

Social Network Analysis

Irwin King, Baichuan Li, Tom Chao Zhou

Department of Computer Science & Engineering
The Chinese University of Hong Kong

6/10/2012

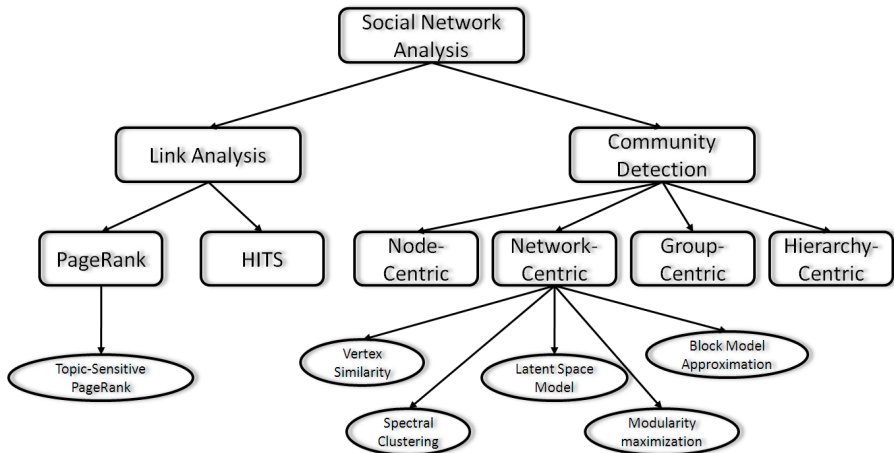


Outline

- 1 Link Analysis
 - PageRank
 - Topic-Sensitive PageRank
 - HITS
 - Demo
- 2 Community Detection
 - Introduction
 - Methods
 - Node-Centric Community Detection
 - Group-Centric Community Detection
 - Network-Centric Community Detection
 - Hierarchy-Centric Community Detection
 - Summary
- 3 References



Content



Outline

- 1 Link Analysis
 - PageRank
 - Topic-Sensitive PageRank
 - HITS
 - Demo
- 2 Community Detection
 - Introduction
 - Methods
 - Node-Centric Community Detection
 - Group-Centric Community Detection
 - Network-Centric Community Detection
 - Hierarchy-Centric Community Detection
 - Summary
- 3 References



Outline

- 1 Link Analysis
 - PageRank
 - Topic-Sensitive PageRank
 - HITS
 - Demo
- 2 Community Detection
 - Introduction
 - Methods
 - Node-Centric Community Detection
 - Group-Centric Community Detection
 - Network-Centric Community Detection
 - Hierarchy-Centric Community Detection
 - Summary
- 3 References



The Web Is a Graph

Google

wcci 2012

SafeSearch off

Search

About 79,600 results (0.36 seconds)

Web

[Welcome to WCCI 2012](#)
www.ieee-wcci2012.org/ieee-wcci2012/index.php?... - Cached
Call for Participation: IEEE Women in Computational Intelligence and Women in Engineering Reception and Panel @ **WCCI 2012**. The IEEE Women in ...

[Special Sessions - WCCI 2012](#)
www.ieee-wcci2012.org/ieee-wcci2012/index.php?... - Cached
Call for Participation: IEEE Women in Computational Intelligence and Women in Engineering Reception and Panel @ **WCCI 2012**. The IEEE Women in ...

[Computational Intelligence: WCCI 2012 Panel Session on ...](#)
computational-intelligence.blogspot.com/2012/... - WCCI-2012 - - Labeled
5 days ago - The following panel session at **WCCI 2012** is organised by the IEEE Computational Intelligence Society's Curriculum Subcommittee (which I ...

[IEEE/WCCI 2012 - Systems and Industrial Engineering - University ...](#)
engr.arizona.edu/wcci2012/... - Cached
Special Session on: Emerging Trends in Fuzzy Cognitive Maps, at the **2012 IEEE International Conference on Fuzzy Systems (FUZZ IEEE 2012)**. Part of the ...

[The PTSP Game Competition](#)
www.ptsp-game.net/ - Cached
Welcome to the Physical Travelling Salesman Problem competition, which will be held at **WCCI 2012** and CIS 2012. This competition is being run by Diego ...

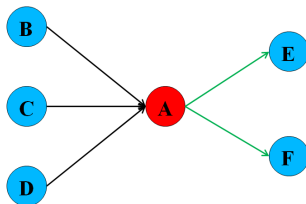
[WCCI 2012 Panel on Real-World Applications of CI](#)
ieee-cis.org/ptsp.com/2012/... - WCCI-2012 panel on real-world... - Cached
17 May 2012 - **WCCI 2012** Panel on Real-World Applications of CI. The Industry Liaison Sub-Committee would like to announce that the **WCCI 2012** Panel ...



PageRank

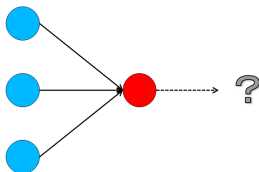
Idea

- Most web pages contain hyperlinks
- Assign a score to each page to measure its **importance** (i.e., PageRank value, usually between 0 and 1)
- A web page propagate its PR through out-links, and absorb others' PRs through in-links



Teleport

- What about the web pages without out-links (dead-ends)?



