
Large Scale Community Detection for Social Computing

with Implementations in Hadoop

Lei Tang

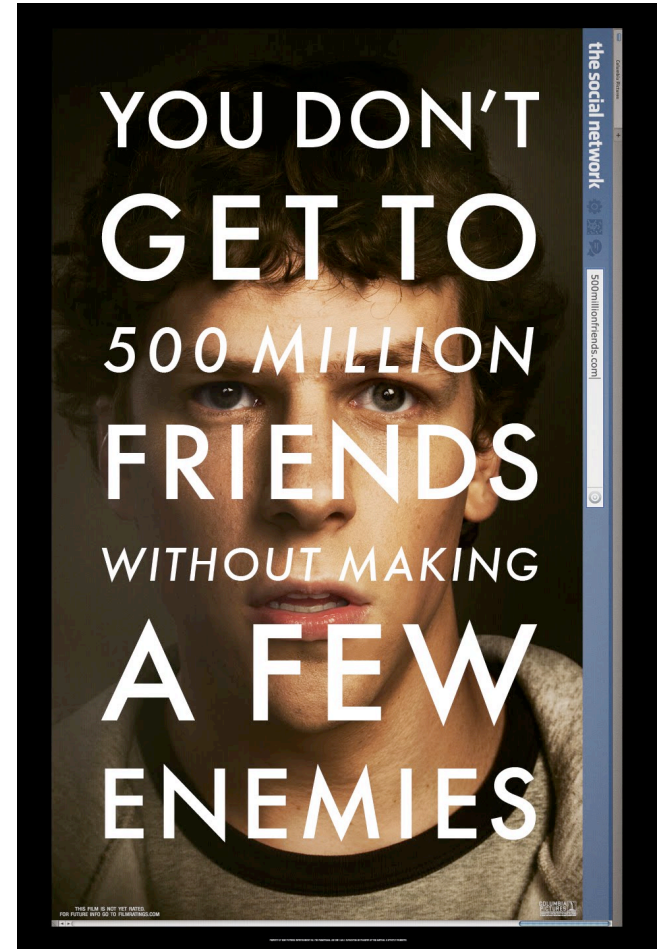
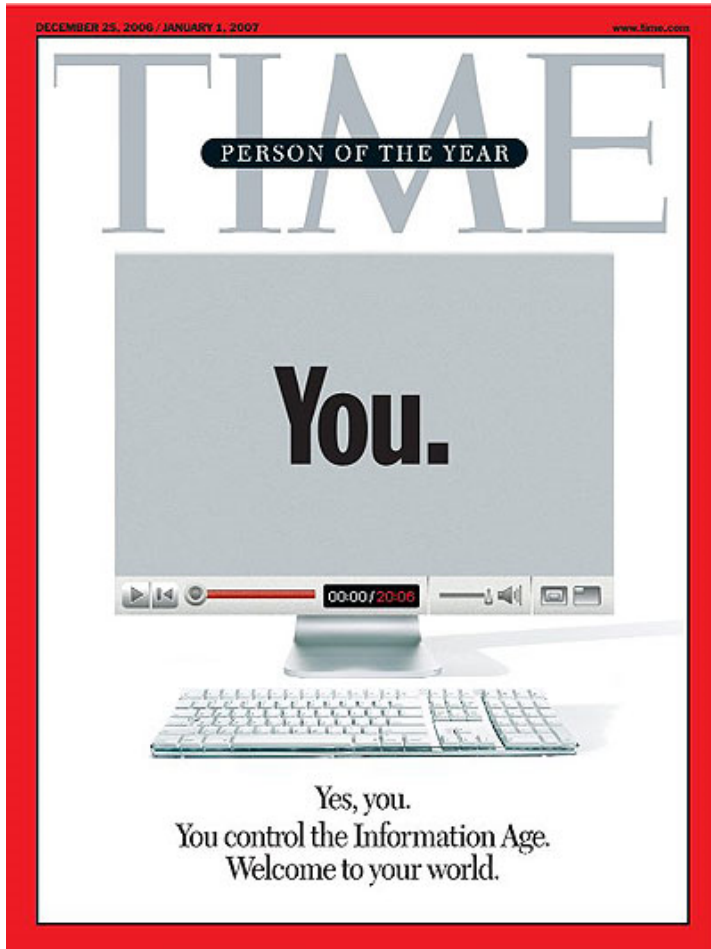
Yahoo! Labs

February 23, 2011

SDForum Software Architecture and Platform

Outline

- Introduction to Social Media and Social Computing
 - Principles of Community Detection
 - Large-Scale Community Detection in Hadoop
 - Applications of Community Detection for Social Computing
-



PARTICIPATING WEB AND SOCIAL MEDIA

Traditional Media

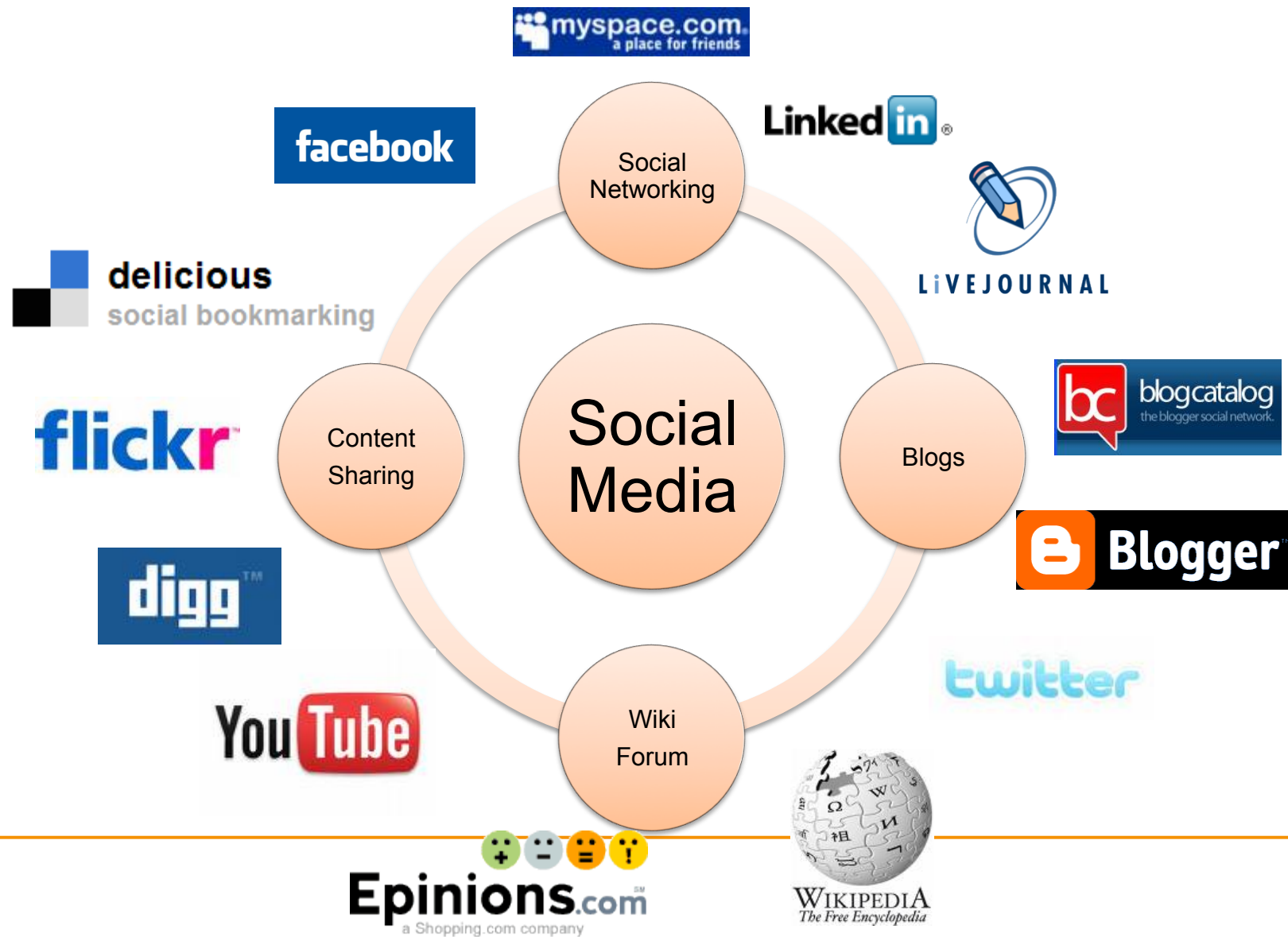


Broadcast Media: One-to-Many



Communication Media: One-to-One

Social Media: Many-to-Many



Characteristics of Social Media

- Everyone can be a media outlet
- Disappearing of communications barrier
 - Rich User Interaction
 - User-Generated Contents
 - User Enriched Contents
 - User developed widgets
 - Collaborative environment
 - Collective Wisdom
 - Long Tail



Broadcast Media
Filter, then Publish



Social Media
Publish, then Filter

Top 20 Most Visited Websites

- Internet traffic report by Alexa on August 3, 2010

Table 1.2: Top 20 Websites in the US

Rank	Site	Rank	Site
1	google.com	11	blogger.com
2	facebook.com	12	msn.com
3	yahoo.com	13	myspace.com
4	youtube.com	14	go.com
5	amazon.com	15	bing.com
6	wikipedia.org	16	aol.com
7	craigslist.org	17	linkedin.com
8	twitter.com	18	cnn.com
9	ebay.com	19	espn.go.com
10	live.com	20	wordpress.com

- 40% of the top 20 websites are social media sites

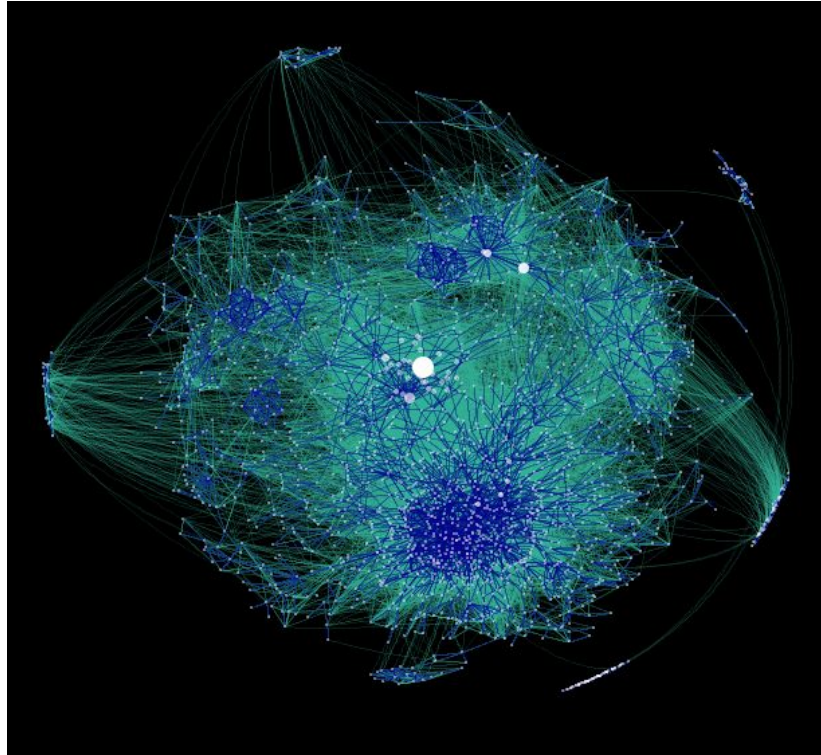
Social Media's Important Role



"social networks will complement, and may replace, some government functions,"
Presidential Election, 2008



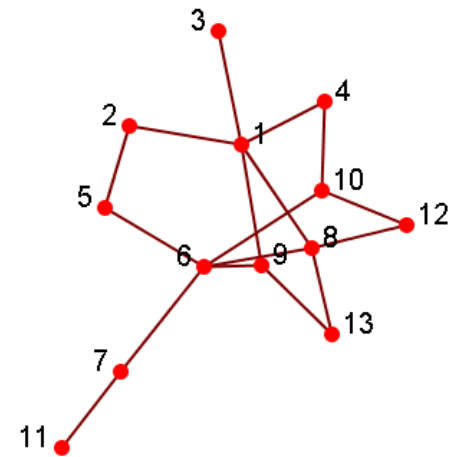
Egypt Protest, 2011



SOCIAL NETWORKS AND DATA MINING

Social Networks

- A social structure made of nodes (individuals or organizations) that are related to each other by various interdependencies like friendship, kinship, etc.
- Graphical representation
 - Nodes = members
 - Edges = relationships
- Various realizations
 - Social bookmarking (Del.icio.us)
 - Friendship networks (facebook, myspace)
 - Blogosphere
 - Media Sharing (Flickr, Youtube)
 - Folksonomies

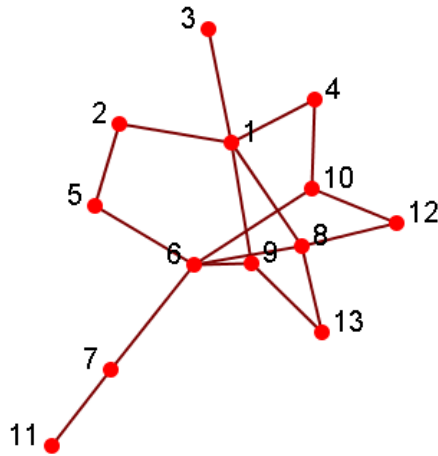


Social Computing and Data Mining

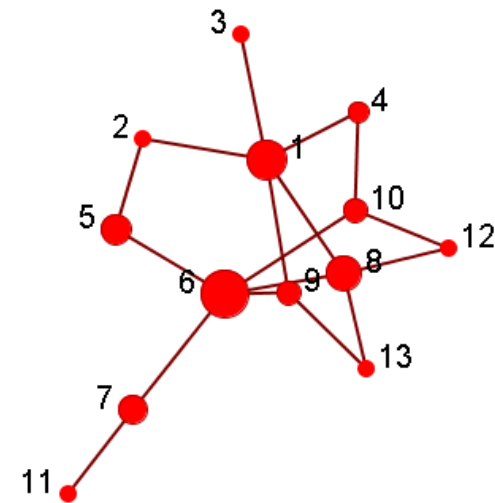
- **Social computing** is concerned with the study of social behavior and social context based on computational systems.
 - **Data Mining Related Tasks**
 - ❑ Centrality Analysis
 - ❑ Community Detection
 - ❑ Classification
 - ❑ Link Prediction
 - ❑ Viral Marketing
 - ❑ Network Modeling
-

