymbol{a} = (a(1), a(2), \ldots, a(n))^T$, and

- use $\boldsymbol{h}$ to denote the column vector with all the authority scores,

  $\boldsymbol{h} = (h(1), h(2), \ldots, h(n))^T$,

- Then,

$$\boldsymbol{a} = \boldsymbol{L}^T\boldsymbol{h} \tag{33}$$

$$\boldsymbol{h} = \boldsymbol{L}\boldsymbol{a} \tag{34}$$

Computation of HITS

- The computation of authority scores and hub scores is the same as the computation of the PageRank scores, using power iteration.
- If we use $\boldsymbol{a}_k$ and $\boldsymbol{h}_k$ to denote authority and hub vectors at the kth iteration, the iterations for generating the final solutions are

$$a_k = L^T L a_{k-1} \tag{35}$$

$$h_k = L L^T h_{k-1} \tag{36}$$

starting with

$$a_0 = h_0 = (1, 1, \ldots, 1), \tag{37}$$

The algorithm

HITS-Iterate(G)

$\boldsymbol{a}_0 = \boldsymbol{h}_0 = (1, 1, \ldots, 1)$;

$k = 1$

Repeat

$\boldsymbol{a}_k = \boldsymbol{L}^T \boldsymbol{L} \boldsymbol{a}_{k-1}$;

$\boldsymbol{h}_k = \boldsymbol{L} \boldsymbol{L}^T \boldsymbol{h}_{k-1}$;

normalize $\boldsymbol{a}_k$;

normalize $\boldsymbol{h}_k$;

$k = k + 1$;

until $\boldsymbol{a}_k$ and $\boldsymbol{h}_k$ do not change significantly;

return $\boldsymbol{a}_k$ and $\boldsymbol{h}_k$

. **Fig. 9.** The HITS algorithm based on power iteration

Relationships with co-citation and bibliographic coupling

- Recall that co-citation of pages *i* and *j*, denoted by C_{ij}, is

$$C_{ij} = \sum_{k=1}^{n} L_{ki} L_{kj} = (\boldsymbol{L}^T \boldsymbol{L})_{ij}$$

  - the authority matrix ($\boldsymbol{L}^T\boldsymbol{L}$) of HITS is the co-citation matrix ***C***
- bibliographic coupling of two pages *i* and *j*, denoted by B_{ij} is

$$B_{ij} = \sum_{k=1}^{n} L_{ik} L_{jk} = (\boldsymbol{L}\boldsymbol{L}^T)_{ij},$$

  - the hub matrix ($\boldsymbol{L}\boldsymbol{L}^T$) of HITS is the bibliographic coupling matrix ***B***

Strengths and weaknesses of HITS

- Strength: its ability to rank pages according to the query topic, which may be able to provide more relevant authority and hub pages.
- Weaknesses:
  - It is easily spammed. It is in fact quite easy to influence HITS since adding out-links in one's own page is so easy.
  - Topic drift. Many pages in the expanded set may not be on topic.
  - Inefficiency at query time: The query time evaluation is slow. Collecting the root set, expanding it and performing eigenvector computation are all expensive operations

Road map

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

Summary

- In this chapter, we introduced
  - Social network analysis, centrality and prestige
  - Co-citation and bibliographic coupling
  - PageRank, which powers Google
  - HITS (foundation of Ask.com)
- 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
  - Named entity 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.