- Random surfer: *teleport*
 - Jumps from a node to any other node in the web graph
 - Choose the destination **uniformly** at random
 - E.g., let N is the total number of nodes in the web graph, the surfer to each node has the probability of $\frac{1}{N}$



Algorithm

- If page A has pages $\{T_1, T_2, \dots, T_n\}$ which point to it, let $Out(T_1)$ denote the number of out-links of T_1 :

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

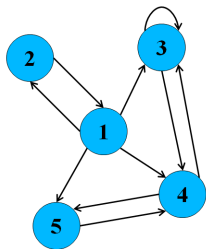
where $d \in (0, 1)$ is a damping factor, N is the total number of web pages

- $\frac{1}{N}$ represents the *teleport* operation



Transition Probability Matrix

- Use a matrix P to represent the surfer probability from one node to the other
 - P_{ij} tells the probability that we visit node j of node i
 - $\forall i, j, P_{ij} \in [0, 1]$
 - $\forall i, \sum_{j=1}^N P_{ij} = 1$



If $\alpha = 0.5$,

$$P = \begin{pmatrix} 0.1 & 0.225 & 0.225 & 0.225 & 0.225 \\ 0.6 & 0.1 & 0.1 & 0.1 & 0.1 \\ 0.1 & 0.1 & 0.35 & 0.35 & 0.1 \\ 0.1 & 0.1 & 0.35 & 0.1 & 0.35 \\ 0.1 & 0.1 & 0.1 & 0.6 & 0.1 \end{pmatrix}$$



Markov Chain

- P is a transition probability matrix for a **Markov chain**
 - A Markov chain is a discrete-time stochastic process
 - Consists of N states
 - The Markov chain can be in one state i at any given time-step, and turn into state j in the next time-step with probability P_{ij}
 - Probability vector $\vec{\pi}$

Ergodic Markov Chain

- A Markov chain is called an **ergodic** chain if it is possible to go **from every state to every state** (non necessary in one move)
- For any ergodic Markov chain, there is a **unique steady-state** probability vector $\vec{\pi}$
 - $\vec{\pi}$ is the principle left eigenvector of P with the largest eigenvalue
 - **PageRank=steady state probability**

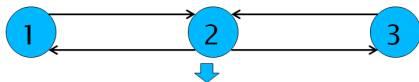


How to Compute PageRank?

- Compute PageRank iteratively
 - Let $\vec{\pi}$ be the initial probability vector
 - At time t , the probability vector becomes $\vec{\pi}P^t$
 - When t is very large, $\vec{\pi}P^{t+1} = \vec{\pi}P^t$, regardless of where we start (The initialization of $\vec{\pi}$ is unimportant)
- Compute PageRank directly
 - $\vec{\pi}P = 1 \cdot P$
 - $\vec{\pi}$ is the eigenvector of P whose corresponding eigenvalue is 1



Example



$$\alpha = 0.5$$

$$P = \frac{1}{2} \begin{pmatrix} 0 & 1 & 0 \\ 1/2 & 0 & 1/2 \\ 0 & 1 & 0 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{pmatrix}$$

$$= \begin{pmatrix} 1/6 & 2/3 & 1/6 \\ 5/12 & 1/6 & 5/12 \\ 1/6 & 2/3 & 1/6 \end{pmatrix}$$

$$\vec{x}_0 = (1 \ 0 \ 0)$$

$$\vec{x}_1 = \vec{x}_0 P = \begin{pmatrix} 1/6 & 2/3 & 1/6 \end{pmatrix}$$

\vec{x}_0	1	0	0
\vec{x}_1	1/6	2/3	1/6
\vec{x}_2	1/3	1/3	1/3
\vec{x}_3	1/4	1/2	1/4
\vec{x}_4	7/24	5/12	7/24
...
\vec{x}	5/18	4/9	5/18



PageRank in Information Retrieval

- Preprocessing
 - Given graph of links, build matrix P
 - Apply teleportation
 - From modified matrix, compute $\vec{\pi}$
 - π_i is the PageRank of page i .
- Query processing
 - Retrieve pages satisfying the query
 - Rank them by their PageRank
 - Return reranked list to the user



PageRank Issues

- Real surfers are not random surfers
 - Back buttons, bookmarks, directories – and search!
- Simple PageRank ranking produces bad results for many pages
 - Consider the query [video service]
 - The Yahoo home page (i) has a very high PageRank and (ii) contains both *video* and *service*.
 - According to PageRank, the Yahoo home page would be top-ranked
 - Clearly not desirable
- In practice: rank according to weighted combination of raw text match, anchor text match, PageRank & other factors