Centrality Analysis/Influence Study

- Identify the most **important** actors in a social network
- Given: a social network
- Output: a list of top-ranking nodes



Top 5 important nodes:
6, 1, 8, 5, 10

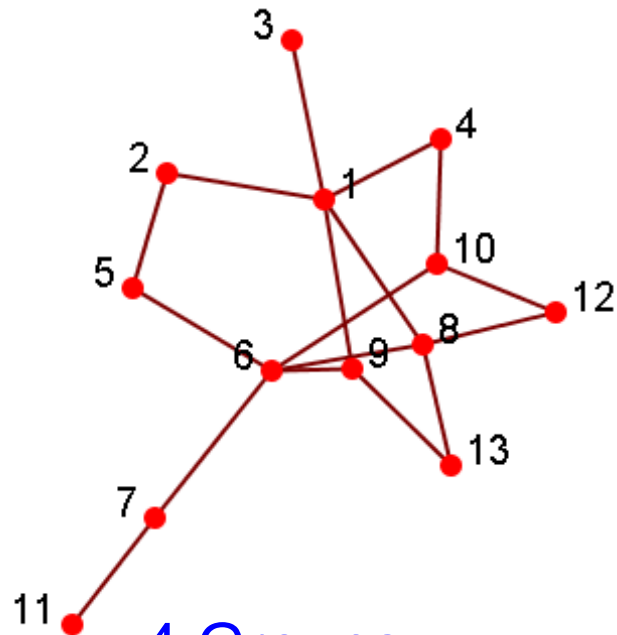


(Nodes resized by
Importance)

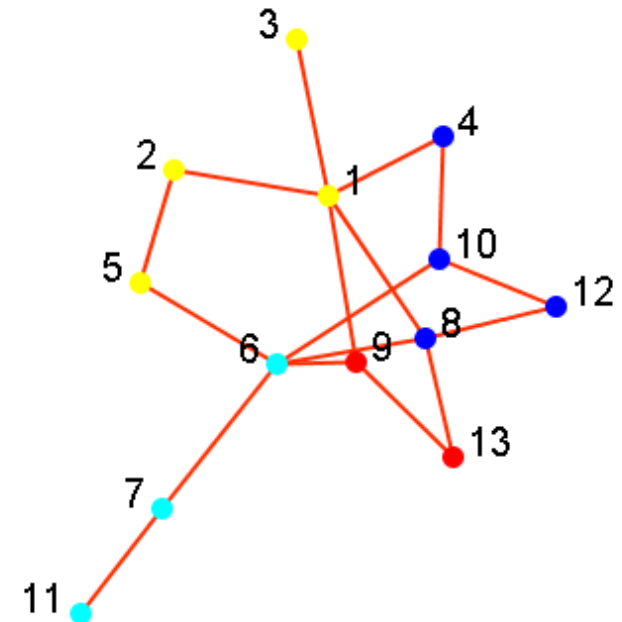
Community Detection

- A community is a set of nodes between which the interactions are (relatively) frequent
a.k.a. group, subgroup, module, cluster
 - Community detection
a.k.a. grouping, clustering, finding cohesive subgroups
 - ❑ Given: a social network
 - ❑ Output: community membership of (some) actors
 - Applications
 - ❑ Understanding the interactions between people
 - ❑ Visualizing and navigating huge networks
 - ❑ Forming the basis for other tasks such as data mining
-

Visualization after Grouping



4 Groups:
{1,2,3,5}
{4,8,10,12}
{6,7,11}
{9,13}

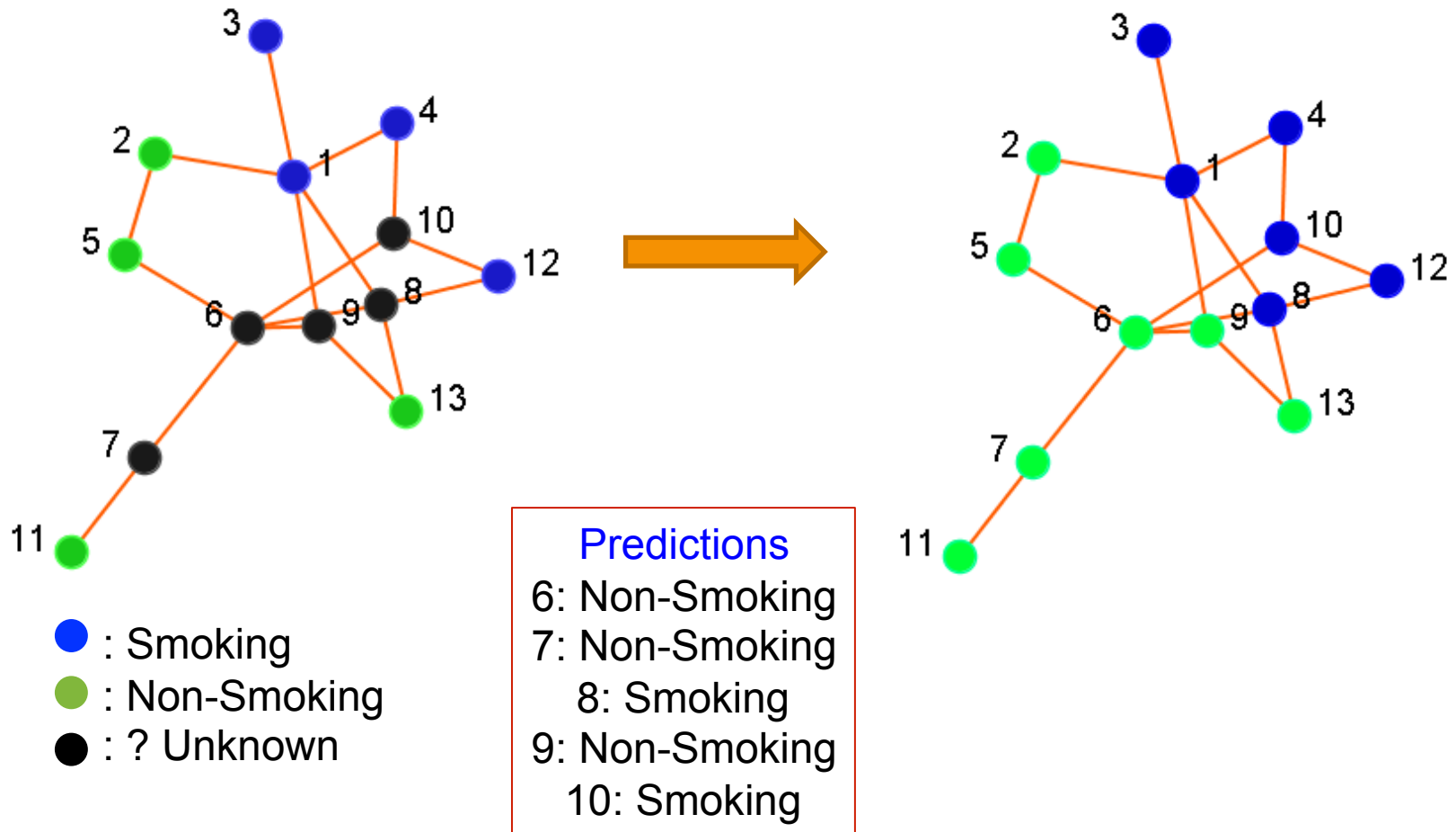


(Nodes colored by
Community Membership)

Classification

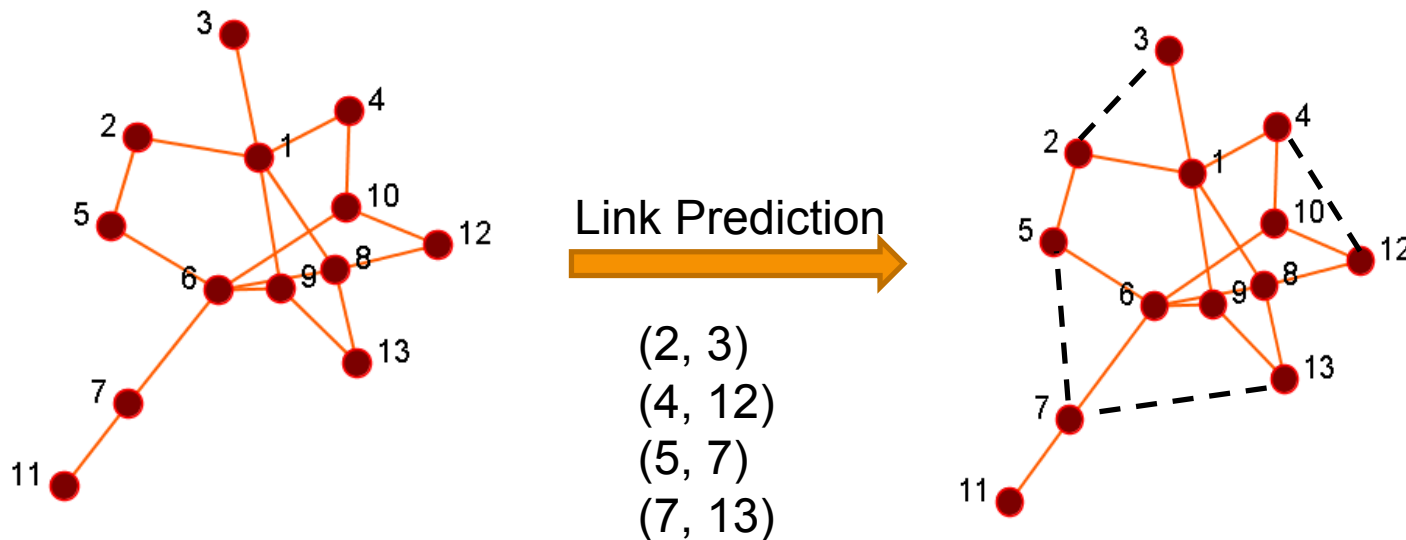
- User Preference or Behavior can be represented as class labels
 - Whether or not clicking on an ad
 - Whether or not interested in certain topics
 - Subscribed to certain political views
 - Like/Dislike a product
 - Given
 - A social network
 - Labels of some actors in the network
 - Output
 - Labels of remaining actors in the network
-

Visualization after Prediction



Link Prediction

- Given a social network, predict which nodes are likely to get connected
- Output a list of (ranked) pairs of nodes
- Example: Friend recommendation in Facebook

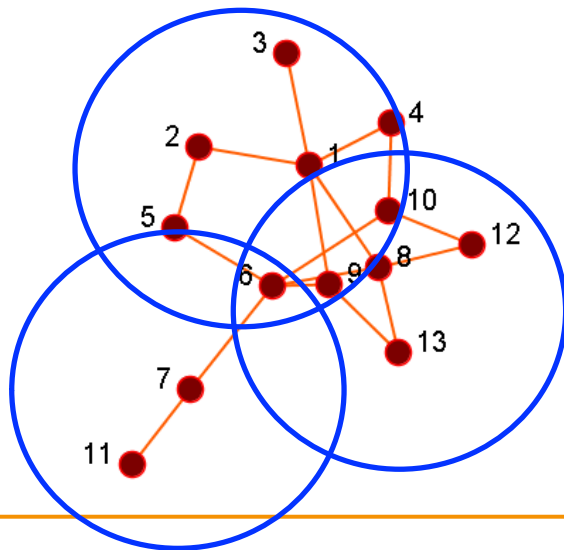


Viral Marketing/Outbreak Detection

- Users have different social capital (or network values) within a social network, hence, how can one make best use of this information?
 - **Viral Marketing:** find out a set of users to provide coupons and promotions to influence other people in the network so my benefit is maximized
 - **Outbreak Detection:** monitor a set of nodes that can help detect outbreaks or interrupt the infection spreading (e.g., H1N1 flu)
 - **Goal:** given a limited budget, how to maximize the overall benefit?
-

An Example of Viral Marketing

- Find the coverage of the whole network of nodes with the minimum number of nodes
- How to realize it – an example
 - **Basic Greedy Selection:** Select the node that maximizes the utility, remove the node and then repeat



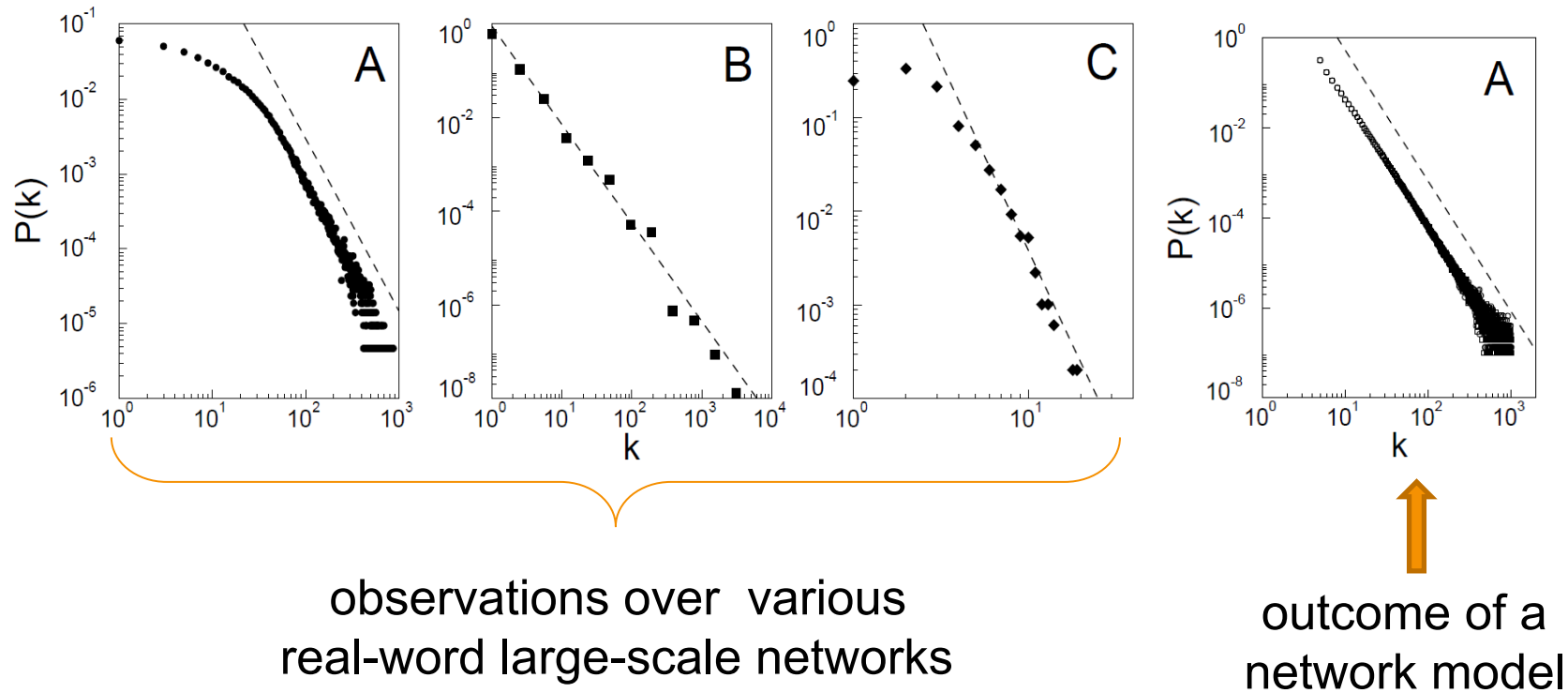
- Select Node 1
- Select Node 8
- Select Node 7

Node 7 is not a node with high centrality!

Network Modeling

- Large Networks demonstrate statistical patterns:
 - ❑ Small-world effect (e.g., 6 degrees of separation)
 - ❑ Power-law distribution (a.k.a. scale-free distribution)
 - ❑ Community structure (high clustering coefficient)
 - Model the network dynamics
 - ❑ Find a mechanism such that the statistical patterns observed in large-scale networks can be reproduced.
 - ❑ Examples: random graph, preferential attachment process
 - Used for simulation to understand network properties
 - ❑ Thomas Shelling's famous [simulation](#): What could cause the segregation of white and black people
 - ❑ Network robustness under attack
-

Comparing Network Models



observations over various
real-world large-scale networks

outcome of a
network model

(Figures borrowed from “*Emergence of Scaling in Random Networks*”)

Social Computing Applications

- Advertisizing via Social Networking
 - Behavior Modeling and Prediction
 - Epidemic Study
 - Collaborative Filtering
 - Crowd Mood Reader
 - Cultural Trend Monitoring
 - Visualization
 - Health 2.0
-










PRINCIPLES OF COMMUNITY DETECTION

Communities














- **Community:** “subsets of actors among whom there are relatively strong, direct, intense, frequent or positive ties.”
-- Wasserman and Faust, *Social Network Analysis, Methods and Applications*
 - Community is a set of actors interacting with each other *frequently*
 - e.g. people attending this conference
 - A set of people without interaction is **NOT** a community
 - e.g. people waiting for a bus at station but don't talk to each other
 - People form communities in Social Media
-

Example of Communities

Communities from Facebook

	<p>Name: Social Computing Type: Organizations Members: 14 members</p>
	<p>Name: Social Computing Type: Internet & Technology Members: 12 members</p>
	<p>Name: Social Computing Magazine Type: Internet & Technology Members: 34 members</p>
	<p>Name: Trustworthy Social Computing Type: Internet & Technology Members: 28 members</p>
	<p>Name: Social Computing for Business Type: Internet & Technology Members: 421 members</p>
	<p>Name: UCLA Social Sciences Computing Type: Internet & Technology Members: 22 members</p>
	<p>Name: Social Media and Computing Type: Organizations Members: 6 members</p>

Communities from Flickr

	<p>I * Urban LIFE in Metropolis //// 4,286 members 31 discussions 89,645 items Created 46 months ago Join? UrbanLIFE, People, Parties, Dance, Musik, Life, Love, Culture, Food and Everything what we could imagine by hearing that word URBANLIFE! Have some FUN! Please add... (more)</p>	
	<p>Islam Is The Way Of Life (Muslim World) 619 members 13 discussions 2,685 items Created 23 months ago Join? The word islām is derived from the Arabic verb aslama, which means to accept, surrender or submit. Thus, Islam means submission to and acceptance of God, and believers must... (more)</p>	
	<p>* THE CELEBRATION OF ~LIFE~ (Post1~Award1) [only living things] 4,871 members 22 discussions 40,519 items Created 21 months ago Join? WELCOME to THE CELEBRATION OF ~LIFE~ (Post1~Award1) PLEASE INVITE & COMMENT USING only THE CODES FOUND BELOW! ☆ ☆ This group is for sharing BEAUTIFUL, TOP QUALITY images... (more)</p>	
	<p>"Enjoy Life!" 2,027 members 10 discussions 39,916 items Created 23 months ago Join? There are lovely moments and adorable scenes in our lives. Some are in front of you, and some are just waiting to be discovered. A gaze from someone we love, might touch the... (more)</p>	
	<p>Baby's life 2,047 members 185 discussions 30,302 items Created 32 months ago Join? This group is designed to highlight milestones and important events in your baby's life (ie 1st time smiling/crawling/sitting in a high chair/reading/playing etc). It can also be... (more)</p>	Only group members pool
	<p>Pond Life 903 members 20 discussions 6,877 items Created 32 months ago Join? Pic of the week: chosen from the pool by the group admins. Nuphar by guus timpers Pond Life is a group for all aquatic flora and fauna. Koi ponds, wildlife ponds, garden ponds,... (more)</p>	
	<p>Second Life 10,288 members 773 discussions 257,870 items Created 61 months ago Join? Welcome to the Second Life pool, the biggest group on Flickr for residents/players of Second Life, the</p>	

Why Communities in Social Media?

- Human beings are **social**
 - Part of Interactions in social media is a glimpse of the physical world
 - People are connected to friends, relatives, and colleagues in the real world as well as online
 - Easy-to-use social media allows people to extend their social life in unprecedented ways
 - Difficult to meet friends in the physical world, but much easier to find friend online with similar interests
-

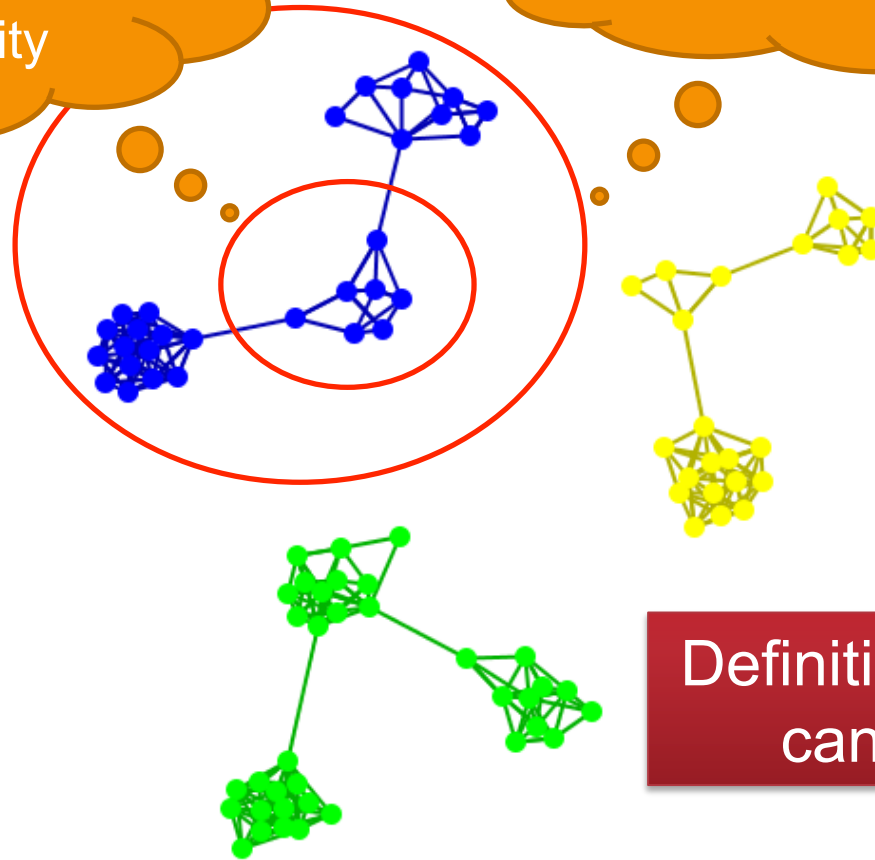
Community Detection

- **Community Detection:** “formalize the strong social groups based on the social network properties”
 - Some social media sites allow people to join **explicit** groups, is it necessary to extract groups based on network topology?
 - ❑ Not all sites provide community platform
 - ❑ Not all people join groups
 - Network interaction provides rich information about the relationship between users
 - ❑ Groups are *implicitly* formed
 - ❑ Can complement other kinds of information
 - ❑ Help network visualization and navigation
 - ❑ Provide basic information for other tasks
-

Subjectivity of Community Definition

A densely-knit community

Each component is a community

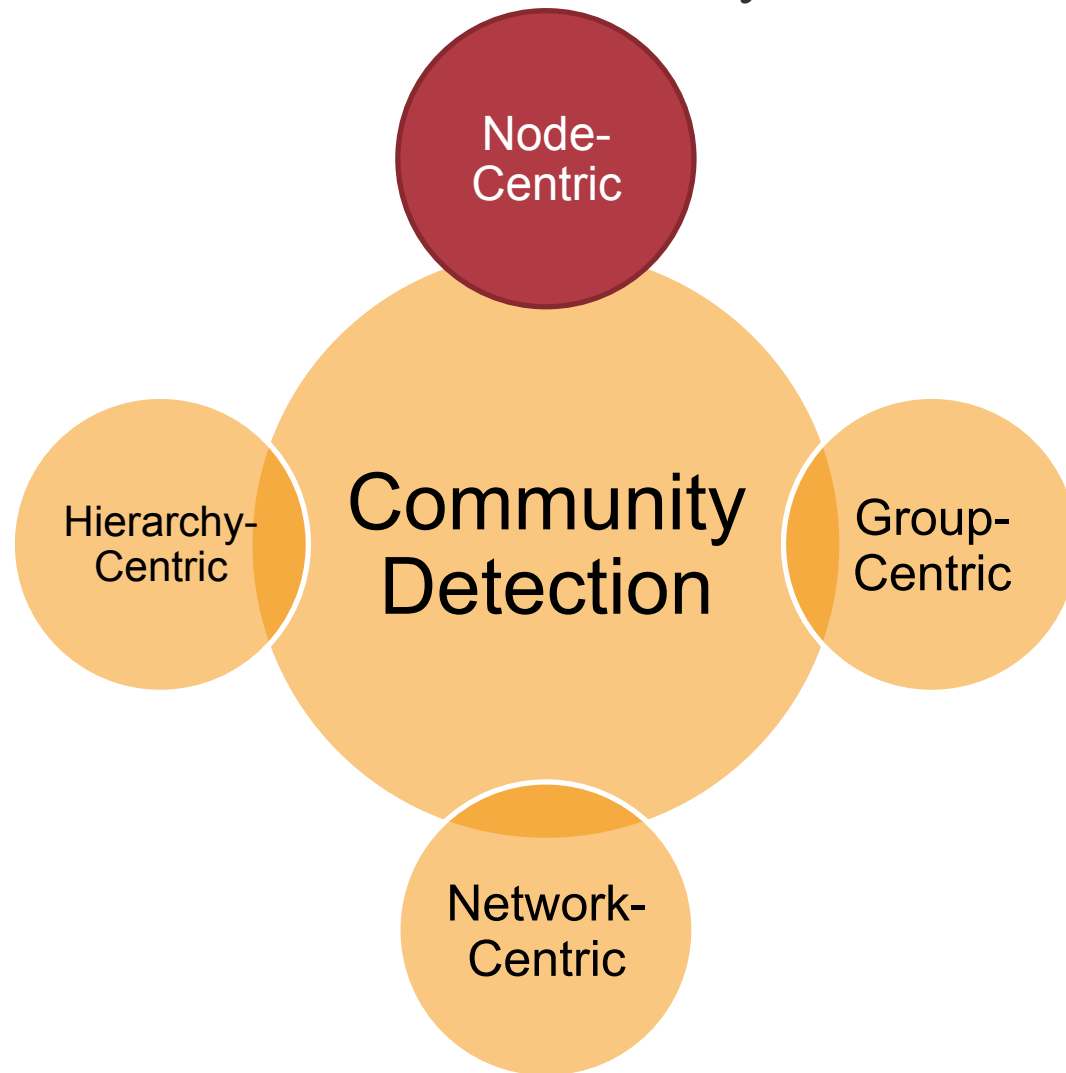


Definition of a community can be subjective.