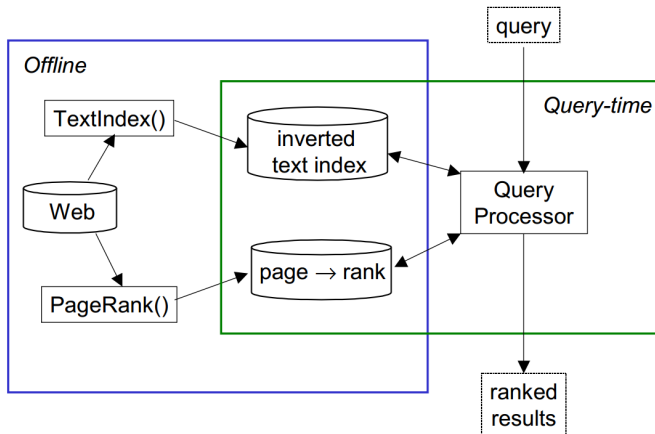
Topic-Sensitive PageRank

Motivation

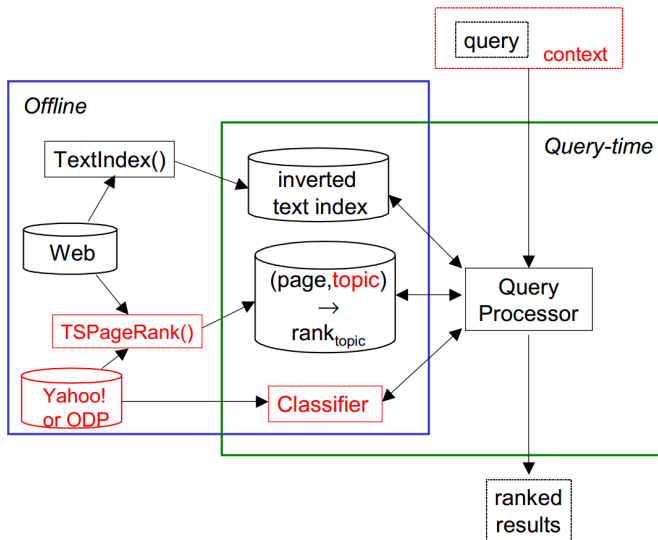
- PageRank provides a general “importance” of a web page
- The “importance” biased to **different topics**
- Compute a set of “importance” scores of a page with respect to various topics



Standard PageRank



Topic-Sensitive PageRank



Phase 1: ODP-biasing

- Generate a set of biased PageRank vectors using a set of basis topics
 - Cluster the Web page repository into a small number of clusters
 - Utilize the hand constructed Open Directory
- Performed offline, during preprocessing of crawled data
- Let T_j be the set of URLs in the ODP category c_j , we compute the damping vector $\mathbf{p} = \mathbf{v}_j$ where

$$v_{ji} = \begin{cases} \frac{1}{|T_j|} & i \in T_j \\ 0 & i \notin T_j \end{cases}$$

The PageRank vector for topic c_j is given by $\mathbf{PR}(\alpha, \mathbf{v}_j)$.

- Compute the 16 class term vectors \mathbf{D}_j where D_{jt} gives the number of occurrences of term t in documents of class c_j .



Phase 2: Query-Time Importance Score

- Performed at query time
- Compute the class probabilities for each of the 16 top-level ODP classes

$$P(c_j|q') = \frac{P(c_j)P(q'|c_j)}{P(q')} \propto P(c_j)\prod_i P(q'_i|c_j)$$

- Retrieve URLs for all documents containing the original query terms q
- Compute the query-sensitive importance score of each of these retrieved URLs

$$S_{qd} = \sum_j P(c_j|q') \cdot r_{jd},$$

where r_{jd} is the rank of document d given by the rank vector $\mathbf{PR}(\alpha, \mathbf{v}_j)$.



Outline

1 Link Analysis

- PageRank
 - Topic-Sensitive PageRank
- **HITS**
- Demo

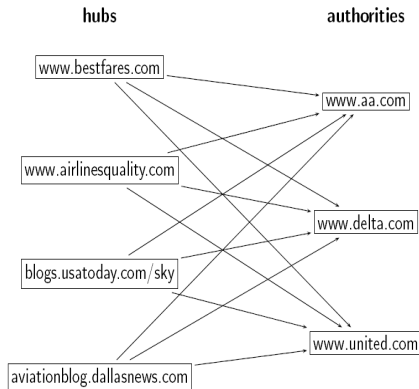
2 Community Detection

- Introduction
- Methods
 - Node-Centric Community Detection
 - Group-Centric Community Detection
 - Network-Centric Community Detection
 - Hierarchy-Centric Community Detection
- Summary

3 References



Two Type of Web Pages



HITS – Hyperlink-Induced Topic Search

- Idea: Two different types of web pages on the web
- Type 1: **Authorities**. An authority page provides direct answers to the information need
 - The home page of the Chicago Bulls sports team
- Type 2: **Hubs**. A hub page contains a number of links to pages answering the information need
 - E.g., for query [chicago bulls]: Bob's list of recommended resources on the Chicago Bulls sports team
- PageRank don't make the distinction between these two



Definition of Hubs and Authorities

- A good hub page for a topic **links to** many authority pages for that topic
- A good authority page for a topic **is linked to** by many hub pages for that topic
- Circular definition – Iterative computation

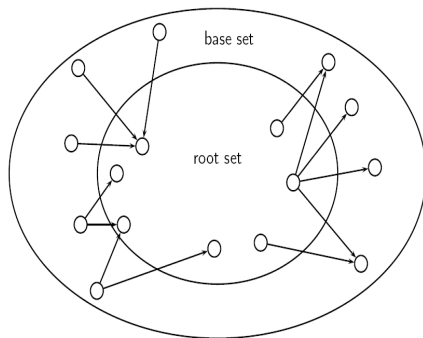


How to Compute Hub and Authority Scores

- Do a regular web search first
- Call the search result the **root set**
- Find all pages that are linked to or link to pages in the root set
- Call this larger set the **base set**
- Finally, compute hubs and authorities for the base set



Root Set and Base Set



- Base set:
 - Nodes that root set nodes link to
 - Nodes that link to root set nodes



Hub and Authority Scores

- Goal: compute for each page d in the base set a **hub score** $h(d)$ and an **authority score** $a(d)$
- Initialization: for all d : $h(d) = 1$, $a(d) = 1$
- Iteratively update all $h(d)$, $a(d)$ until convergence
 - For all d : $h(d) = \sum_{d \mapsto y} a(y)$
 - For all d : $a(d) = \sum_{y \mapsto d} h(y)$
- After convergence:
 - Output pages with highest h scores as top hubs
 - Output pages with highest a scores as top authorities
 - So we output **two** ranked lists



Details

- Scaling
 - To prevent the $a()$ and $h()$ values from getting too big, can scale down after each iteration
 - Scaling factor doesn't really matter
 - We care about the **relative** (as opposed to absolute) values of the scores
- In most cases, the algorithm converges after a few iterations