Taxonomy of Community Criteria

- Criteria vary depending on the tasks
 - Roughly, community detection methods can be divided into 4 categories (not exclusive):
 - **Node-Centric Community**
 - **Each node** in a group satisfies certain properties
 - **Group-Centric Community**
 - Consider the connections **within a group** as a whole. The group has to satisfy certain properties without zooming into node-level
 - **Network-Centric Community**
 - Partition **the whole network** into several disjoint sets
 - **Hierarchy-Centric Community**
 - Construct a **hierarchical structure** of communities
-

Node-Centric Community Detection

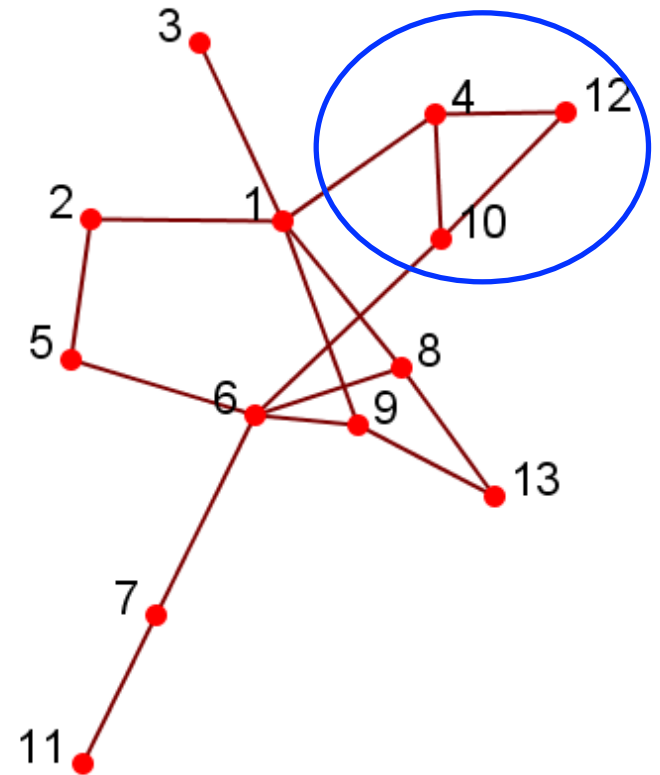


Node-Centric Community Detection

- Nodes satisfy different properties
 - Complete Mutuality
 - cliques
 - Reachability of members
 - k-clique, k-clan, k-club
 - Nodal degrees
 - k-plex, k-core
 - Relative frequency of Within-Outside Ties
 - LS sets, Lambda sets
 - Commonly used in traditional social network analysis
 - Here, we discuss some representative ones
-

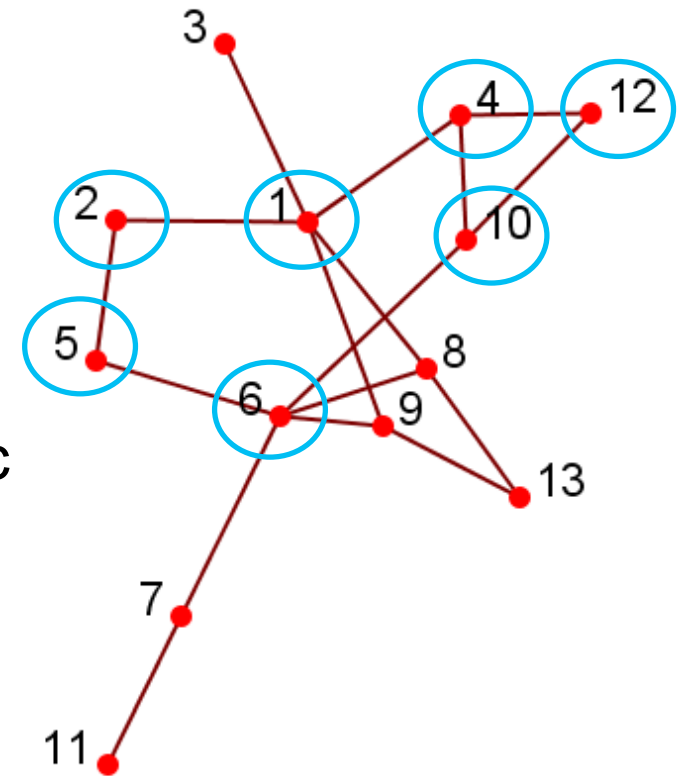
Complete Mutuality: Clique

- A maximal complete subgraph of three or more nodes all of which are adjacent to each other
- NP-hard to find the maximal clique
- Recursive pruning: To find a clique of size k , remove those nodes with less than $k-1$ degrees
- Very strict definition, unstable
- Normally use cliques as a core or seed to explore larger communities



Geodesic

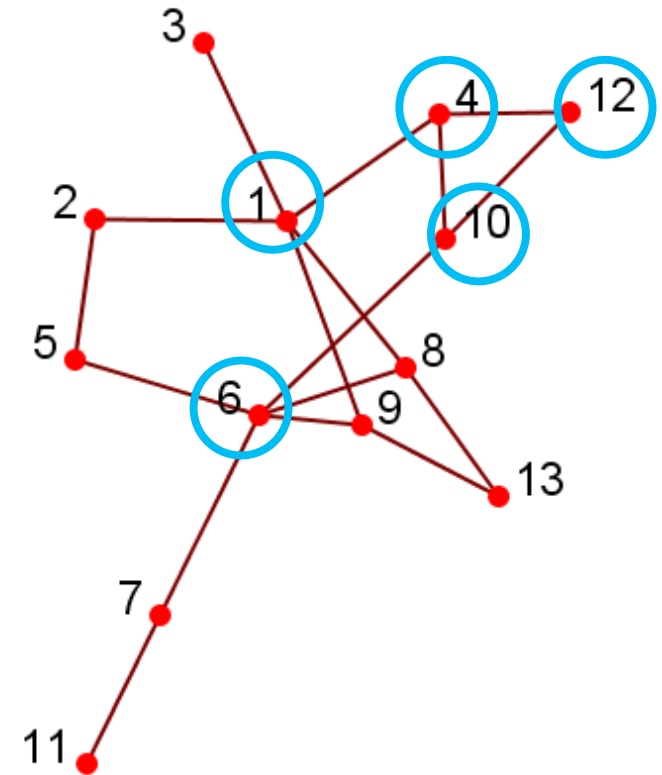
- Reachability is calibrated by the **Geodesic distance**
- **Geodesic**: a shortest path between two nodes (12 and 6)
 - Two paths: 12-4-1-2-5-6, 12-10-6
 - 12-10-6 is a geodesic
- **Geodesic distance**: #hops in geodesic between two nodes
 - e.g., $d(12, 6) = 2$, $d(3, 11) = 5$
- **Diameter**: the maximal geodesic distance for any 2 nodes in a network
 - #hops of the longest shortest path



Diameter = 5

Reachability: k-clique, k-club

- Any node in a group should be reachable in k hops
- **k-clique**: a maximal subgraph in which the largest geodesic distance between any nodes $\leq k$
- A k-clique can have diameter larger than k within the subgraph
 - e.g., 2-clique {12, 4, 10, 1, 6}
 - Within the subgraph $d(1, 6) = 3$
- **k-club**: a substructure of diameter $\leq k$
 - e.g., {1,2,5,6,8,9}, {12, 4, 10, 1} are 2-clubs



Nodal Degrees: k-plex, k-core

- Each node should have a certain number of connections to nodes within the group
 - **k-core**: a substructure that each node connects to at least k members within the group
 - **k-plex**: for a group with n_s nodes, each node should be adjacent no fewer than $n_s - k$ in the group
 - The definitions are complementary
 - A k-core is a $(n_s - k)$ -plex
 - Networks in social media tend to follow a power law distribution, are k-plex and k-core suitable for large-scale network analysis?
-

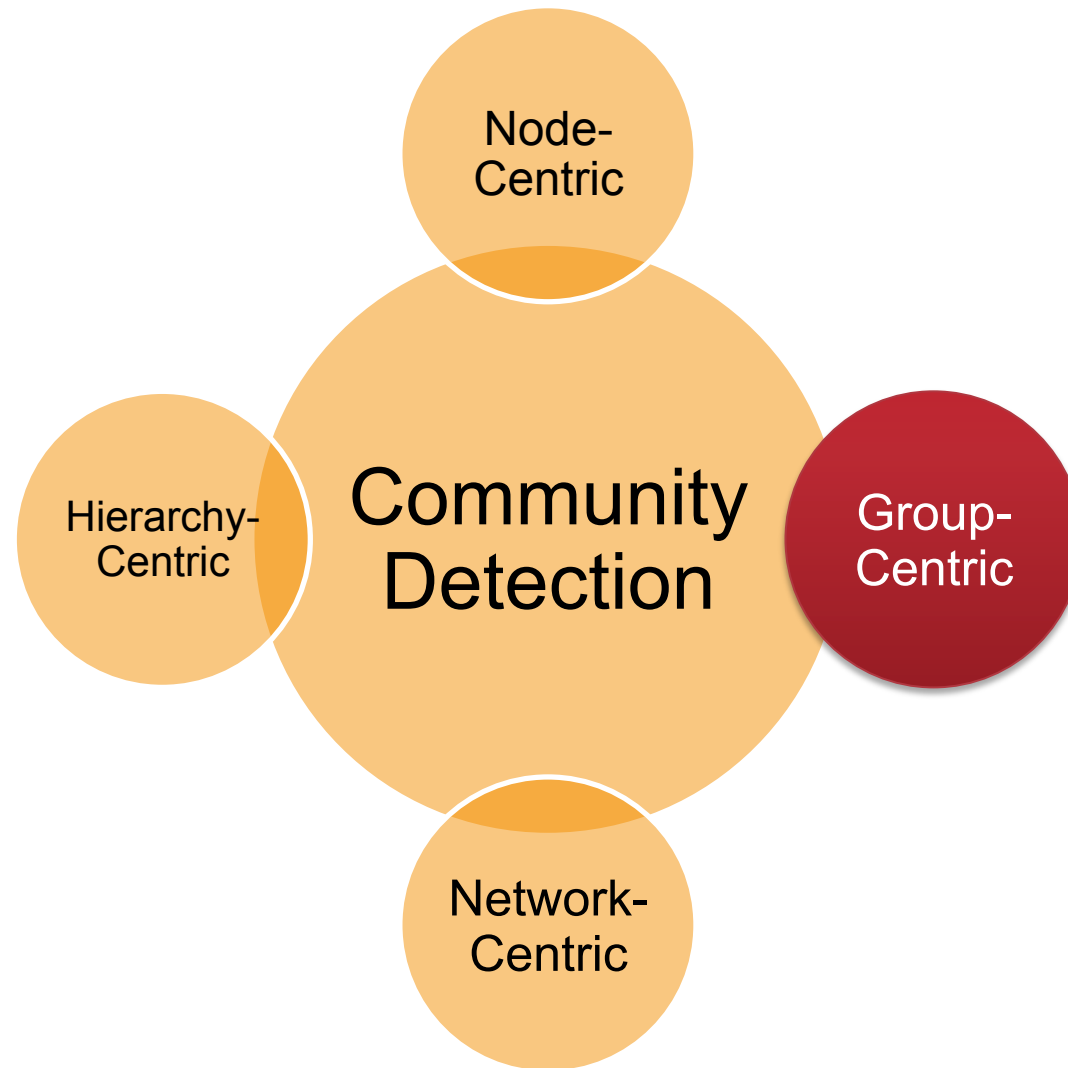
Within-Outside Ties: LS sets

- **LS sets**: Any of its proper subsets has more ties to other nodes in the group than outside the group
 - Too strict, not reasonable for network analysis
 - A relaxed definition is **Lambda sets**
 - Require the computation of edge-connectivity between any pair of nodes via minimum-cut, maximum-flow algorithm
-

Recap of Node-Centric Communities

- Each node has to satisfy certain properties
 - Complete mutuality
 - Reachability
 - Nodal degrees
 - Within-Outside Ties
 - Limitations:
 - Too strict, but can be used as the core of a community
 - Not scalable, commonly used in network analysis with small-size network
 - Sometimes not consistent with property of large-scale networks
 - e.g., nodal degrees for scale-free networks
-

Group-Centric Community Detection



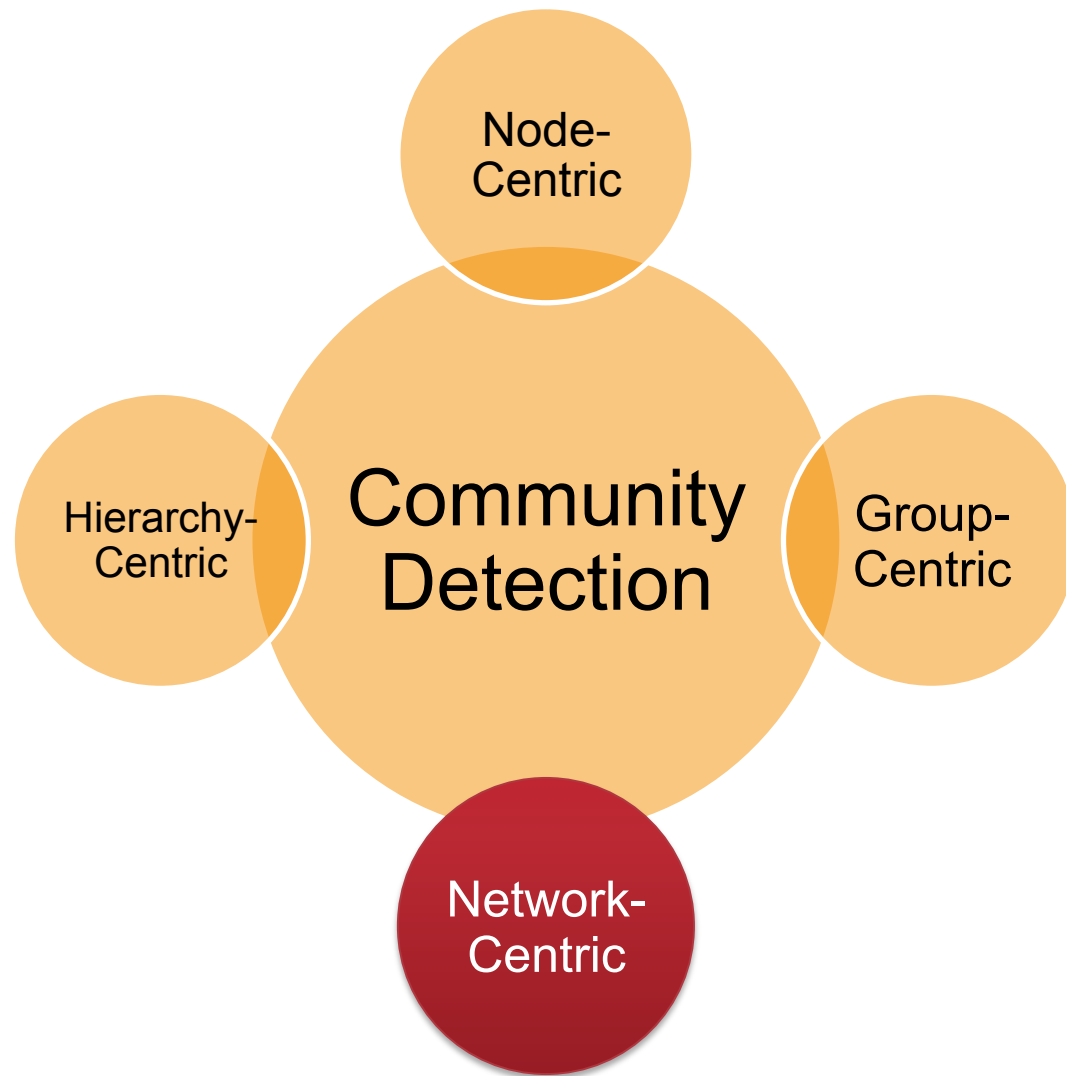
Group-Centric Community Detection

- Consider the connections within a group as whole,
- OK for some nodes to have low connectivity
- A subgraph with V_s nodes and E_s edges is a γ -dense **quasi-clique** if

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

- Recursive pruning:
 - Sample a subgraph, find a maximal γ -dense quasi-clique (the resultant size = k)
 - Remove the nodes that
 - whose degree $< k \gamma$
 - all their neighbors with degree $< k \gamma$

Network-Centric Community Detection

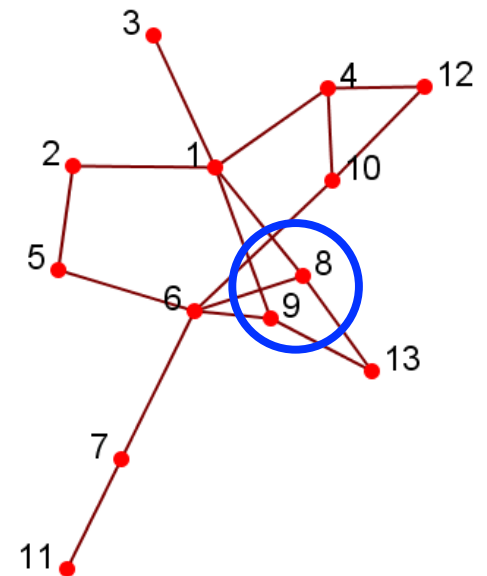


Network-Centric Community Detection

- To form a group, we need to consider the connections of the nodes globally.
 - Goal: **partition the network into disjoint sets**
 - Groups based on **Node Similarity**
 - Groups based on **Latent Space Model**
 - Groups based on **Block Model Approximation**
 - Groups based on **Cut Minimization**
 - Groups based on **Modularity Maximization**
-

Node Similarity

- Node similarity is defined by how similar their interaction patterns are
- Two nodes are **structurally equivalent** if they connect to the same set of actors
 - e.g., nodes 8 and 9 are structurally equivalent
- Groups are defined over equivalent nodes
 - Too strict
 - Rarely occur in a large-scale
 - Relaxed equivalence class is difficult to compute
- In practice, use **vector similarity**
 - e.g., cosine similarity, Jaccard similarity



Vector Similarity

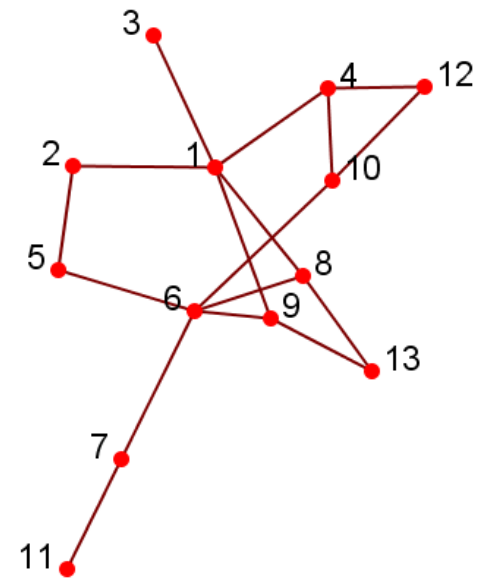
		1	2	3	4	5	6	7	8	9	10	11	12	13
a vector	5		1				1							
structurally equivalent	8	1					1							1
	9	1					1							1

Cosine Similarity: $\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$

$$\text{sim}(5,8) = \frac{1}{\sqrt{2} \times \sqrt{3}} = \frac{1}{\sqrt{6}}$$

Jaccard Similarity: $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$

$$J(5,8) = \frac{|\{6\}|}{|\{1,2,6,13\}|} = 1/4$$



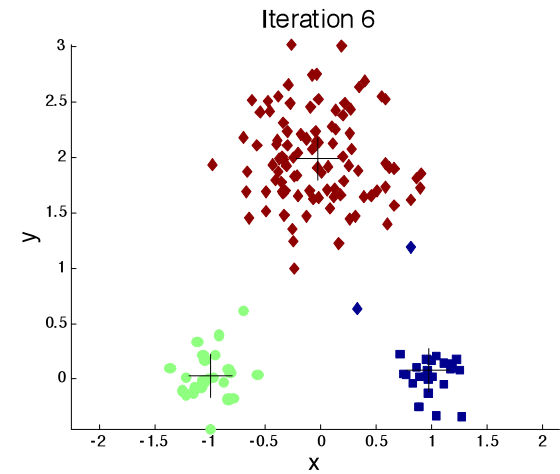
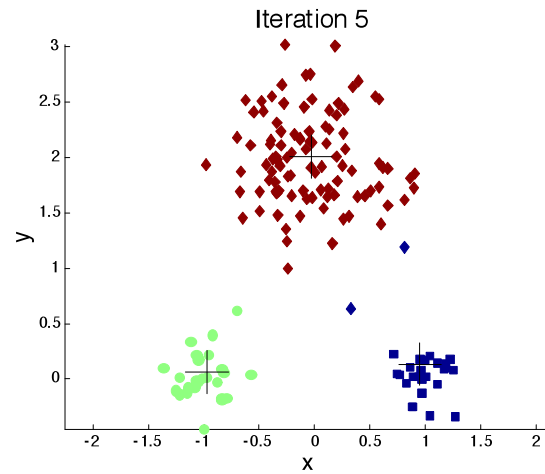
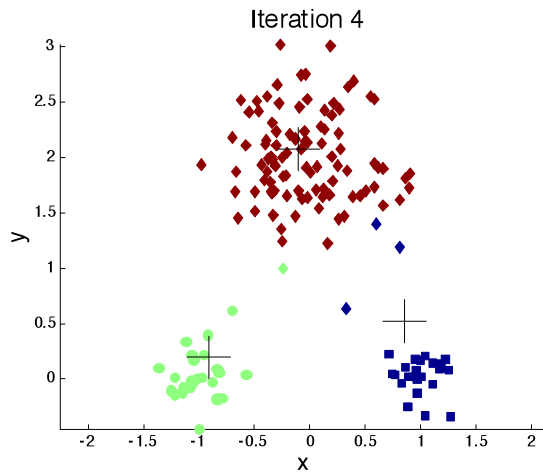
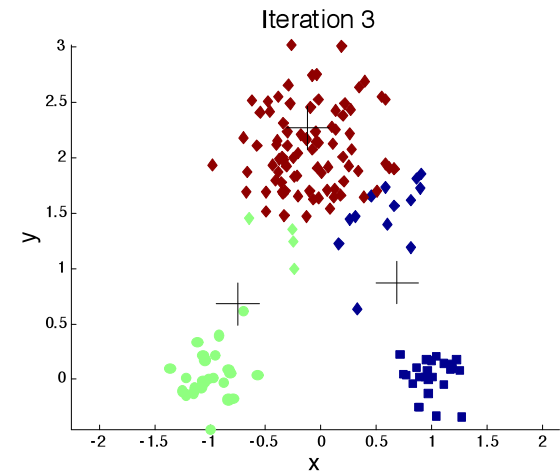
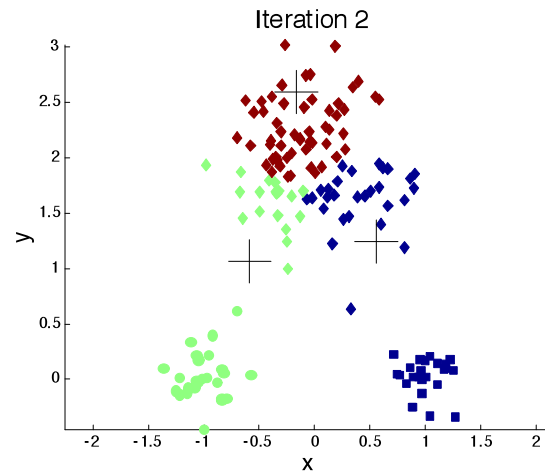
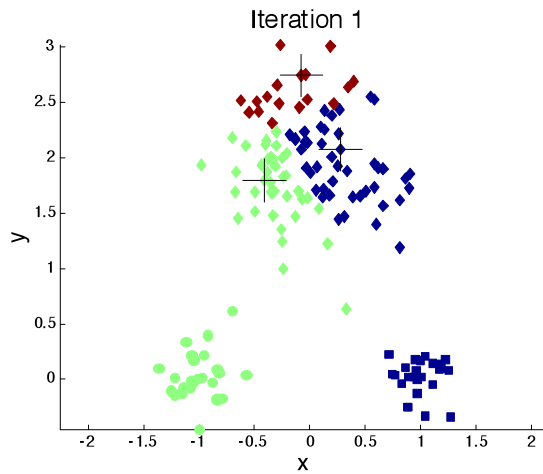
Clustering based on Node Similarity

- For practical use with huge networks:
 - ❑ Consider the connections as features
 - ❑ Use Cosine or Jaccard similarity to compute vertex similarity
 - ❑ Apply classical k-means clustering Algorithm
- K-means Clustering Algorithm
 - ❑ Each cluster is associated with a centroid (center point)
 - ❑ Each node is assigned to the cluster with the closest centroid

Algorithm 1 Basic K-means Algorithm.

- 1: Select K points as the initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning all points to the closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** The centroids don't change
-

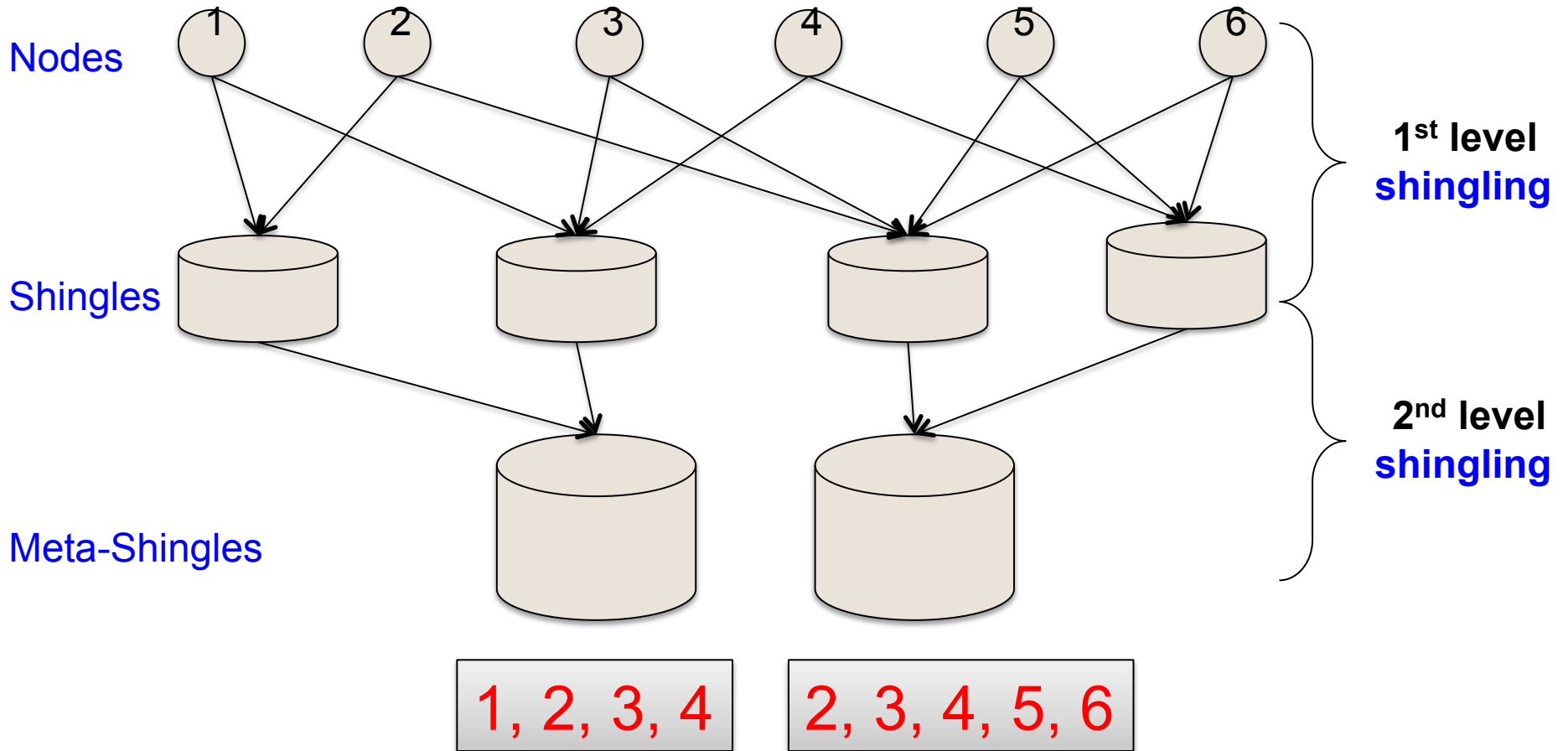
Illustration of k-means clustering



Shingling

- Pair-wise computation of similarity can be time consuming with millions of nodes
 - **Shingling** can be exploited
 - Mapping each vector into multiple shingles so the Jaccard similarity between two vectors can be computed by comparing the shingles
 - Implemented using a quick hash function
 - Similar vectors share more shingles after transformation
 - Nodes of the same shingle can be considered belonging to one community
 - In reality, we can apply 2-level shingling
-

Fast Two-Level Shingling



Groups on Latent-Space Models

- Latent-space models: Transform the nodes in a network into a lower-dimensional space such that the distance or similarity between nodes are kept in the Euclidean space

- **Multidimensional Scaling (MDS)**

- Given a network, construct a proximity matrix to denote the distance between nodes (e.g. geodesic distance)
- Let D denotes the *square distance* between nodes
- $S \in R^{n \times k}$ denotes the coordinates in the lower-dimensional space

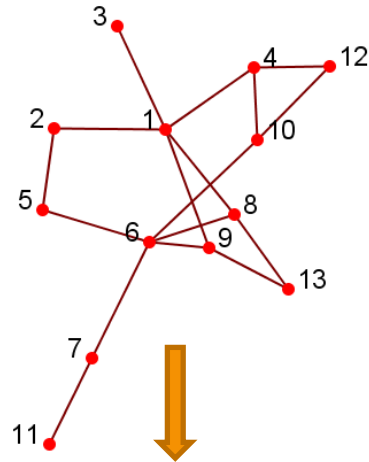
$$SS^T = -\frac{1}{2} \left(I - \frac{1}{n} ee^T \right) D \left(I - \frac{1}{n} ee^T \right) = \Delta(D)$$