Example: Authorities for query [Chicago Bulls]

- 0.85 www.nba.com/bulls
 - 0.25 www.essex1.com/people/jmiller/bulls.htm
“da Bulls”
 - 0.20 www.nando.net/SportServer/basketball/nba/chi.html
“The Chicago Bulls”
 - 0.15 users.aol.com/rynecub/bulls.htm
“The Chicago Bulls Home Page”
 - 0.13 www.geocities.com/Colosseum/6095
“Chicago Bulls”
- (Ben-Shaul et al, WWW8)



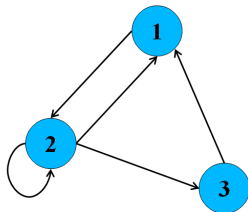
Example: Hubs for query [Chicago Bulls]

- 1.62 www.geocities.com/Colosseum/1778
“Unbelieveabulls!!!!!!”
 - 1.24 www.webring.org/cgi-bin/webring?ring=chbulls
“Erin’s Chicago Bulls Page”
 - 0.74 www.geocities.com/Hollywood/Lot/3330/Bulls.html
“Chicago Bulls”
 - 0.52 www.nobull.net/web_position/kw-search-15-M2.htm
“Excite Search Results: bulls”
 - 0.52 www.halcyon.com/wordsltd/bball/bulls.htm
“Chicago Bulls Links”
- (Ben-Shaul et al, WWW8)



Adjacency Matrix

- We define an $N \times N$ **adjacency matrix** A
 - For $1 \leq i, j \leq N$, the matrix entry A_{ij} tells us whether there is a link from page i to page j ($A_{ij} = 1$) or not ($A_{ij} = 0$)



	d_1	d_2	d_3
d_1	0	1	0
d_2	1	1	1
d_3	1	0	0



Matrix Form of HITS

- Define the hub vector $\vec{h} = (h_1, \dots, h_N)$ where h_i is the hub score of page d_i
- Similarly for \vec{a}
- $h(d) = \sum_{d \mapsto y} a(y)$: $\vec{h} = A\vec{a}$
- $a(d) = \sum_{y \mapsto d} h(y)$: $\vec{a} = A^T \vec{h}$
- By substitution we get: $\vec{h} = AA^T \vec{h}$ and $\vec{a} = A^T A \vec{a}$
- Thus, \vec{h} is an **eigenvector of AA^T** and \vec{a} is an **eigenvector of $A^T A$**



Example

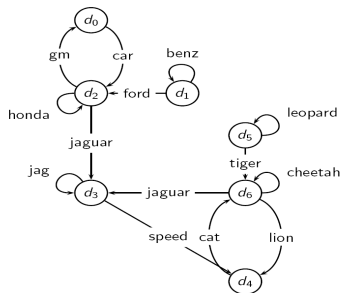


Table: Adjacent Matrix A

	d_0	d_1	d_2	d_3	d_4	d_5	d_6
d_0	0	0	1	0	0	0	0
d_1	0	1	1	0	0	0	0
d_2	1	0	1	2	0	0	0
d_3	0	0	0	1	1	0	0
d_4	0	0	0	0	0	0	1
d_5	0	0	0	0	0	1	1
d_6	0	0	0	2	1	0	1



Hub Vectors

- Set \vec{h}_0 uniformly
- $\vec{h}_i = \frac{1}{d_i} A \cdot \vec{a}_i, i \geq 1$

	\vec{h}_0	\vec{h}_1	\vec{h}_2	\vec{h}_3	\vec{h}_4	\vec{h}_5
d_0	0.14	0.06	0.04	0.04	0.03	0.03
d_1	0.14	0.08	0.05	0.04	0.04	0.04
d_2	0.14	0.28	0.32	0.33	0.33	0.33
d_3	0.14	0.14	0.17	0.18	0.18	0.18
d_4	0.14	0.06	0.04	0.04	0.04	0.04
d_5	0.14	0.08	0.05	0.04	0.04	0.04
d_6	0.14	0.30	0.33	0.34	0.35	0.35



Authority Vectors

- Set \vec{a}_0 uniformly
- $\vec{a}_i = \frac{1}{c_i} A^T \cdot \vec{h}_{i-1}, i \geq 1$

	\vec{a}_1	\vec{a}_2	\vec{a}_3	\vec{a}_4	\vec{a}_5	\vec{a}_6	\vec{a}_7
d_0	0.06	0.09	0.10	0.10	0.10	0.10	0.10
d_1	0.06	0.03	0.01	0.01	0.01	0.01	0.01
d_2	0.19	0.14	0.13	0.12	0.12	0.12	0.12
d_3	0.31	0.43	0.46	0.46	0.46	0.47	0.47
d_4	0.13	0.14	0.16	0.16	0.16	0.16	0.16
d_5	0.06	0.03	0.02	0.01	0.01	0.01	0.01
d_6	0.19	0.14	0.13	0.13	0.13	0.13	0.13



Top-ranked Pages

- Pages with highest in-degree: d_2, d_3, d_6
- Pages with highest out-degree: d_2, d_6
- Pages with highest PageRank: d_6
- Pages with highest hub score: d_6 (close: d_2)
- Pages with highest authority score: d_3



PageRank vs. HITS

- PageRank can be precomputed, HITS has to be computed at query time
 - HITS is too expensive in most application scenarios.
- PageRank and HITS are different in
 - the eigenproblem formalization
 - the set of pages to apply the formalization to.
- On the web, a good hub almost always is also a good authority.