- **Objective:** minimize the difference $\min \| \Delta(D) - SS^T \|_F$
- Let $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_k)$ (the top-k eigenvalues of Δ), V the top-k eigenvectors

- **Solution:**
$$S = V \Lambda^{1/2}$$

- Apply k-means to S to obtain clusters

MDS-example



Geodesic Distance Matrix

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0	1	1	1	2	2	3	1	1	2	4	2	2
2	1	0	2	2	1	2	3	2	2	3	4	3	3
3	1	2	0	2	3	3	4	2	2	3	5	3	3
4	1	2	2	0	3	2	3	2	2	1	4	1	3
5	2	1	3	3	0	1	2	2	2	2	3	3	3
6	2	2	3	2	1	0	1	1	1	1	2	2	2
7	3	3	4	3	2	1	0	2	2	2	1	3	3
8	1	2	2	2	2	1	2	0	2	2	3	3	1
9	1	2	2	2	2	1	2	2	0	2	3	3	1
10	2	3	3	1	2	1	2	2	2	0	3	1	3
11	4	4	5	4	3	2	1	3	3	3	0	4	4
12	2	3	3	1	3	2	3	3	3	1	4	0	4
13	2	3	3	3	3	2	3	1	1	3	4	4	0

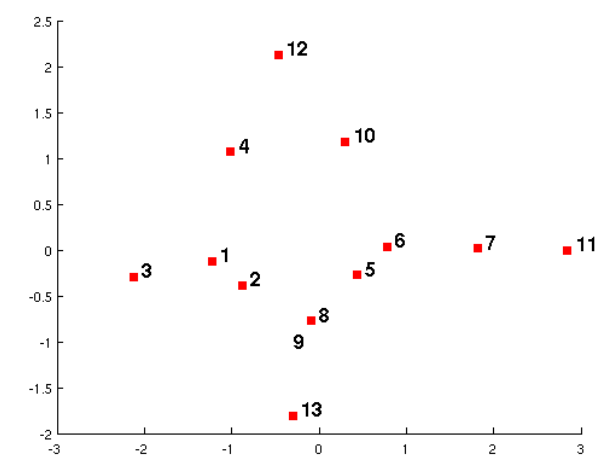
MDS →

1, 2, 3, 4, 10, 12 5, 6, 7, 8, 9, 11, 13

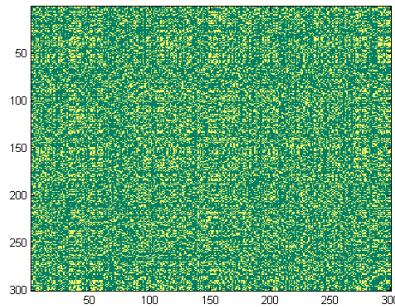
↑ k-means

S

-1.22	-0.12
-0.88	-0.39
-2.12	-0.29
-1.01	1.07
0.43	-0.28
0.78	0.04
1.81	0.02
-0.09	-0.77
-0.09	-0.77
0.30	1.18
2.85	0.00
-0.47	2.13
-0.29	-1.81

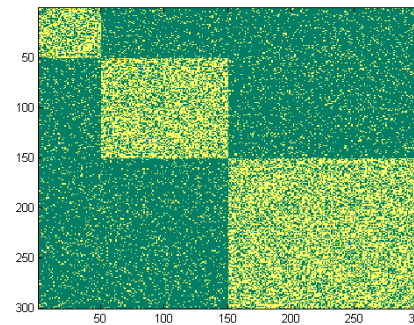


Block-Model Approximation



Network Interaction Matrix

After
Reordering
→



Block Structure

➤ **Objective:** Minimize the difference between an interaction matrix and a block structure

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

s.t. $S \in \{0, 1\}^{n \times k}, \Sigma \in R^{k \times k}$ is diagonal

S is a
community
indicator matrix

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

Cut-Minimization

- Between-group interactions should be infrequent
- **Cut**: number of edges between two sets of nodes
- **Objective**: minimize the cut

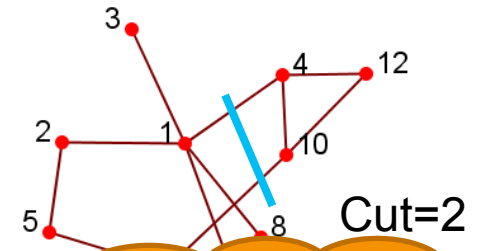
$$\text{cut}(C_1, C_2, \dots, C_k) = \sum_{i=1}^k \text{cut}(C_i, \overline{C_i})$$

- Limitations: often find communities of only one node
- Need to consider the group size

- Two commonly-used variants:

$$\text{Ratio-cut}(C_1, C_2, \dots, C_k) = \sum_{i=1}^k \frac{\text{cut}(C_i, \overline{C_i})}{|V_i|}$$

$$\text{Normalized-cut}(C_1, C_2, \dots, C_k) = \sum_{i=1}^k \frac{\text{cut}(C_i, \overline{C_i})}{\text{vol}(V_i)}$$



Number of nodes
in a community

Number of
within-group
interactions

Graph Laplacian

- Can be relaxed into the following min-trace problem

$$\min_{S \in \mathbb{R}^{n \times k}} \text{Tr}(S^T L S) \quad \text{s.t. } S^T S = I$$

- L is the (normalized) **Graph Laplacian**

$$\begin{aligned} L &= D - A \\ \text{normalized-}L &= I - D^{-1/2} A D^{-1/2} \end{aligned} \quad D = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_n \end{pmatrix}$$

- **Solution:** S are the eigenvectors of L with smallest eigenvalues (except the first one)
- Post-Processing: apply k-means to S
- a.k.a. **Spectral Clustering**

Modularity Maximization

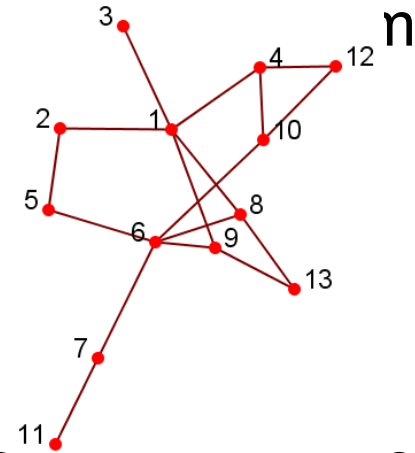
- **Modularity** measures the group interactions compared with the **expected random connections** in the group
- In a network with m edges, for two nodes with degree d_i and d_j , the expected random connections are
- The interaction utility in a group:

$$d_i d_j / 2m$$

$$\sum_{i \in C, j \in C} A_{ij} - d_i d_j / 2m$$

- To partition the group into multiple groups we maximize

$$\max \frac{1}{2m} \sum_C \sum_{i \in C, j \in C} A_{ij} - d_i d_j / 2m$$



Expected Number of edges between 6 and 9 is $5 \cdot 3 / (2 \cdot 17) = 15/34$

Modularity Matrix

- The modularity maximization can also be formulated in 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
-

Properties of Modularity

- Properties of modularity:
 - Between $(-1, 1)$
 - Modularity = 0 If all nodes are clustered into one group
 - Can automatically determine optimal number of clusters
 - Resolution limit of modularity
 - Modularity maximization might return a community consists multiple small modules
-

Matrix Factorization Form

- For latent space models, block models, spectral clustering and modularity maximization
- All can be formulated as

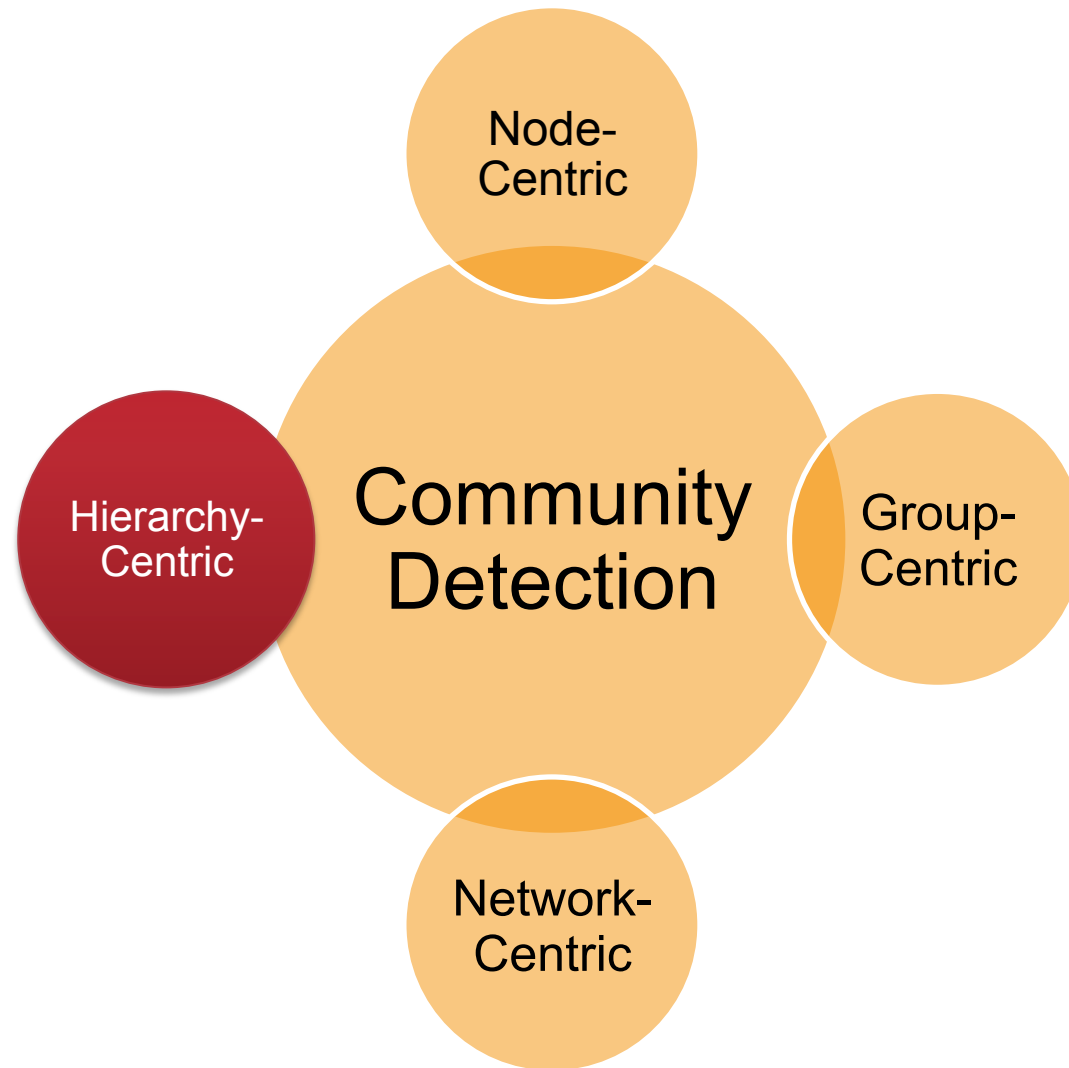
$$\begin{aligned} \max(\min)_S \quad & Tr(S^T X S) \\ \text{s.t.} \quad & S^T S = I \end{aligned}$$

$$X = \begin{cases} \Delta(D) & \text{(Latent Space Models)} \\ \text{Sociomatrix} & \text{(Block Model Approximation)} \\ \text{Graph Laplacian} & \text{(Cut Minimization)} \\ \text{Modularity Matrix} & \text{(Modularity maximization)} \end{cases}$$

Recap of Network-Centric Community

- Network-Centric Community Detection
 - Groups based on Node Similarity
 - Groups based on Latent Space Models
 - Groups based on Cut Minimization
 - Groups based on Block-Model Approximation
 - Groups based on Modularity maximization
 - **Goal:** Partition network nodes into several disjoint sets
 - **Limitation:** Require the user to specify the number of communities beforehand
-

Hierarchy-Centric Community Detection

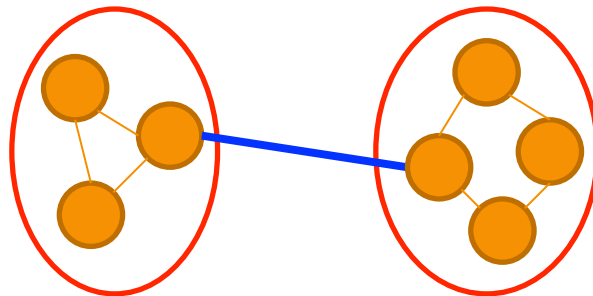


Hierarchy-Centric Community Detection

- **Goal:** Build a hierarchical structure of communities based on network topology
 - Facilitate the analysis at different resolutions
 - Representative Approaches:
 - Divisive Hierarchical Clustering
 - Agglomerative Hierarchical Clustering
-

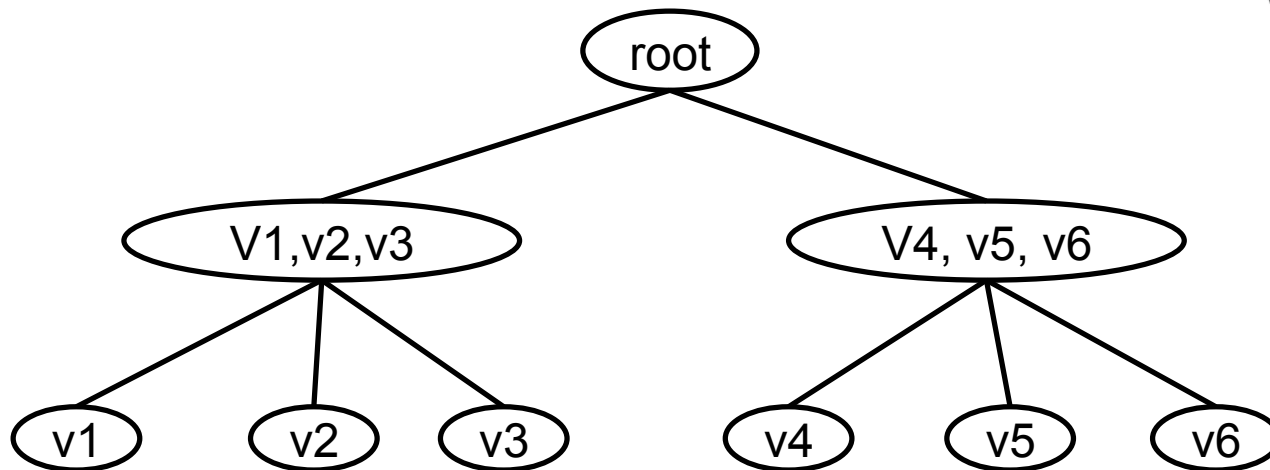
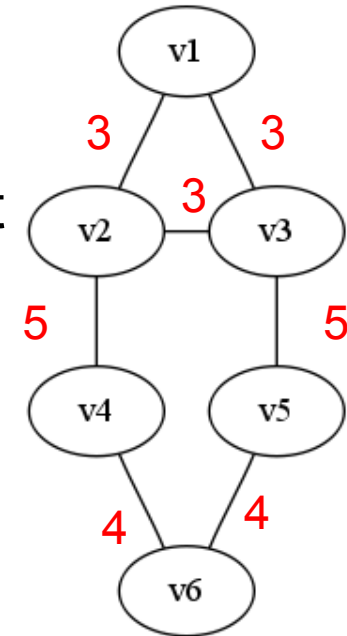
Divisive Hierarchical Clustering

- Divisive Hierarchical Clustering
 - Partition the nodes into several sets
 - Each set is further partitioned into smaller sets
- Network-centric methods can be applied for partition
- One particular example is based on edge-betweenness
- **Edge-Betweenness:** Number of shortest paths between any pair of nodes that pass through the edge
- Between-group edges tend to have larger edge-betweenness



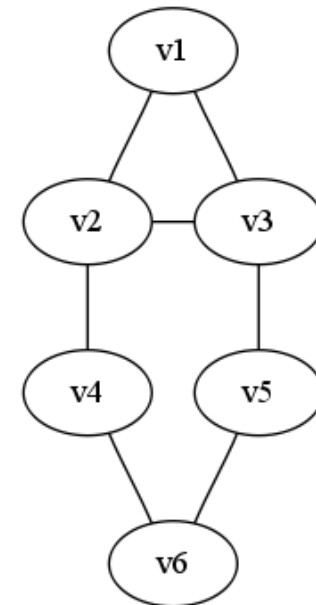
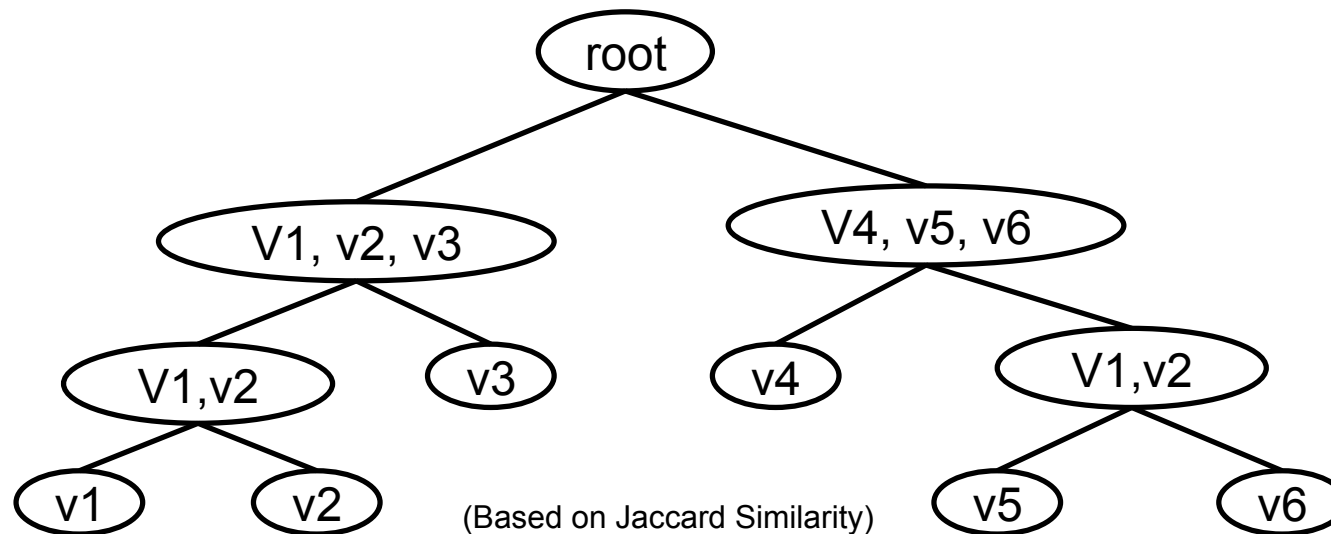
Divisive clustering on Edge-Betweenness

- Progressively remove edges with the highest betweenness
 - ❑ Remove $e(2,4)$, $e(3,5)$
 - ❑ Remove $e(4,6)$, $e(5,6)$
 - ❑ Remove $e(1,2)$, $e(2,3)$, $e(3,1)$



Agglomerative Hierarchical Clustering

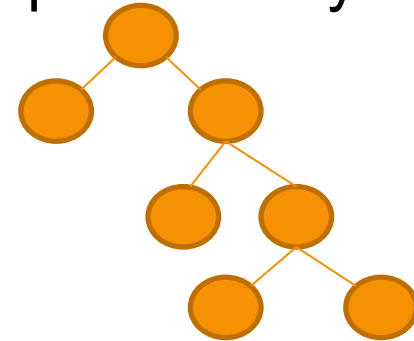
- Initialize each node as a community
- Choose two communities satisfying certain **criteria** and merge them into larger ones
 - ❑ Maximum Modularity Increase
 - ❑ Maximum Node Similarity



Recap of Hierarchical Clustering

- Most hierarchical clustering algorithm output a binary tree

- Each node has two children nodes
- Might be highly imbalanced



- Agglomerative clustering can be very sensitive to the nodes processing order and merging criteria adopted.
 - Divisive clustering is more stable, but generally more computationally expensive
-

Summary of Community Detection

- The Optimal Method?



- It varies depending on applications, networks, computational resources etc.

- Scalability can be a concern for networks in social media

- Other lines of research

- ❑ Communities in directed networks
 - ❑ Overlapping communities
 - ❑ Community evolution
 - ❑ Group profiling and interpretation
-

IMPLEMENTATIONS IN MAP- REDUCE

Scale of Networks

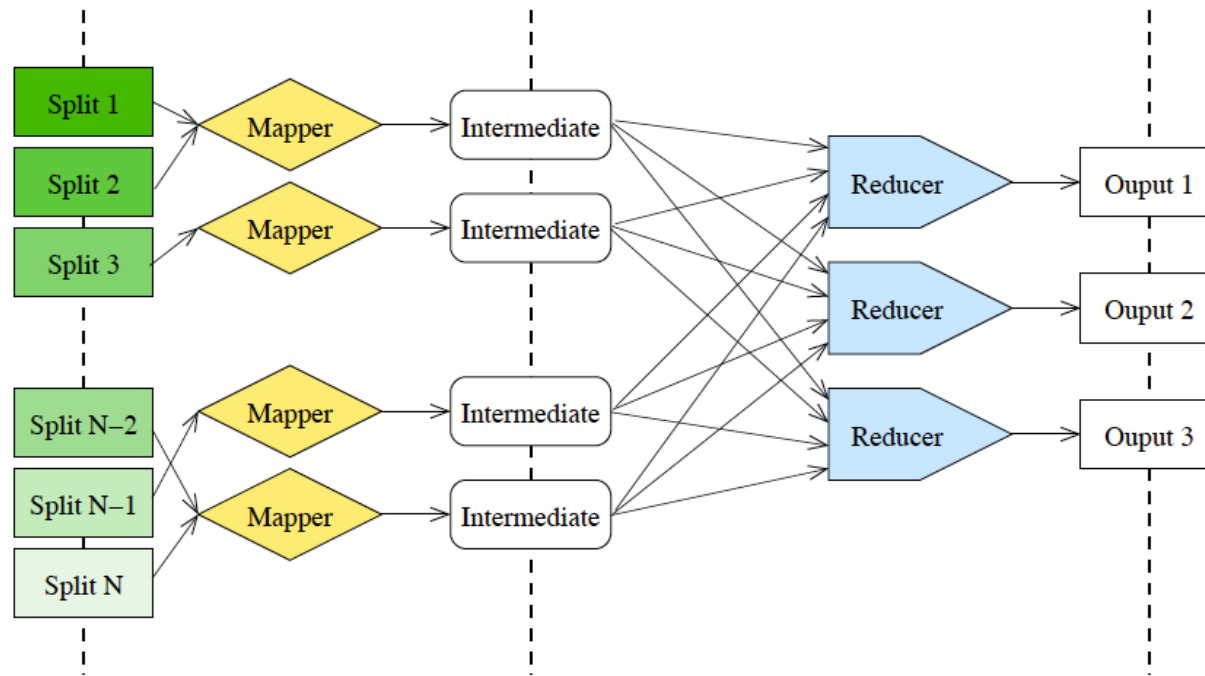
- 1970s: 10^1 nodes (now considered as toy example)
- 1990s: 10^4 nodes (say, coauthorship network)
- Nowadays: $>10^8$ nodes
 - ❑ Mail, Messenger, Facebook, Twitter, LinkedIn
 - ❑ May contain other meta information about nodes and edges
 - ❑ Exceed memory limits of a “luxury” workstation
 - ❑ Require **considerable storage**
- e.g., Yahoo IM graph
 - ❑ hundreds of millions of nodes
 - ❑ billions of connections
 - ❑ occupies more than 300 GB



Networks are
scale-free; But
algorithms are
NOT

MapReduce

- Inspired from the primitives of Lisp for list processing
- Fundamental idea: **move computation to data**
- Mapper: $\langle \text{key}_{in}, \text{value}_{in} \rangle \rightarrow \langle \text{key}_{intermediate}, \text{value}_{intermediate} \rangle$
- Reducer: $\langle \text{key}_{intermediate}, \{\text{value}_{intermediate}\} \rangle \rightarrow \langle \text{key}_{out}, \text{value}_{out} \rangle$



MapReduce Example

- Essentially a **distributed grep-sort-aggregate**
- Word-Count example
- Unix Pipe: **cat input | emitword | sort | uniq -c**
- MapReduce: Mapper, Reducer

```
sub emitword{  
  while ( my $line = <STDIN>){  
    chomp $line;  
    my @words = split ' ', $line;  
    foreach my $word (@words){  
      # emit (word, 1)  
      print $word, "\t", 1, "\n";  
    }  
  }  
}
```

uniq -c

Taken care by
MapReduce Framework

Hadoop

- An open source implementation to MapReduce
 - Very easy to install and use (you can install Hadoop in your local box in few minutes)
 - Hadoop is Not ...
 - ❑ Not for high availability (failures happen all the time)
 - ❑ Not designed for low latency
 - ❑ Not geographically distributed
 - Hadoop cluster does not span over multiple colos
 - Good for
 - ❑ Fault tolerance in scale; transparent to users
 - ❑ High throughput for processing data
-

Existing Solutions other than Hadoop

- **Approximation:**
 - ❑ Subsample a network
 - ❑ identify communities in the small network
 - ❑ Recover the community structure of the whole graph (Nystrom's method)
 - **METIS: Multi-Level Method for Graph Partition**
 - ❑ Coarse a network level by level into a small graph
 - ❑ Partition the small graph
 - ❑ Recover the partition of the original graph by uncoarsing gradually
 - **MPI-based solutions**
 - ❑ **ParMETIS**: Distributed version of METIS
 - ❑ **PARPACK**: Parallel ARPACK
-

Software based on Hadoop

- **XRIME:** <http://xrime.sourceforge.net/>
 - ❑ Hadoop-based large scale social network analysis
 - ❑ Support some commonly-used SNA metrics
 - connected components, bi-connected components
 - communities: k-core, maximal cliques
 - PageRank, HITS, clustering coefficient
 - ❑ Not (well) documented
 - **Mahout:** <http://mahout.apache.org/>
 - ❑ Scalable Machine Learning and Data Mining Library
 - ❑ Include some clustering implementations
 - k-means clustering, Dirichlet process clustering, LDA
 - spectral clustering (only binary case), SVD
 - ❑ Not very mature and stable yet
-

k-means for Undirected Networks

- For practical use with huge networks:
 - ❑ Consider the connections as features
 - ❑ Use Cosine or Jaccard similarity to compute vertex similarity
 - ❑ Apply classical k-means clustering Algorithm
- K-means Clustering Algorithm
 - ❑ Each cluster is associated with a centroid (center point)
 - ❑ Each node is assigned to the cluster with the closest centroid

Algorithm 1 Basic K-means Algorithm.