Outline

1 Link Analysis

- PageRank
 - Topic-Sensitive PageRank
- HITS
- Demo

2 Community Detection

- Introduction
- Methods
 - Node-Centric Community Detection
 - Group-Centric Community Detection
 - Network-Centric Community Detection
 - Hierarchy-Centric Community Detection
- Summary

3 References



R Package for PageRank

Resources

- Package: <http://cran.r-project.org/web/packages/igraph/index.html>
- Function: <http://igraph.sourceforge.net/doc/R/page.rank.html>
- Manual: <http://cran.r-project.org/web/packages/igraph/igraph.pdf>
- Author: Tamas Nepusz and Gabor Csardi

Description

page.rank igraph: Calculates the Google PageRank for the specified vertices.



How to use

Usage

```
page.rank (graph, vids = V(graph), directed = TRUE, damping = 0.85,  
weights = NULL, options = igraph.arnpack.default)
```

Example

```
g = random.graph.game(20, 5/20, directed=TRUE)  
page.rank(g)  
g2 = graph.star(10)  
page.rank(g2)
```



R Package for HITS

Resources

- Package: <http://cran.r-project.org/web/packages/igraph/index.html>
- Function: <http://igraph.sourceforge.net/doc/R/kleinberg.html>
- Manual: <http://cran.r-project.org/web/packages/igraph/igraph.pdf>
- Author: Gabor Csardi

Description

kleinberg igraph: Kleinberg's hub and authority scores.



How to use

Usage

```
authority.score (graph, scale = TRUE, options = igraph.arpack.default)  
hub.score (graph, scale = TRUE, options = igraph.arpack.default)
```

Example

```
#An in-star  
g = graph.star(10)  
hub.score(g)  
authority.score(g)  
#A ring  
g2 = graph.ring(10)  
hub.score(g2)  
authority.score(g2)
```



Outline

- 1 Link Analysis
 - PageRank
 - Topic-Sensitive PageRank
 - HITS
 - Demo
- 2 Community Detection
 - Introduction
 - Methods
 - Node-Centric Community Detection
 - Group-Centric Community Detection
 - Network-Centric Community Detection
 - Hierarchy-Centric Community Detection
 - Summary
- 3 References



Outline

- 1 Link Analysis
 - PageRank
 - Topic-Sensitive PageRank
 - HITS
 - Demo
- 2 Community Detection
 - Introduction
 - Methods
 - Node-Centric Community Detection
 - Group-Centric Community Detection
 - Network-Centric Community Detection
 - Hierarchy-Centric Community Detection
 - Summary
- 3 References



Communities

Community

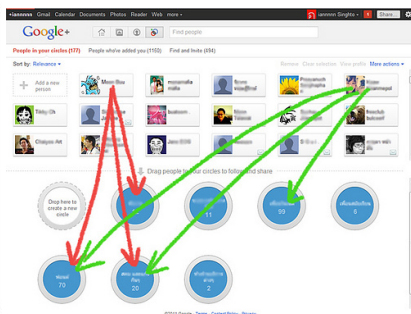
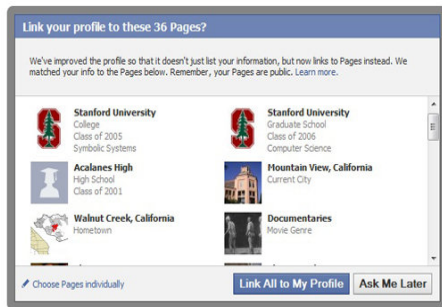
A community is formed by individuals such that those within a group **interact** with each other **more frequently** than with those outside the group.

- Users form communities in social media
- Community is formed through frequent interacting
- A set of users who do not interact with each other is not a community

Why Communities Are Formed?

- Human beings are social
- Social media are easy to use
 - People's social lives are easy to extend with the help of social media
- People connect with friends, relatives, colleges, etc. in the physical world as well as online

Examples of Communities



Community Detection

Two Types of Users

- ① **Explicit Groups:** Formed by user subscriptions
 - E.g., Groups in Facebook
- ② **Implicit Groups:** implicitly formed by social interactions
 - E.g., Community question answering

Community Detection

Discovering groups in a network where individuals' group memberships are not explicitly given



Outline

- 1 Link Analysis
 - PageRank
 - Topic-Sensitive PageRank
 - HITS
 - Demo
- 2 Community Detection
 - Introduction
 - Methods
 - Node-Centric Community Detection
 - Group-Centric Community Detection
 - Network-Centric Community Detection
 - Hierarchy-Centric Community Detection
 - Summary
- 3 References



Approaches

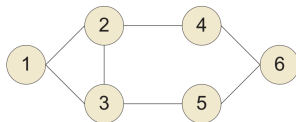
Four categories

- Node-centric approach
 - Each **node** in a group satisfies certain properties
- Group-centric approach
 - Consider the connections **inside a group** as a whole
- **Network-centric approach**
 - Partition nodes of a **network** into several disjoint sets
- Hierarchy-centric approach
 - Build a **hierarchical structure** of communities based on network topology



Node-Centric Community Detection

- Nodes satisfying certain properties within a group
 - Complete mutuality
 - cliques: A clique is a maximum complete subgraph in which all nodes are adjacent to each other
 - Reachability of members
 - k-clique: A k-clique is a maximal subgraph in which the largest geodesic distance between any two nodes is no greater than k
 - k-clan: The geodesic distance **within the group** to be no greater than k



cliques: {1, 2, 3}

2-cliques: {1, 2, 3, 4, 5}, {2, 3, 4, 5, 6}

2-clubs: {1, 2, 3, 4}, {1, 2, 3, 5}, {2, 3, 4, 5, 6}



Group-Centric Community Detection

Density-Based Groups

- It is acceptable for some nodes to have low connectivity
- The whole group satisfies a certain condition
 - E.g., the group density \geq a given threshold
- A subgraph $G_s(V_s, E_s)$ is γ -dense (*quasi-clique*, Abello et al., 2002) if

$$\frac{E_s}{V_s(V_s - 1)/2} \geq \gamma$$

- Greedy search through recursive pruning
 - Local search: sample a subgraph and find a maximum γ -dense quasi-clique (say, of size k)
 - Heuristic pruning: remove nodes with degree less than $k \cdot \gamma$



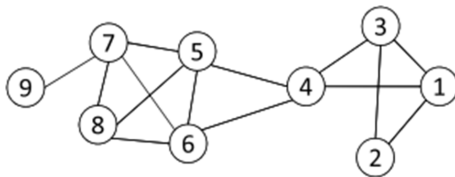
Network-Centric Community Detection

- Consider the **global topology** of a network
- Partition nodes of a network into **disjoint sets**
- Optimize a criterion defined over a partition rather than over one group
- Approaches:
 - Clustering based on vertex similarity
 - Latent space models (multi-dimensional scaling)
 - Block model approximation
 - Spectral clustering
 - Modularity maximization



Clustering Based on Vertex Similarity

- Vertex similarity is defined in terms of **the similarity of their social circles**
- Structural equivalence: two nodes are structurally equivalent iff they are connecting to the same set of actors

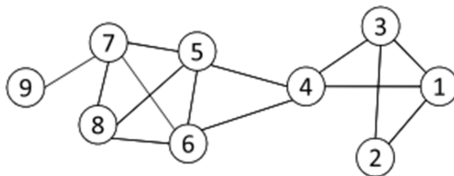


- Nodes 1 and 3 are structurally equivalent; So are nodes 5 and 6.
- Structural equivalence is too restrict for practical use
- Apply k-means to find communities



Vertex Similarity Measurements

- Cosine Similarity: $\text{Cosine}(v_i, v_j) = \frac{|N_i \cap N_j|}{\sqrt{|N_i| \cdot |N_j|}}$
- Jaccard Similarity: $\text{Jaccard}(v_i, v_j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}$



$$\text{Cosine}(4, 6) = \frac{1}{\sqrt{4 \cdot 4}} = \frac{1}{4}$$

$$\text{Jaccard}(4, 6) = \frac{|\{5\}|}{|\{1, 3, 4, 5, 6, 7, 8\}|} = \frac{1}{7}$$



Latent Space Models

- Map nodes into a low-dimensional Euclidean space such that the proximity between nodes based on network connectivity are kept in the new space
- Multi-dimensional scaling (MDS)
 - Given a network, construct a proximity matrix $P \in \mathbb{R}^{n \times n}$ representing the pairwise distance between nodes
 - Let $S \in \mathbb{R}^{n \times k}$ denote the coordinates of nodes in the low-dimensional space

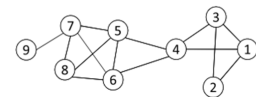
$$SS^T \approx -\frac{1}{2}(I - \frac{1}{n}ee^T)(P \circ P)(I - \frac{1}{n}ee^T) = \tilde{P},$$

where \circ is the element-wise matrix multiplication

- Objective: $\min \|SS^T - \tilde{P}\|_F^2$
 - Let $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_k)$ (the top-k eigenvalues of \tilde{P}), V the top-k eigenvectors
 - Solution: $S = \Lambda V^{1/2}$
- Apply k-means to S to obtain communities



Example of MDS



geodesic
distance

$$P = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 2 & 3 & 3 & 4 \\ 1 & 0 & 1 & 2 & 3 & 3 & 4 & 4 & 5 \\ 1 & 1 & 0 & 1 & 2 & 2 & 3 & 3 & 4 \\ 1 & 2 & 1 & 0 & 1 & 1 & 2 & 2 & 3 \\ 2 & 3 & 2 & 1 & 0 & 1 & 1 & 1 & 2 \\ 2 & 3 & 2 & 1 & 1 & 0 & 1 & 1 & 2 \\ 3 & 4 & 3 & 2 & 1 & 1 & 0 & 1 & 1 \\ 3 & 4 & 3 & 2 & 1 & 1 & 1 & 0 & 2 \\ 4 & 5 & 4 & 3 & 2 & 2 & 1 & 2 & 0 \end{bmatrix}$$



$$V = \begin{bmatrix} -0.33 & 0.05 \\ -0.55 & 0.14 \\ -0.33 & 0.05 \\ -0.11 & -0.01 \\ 0.10 & -0.06 \\ 0.10 & -0.06 \\ 0.32 & 0.11 \\ 0.28 & -0.79 \\ 0.52 & 0.58 \end{bmatrix}, \Lambda = \begin{bmatrix} 21.56 & 0 \\ 0 & 1.46 \end{bmatrix}, S = V\Lambda^{1/2} = \begin{bmatrix} -1.51 & 0.06 \\ -2.56 & 0.17 \\ -1.51 & 0.06 \\ -0.53 & -0.01 \\ 0.47 & -0.08 \\ 0.47 & -0.08 \\ 1.47 & 0.14 \\ 1.29 & -0.95 \\ 2.42 & 0.70 \end{bmatrix}$$

Two communities:

{1, 2, 3, 4} and {5, 6, 7, 8, 9}

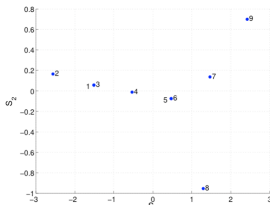


Figure: From <http://dmml.asu.edu/cdm/slides/chapter3.pdf>



Block Model Approximation

Adjacency Matrix									
-	1	1	1	0	0	0	0	0	0
1	-	1	0	0	0	0	0	0	0
1	1	-	1	0	0	0	0	0	0
1	0	1	-	1	1	0	0	0	0
0	0	0	1	-	1	1	1	0	0
0	0	0	1	1	-	1	1	0	0
0	0	0	0	1	1	-	1	1	0
0	0	0	0	1	1	1	-	1	1
0	0	0	0	1	1	1	1	-	0
0	0	0	0	0	0	1	0	-	-

Ideal Block Structure									
1	1	1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1
0	0	0	0	1	1	1	1	1	1

- Objective: Minimize the difference between an adjacency matrix and a block structure

$$\min_{S, \Sigma} \|A - S\Sigma S^T\|_F^2$$

where $S \in \{0, 1\}^{n \times k}$, and $\Sigma \in R^{k \times k}$ is diagonal

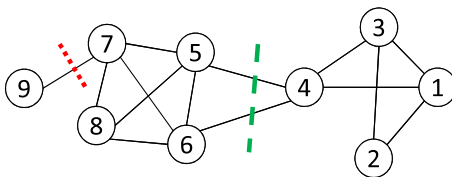
- Challenge: S is discrete, difficult to solve
- Relaxation: Allow S to be continuous satisfying $S^T S = I_k$
- Solution: the top k eigenvectors of A
- Apply k-means to S to obtain communities



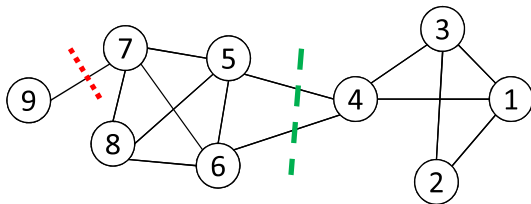
Cut

- Community detection \rightarrow graph partition \rightarrow **minimum cut** problem
- Cut: A partition of vertices of a graph into two disjoint sets
- Minimum cut: Find a graph partition such that the number of edges among different sets is minimized
 - Minimum cut often returns an **imbalanced partition**, e.g., node 9
 - Consider community size
 - Let C_i denote a community, $|C_i|$ represent the number of nodes in C_i , and $vol(C_i)$ measure the total degrees of nodes in C_i

$$RatioCut(\pi) = \frac{1}{k} \sum_{i=1}^k \frac{cut(C_i, \bar{C}_i)}{|C_i|} \quad NormalizedCut(\pi) = \frac{1}{k} \sum_{i=1}^k \frac{cut(C_i, \bar{C}_i)}{vol(C_i)}$$



Ratio Cut & Normalized Cut Example



- For partition in red (π_1)
 - $RatioCut(\pi_1) = \frac{1}{2}(\frac{1}{1} + \frac{1}{8}) = 0.56$
 - $NormalizedCut(\pi_1) = \frac{1}{2}(\frac{1}{1} + \frac{1}{27}) = 0.52$
- For partition in green (π_2)
 - $RatioCut(\pi_2) = \frac{1}{2}(\frac{2}{4} + \frac{2}{5}) = 0.45 < RatioCut(\pi_1)$
 - $NormalizedCut(\pi_2) = \frac{1}{2}(\frac{2}{12} + \frac{2}{16}) = 0.15 < NormalizedCut(\pi_1)$
- Smaller values mean more balanced partition



Spectral Clustering

- Finding the minimum ratio cut and normalized cut are NP-hard
- An approximation is **spectral clustering**

$$\min_{S \in \{0,1\}^{n \times k}} \text{Tr}(S^T \tilde{L} S) \quad \text{s.t.}, S^T S = I_k$$

- \tilde{L} is the (normalized) **Graph Laplacian**

$$\tilde{L} = D - A$$

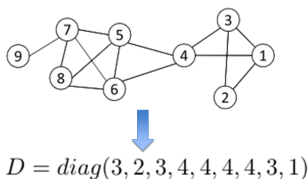
$$\text{Normalized } L = I - D^{-1/2} A D^{-1/2}$$

$$D = \text{diag}\{d_1, d_2, \dots, d_n\}$$

- Solution: S are the eigenvectors of L with smallest eigenvalues (except the first one)
- Apply k-means to S to obtain communities



Spectral Clustering Example



$$\tilde{L} = D - A = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 4 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 4 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 4 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 4 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix} \Rightarrow S = \begin{bmatrix} 0.33 & -0.38 \\ 0.33 & -0.48 \\ 0.33 & -0.38 \\ 0.33 & -0.12 \\ 0.33 & 0.16 \\ 0.33 & 0.16 \\ 0.33 & 0.30 \\ 0.33 & 0.24 \\ 0.33 & 0.51 \end{bmatrix}$$

Two communities:
 $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8, 9\}$



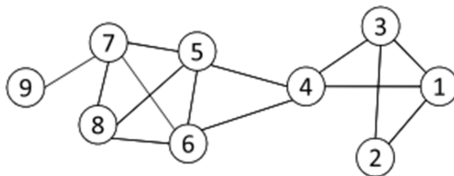
k-means

Figure: From <http://dmml.asu.edu/cdm/slides/chapter3.pdf>



Modularity Maximization

- **Modularity** measures the network interactions compared with the expected random connections
- In a network with m edges, for two nodes with degree d_i and d_j , the expected random connections are $\frac{d_i d_j}{2m}$



- The expected number of edges between nodes 1 and 2 is $3 \times 2 / (2 \times 14) = 3/14$
- Strength of a community: $\sum_{i \in C, j \in C} (A_{ij} - d_i d_j / 2m)$
- Modularity: $Q = \frac{1}{2m} \sum_C \sum_{i \in C, j \in C} (A_{ij} - d_i d_j / 2m)$



Matrix Formation

- The modularity maximization can be reformed in the matrix form:

$$Q = \frac{1}{2m} \text{Tr}(S^T B S)$$

- B is the modularity matrix

$$B_{ij} = A_{ij} - d_i d_j / 2m$$

- Solution: top eigenvectors of the modularity matrix
- Modularity: $Q = \frac{1}{2m} \sum_C \sum_{i \in C, j \in C} (A_{ij} - d_i d_j / 2m)$
- Apply k-means to S to obtain communities



Modularity Maximization Example

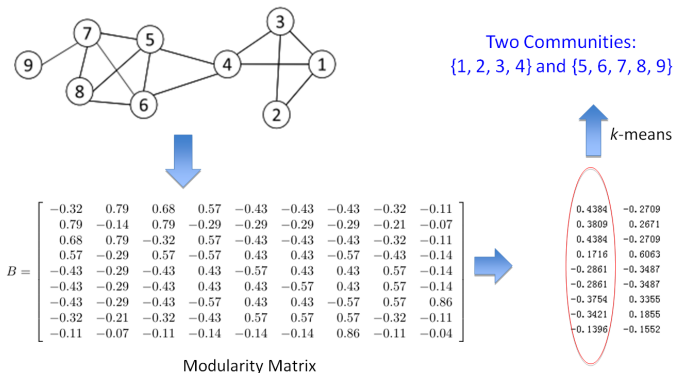


Figure: From <http://dmml.asu.edu/cdm/slides/chapter3.pdf>



A Unified Process

- Goal of network-centric community detection: Partition network nodes into several disjoint sets

$$\text{Utility Matrix } M = \begin{cases} \tilde{P} & (\text{latent space models}) \\ A & (\text{block model approximation}) \\ \tilde{L} & (\text{spectral clustering}) \\ B & (\text{modularity maximization}) \end{cases}$$

- Limitation: The number of communities requires manual setting



Hierarchy-Centric Community Detection

- Goal: Build a hierarchical structure of communities based on network topology
- Facilitate the analysis at different resolutions
- Approaches:
 - Top-down: Divisive hierarchical clustering
 - Bottom-up: Agglomerative hierarchical clustering



Outline

- 1 Link Analysis
 - PageRank
 - Topic-Sensitive PageRank
 - HITS
 - Demo
- 2 Community Detection
 - Introduction
 - Methods
 - Node-Centric Community Detection
 - Group-Centric Community Detection
 - Network-Centric Community Detection
 - Hierarchy-Centric Community Detection
 - Summary
- 3 References



Summary

- Goal: Discovering groups in a network where individuals' group memberships are not explicitly given
- Approaches
 - Node-centric approach
 - Each **node** in a group satisfies certain properties
 - Group-centric approach
 - Consider the connections **inside a group** as a whole
 - Network-centric approach
 - Partition nodes of a **network** into several disjoint sets
 - Hierarchy-centric approach
 - Build a **hierarchical structure** of communities based on network topology
- Which one to choose?
- Scalability issue in real applications



Outline

- 1 Link Analysis
 - PageRank
 - Topic-Sensitive PageRank
 - HITS
 - Demo
- 2 Community Detection
 - Introduction
 - Methods
 - Node-Centric Community Detection
 - Group-Centric Community Detection
 - Network-Centric Community Detection
 - Hierarchy-Centric Community Detection
 - Summary
- 3 References



References

- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, An Introduction to Information Retrieval (Book), 2008.
- Brin, S. and Page, L., The Anatomy of a Large-Scale Hypertextual Web Search Engine, 1998.
- Jon M. Kleinberg, Authoritative sources in a hyperlinked environment, 1999.
- Taher H. Haveliwala, Topic-sensitive PageRank, 2002.



References

- L. Tang and H. Li, Community Detection and Mining in Social Media (Book), 2010.
- H. Liu, L. Tang, and N. Agarwal, Community Detection and Behavior Study for Social Computing (Tutorial), 2009.
- R. Andersen and K. J. Lang, Communities from seed sets, WWW, 2006:
- S. Fortunato, Community detection in graphs, 2010.



References

- D. Gibson, R. Kumar, and A. Tomkins, Discovering large dense subgraphs in massive graphs, VLDB, 2005.
- M. S. Handcock, A. E. Raftery, and J. M. Tantrum, Model-based clustering for social networks, 2007.
- P. D. Hoff, A. E. Raftery, and M. S. Handcock, Latent space approaches to social network analysis, 2002.
- A. Java, A. Joshi, and T. Finin. Detecting Communities via Simultaneous Clustering of Graphs and Folksonomies, WebKDD, 2008.



References

- R. Kumar, J. Novak, and A. Tomkins, Structure and evolution of online social networks, KDD, 2006.
- Y.-R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. Analyzing communities and their evolutions in dynamic social networks, TKDD, 2009.
- B. Long, Z.M. Zhang, X.Wu, and P. S. Yu, Spectral clustering for multi-type relational data, ICML, 2006.
- B. Long, P. S. Yu, and Z.M. Zhang, A general model for multiple view unsupervised learning, SDM, 2008.
- I. Borg and P. Groenen, Modern Multidimensional Scaling: theory and applications (2nd ed.) (Book), 2005.



QA

Thanks for your attention!