- 1: Select K points as the initial centroids.
 - 2: **repeat**
 - 3: Form K clusters by assigning all points to the closest centroid.
 - 4: Recompute the centroid of each cluster.
 - 5: **until** The centroids don't change
-

k-means in MapReduce

- Initialization:

- represent network data in proper format: [adjacency list](#)
- Normalization: assign proper weights to each edge
- Random select some vertices as cluster centroids

- Iterate until convergence

- Mapper:

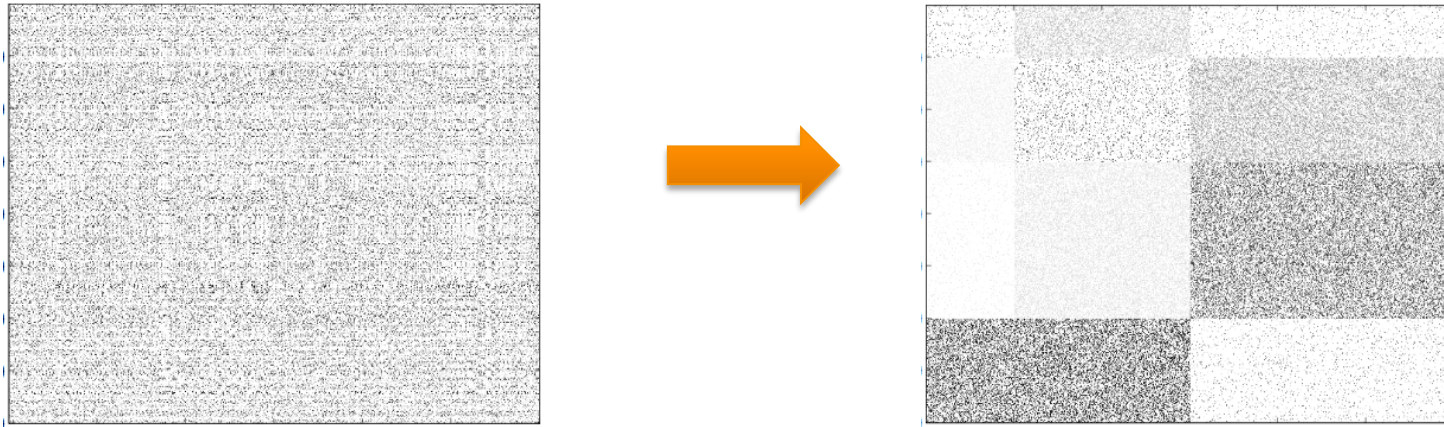
- Broadcast the centroid info to all cluster nodes
- For each vertex
 - compute its similarity to each centroid
 - Assign the vertex to the cluster of the closest centroid
- Emit (cluster_ID, vertex)

- Reducer:

- For each cluster_ID
 - aggregate its member vertices info to compute the new centroid
-

Clustering in Directed Networks

- Many networks are directed
 - mail, messenger, twitter following-follower
- Assuming separate communities for rows & columns



$$A \approx \mathbf{R}_k \mathbf{G}_{k \times l} \mathbf{C}_l$$

R: the community assignment in rows

C: the community assignment in columns

G: the interaction density between R and C communities

Algorithm

Procedure 1 CC (\mathbf{A} , k , l)

- 1: Initialize \mathbf{r} and \mathbf{c} .
 - 2: Compute the group statistics matrix \mathbf{G} .
 - 3: **repeat**
 - 4: **for each** row $i = 1..m$ **do**
 - 5: **for each** row group label $p = 1..k$ **do**
 - 6: Assign $r(i) \leftarrow p$ if this minimizes error
 - 7: Update \mathbf{G} , \mathbf{r}
 - 8: Do the same for columns
 - 9: **until** cost does not decrease
 - 10: **return** \mathbf{r} and \mathbf{c}
-

Implementation in Hadoop

Mapper:

Broadcast G and c
Assign community for each row;
Emit (cluster_ID, row_statistics)

Reducer:

Update Group statistics

Procedure 2 CCRWMAPPER (k, v)

Globals: Cluster statistics G , labels c

Source node is $i \equiv k$

Adjacency list of i is $a_i \equiv V$

Compute row statistics $g_i := \text{ROWSTATISTICS}(a_i, c)$

for each group label $p = 1..k$ **do**

if assigning i to p would lower cost **then**

$r(i) \leftarrow p$

emit $\langle r(i), (g_i, \{i\}) \rangle$

Procedure 3 CCRWREDUCER (k, V)

Row group label is $p \equiv k$

Initialize $g_p \leftarrow 0, I_p \leftarrow \emptyset$

for each map value $(g, I) \in V$ **do**

$g_p \leftarrow \text{COMBINESTATISTICS}(g_p, g)$

$I_p \leftarrow I_p \cup I$

emit $\langle p, (g_p, I_p) \rangle$

Update the group interaction matrix G

Procedure 4 COLLECTRESULTS

Post-processing:

Update G and r

Initialize $G \leftarrow 0, r \leftarrow 0$

for reduce output $\langle p, (\mathbf{g}_p, I_p) \rangle$ **do**

$g_p: \leftarrow \mathbf{g}_p$

$r(i) \leftarrow p$, for all $i \in I_p$

return G and r

Update the column community is essentially a similar process.
Involve **multiple iterations of MapReduce**

Limitations

- Some information are **broadcasted to all cluster nodes**
 - ❑ K-means for undirected network: centroid info
 - ❑ Clustering for directed network: the group assignment, group interaction matrix
 - If **the number of communities is huge, or soft clustering**
 - ❑ the info cannot be loaded into the memory of one cluster node
 - ❑ the broadcasting process may take a while
 - ❑ Implementations in that case becomes quite messy 😞
 - ❑ Need multiple MapReduce tasks to achieve one single iteration.
 - Look ahead
 - ❑ Soft clustering on graphs with Hadoop
 - ❑ Community structure in large networks follow some pattern. Should we adopt a different procedure?
-

Social Computing Application

PREDICTION VIA SOCIAL CONNECTIONS

Network-based Prediction

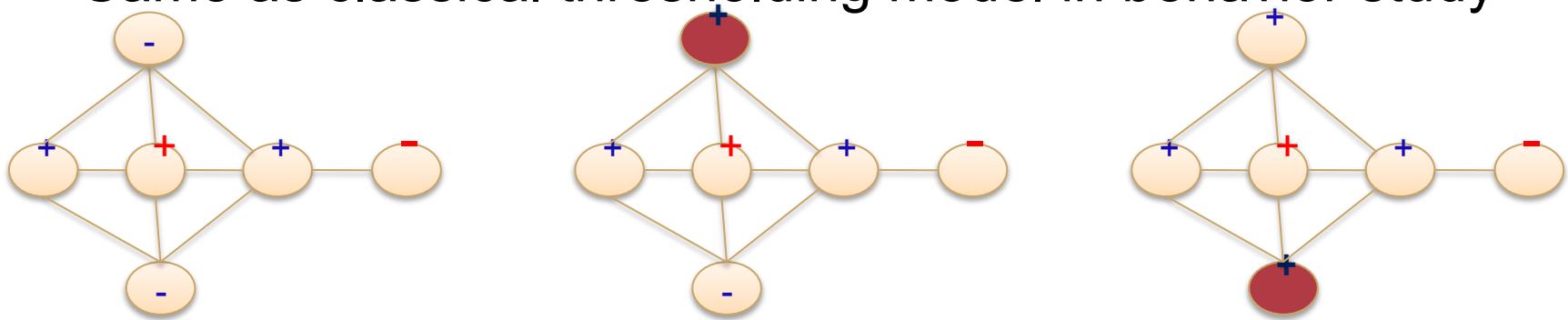
- User Preference or Behavior can be represented by labels (+/-)
 - Whether or not clicking on an ad
 - Whether or not interested in certain topics
 - Subscribed to certain political views
 - Like/Dislike a product

 - **Given:**
 - A social network (i.e., connectivity information)
 - Some actors with identified labels

 - **Output:**
 - Labels of other actors within the same network
-

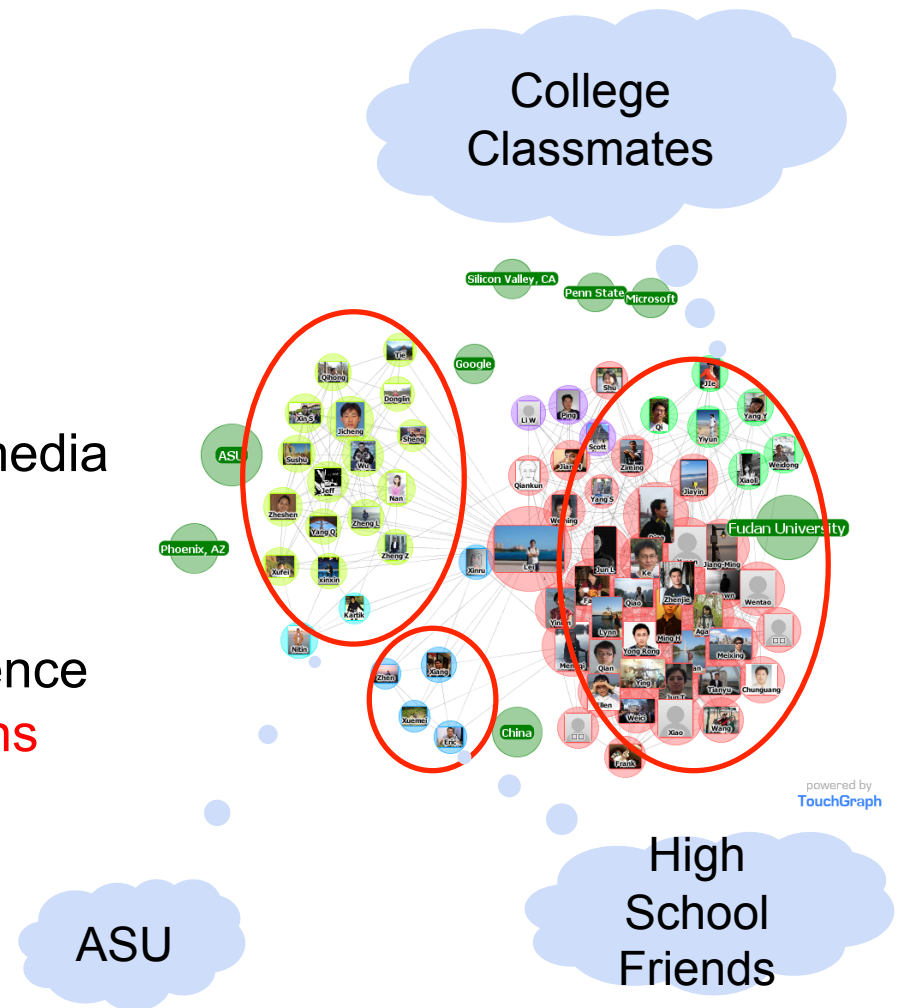
Approach I: Collective Inference

- Markov Assumption
 - The label of one node depends on that of its neighbors
- Training
 - Build a relational model based on labels of neighbors
- Prediction --- **Collective inference**
 - Predict the label of one node while fixing labels of its neighbors
 - Iterate until convergence
- Same as classical thresholding model in behavior study

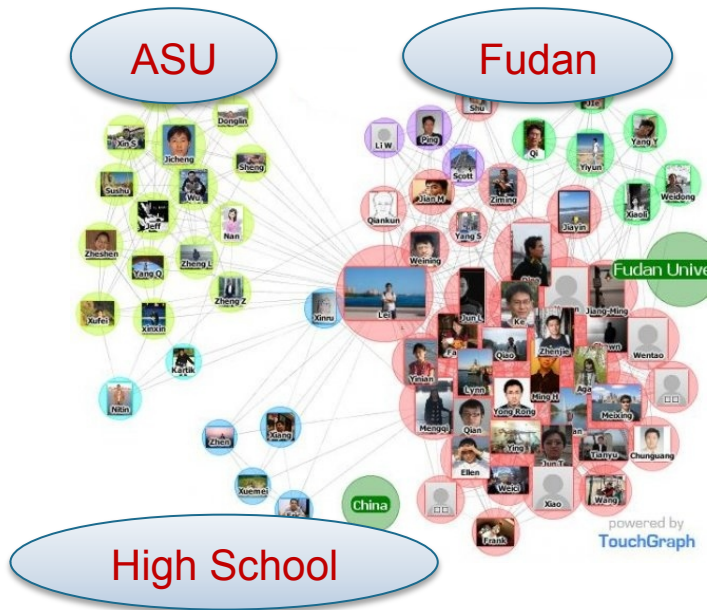


Heterogeneous Relations

- Connections in a social network are heterogeneous
- Relation type information in social media is not always available
- Direct application of collective inference to social media **treats all connections equivalently**



Social Dimensions

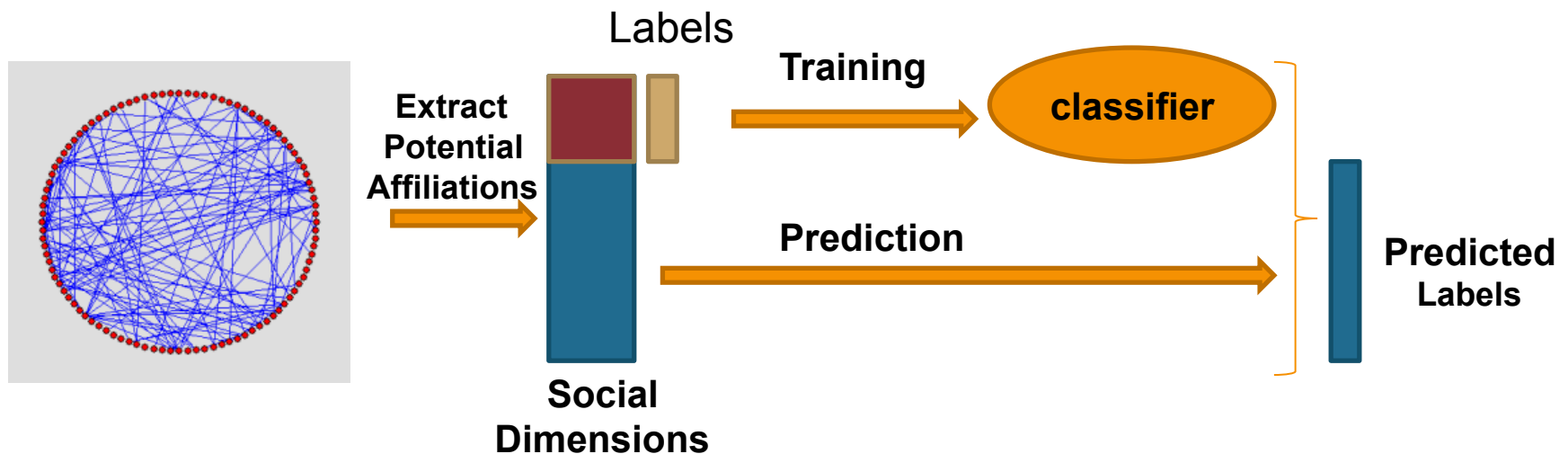


	ASU	Fudan	High School
Lei	1	1	1
Actor ₁	1	0	0
Actor ₂	0	1	1
.....

One actor can be involved in multiple affiliations

- **Challenge:** Relation (affiliation) information is unknown.
 - 1) **How to extract the social dimensions?**
 - Actors of the same affiliation interact with each other frequently
→ **Community Detection**
 - 2) **Which affiliations are informative for behavior prediction?**
 - Let label information help → **Supervised Learning**

Approach II: Social-Dimension Approach (SocioDim)



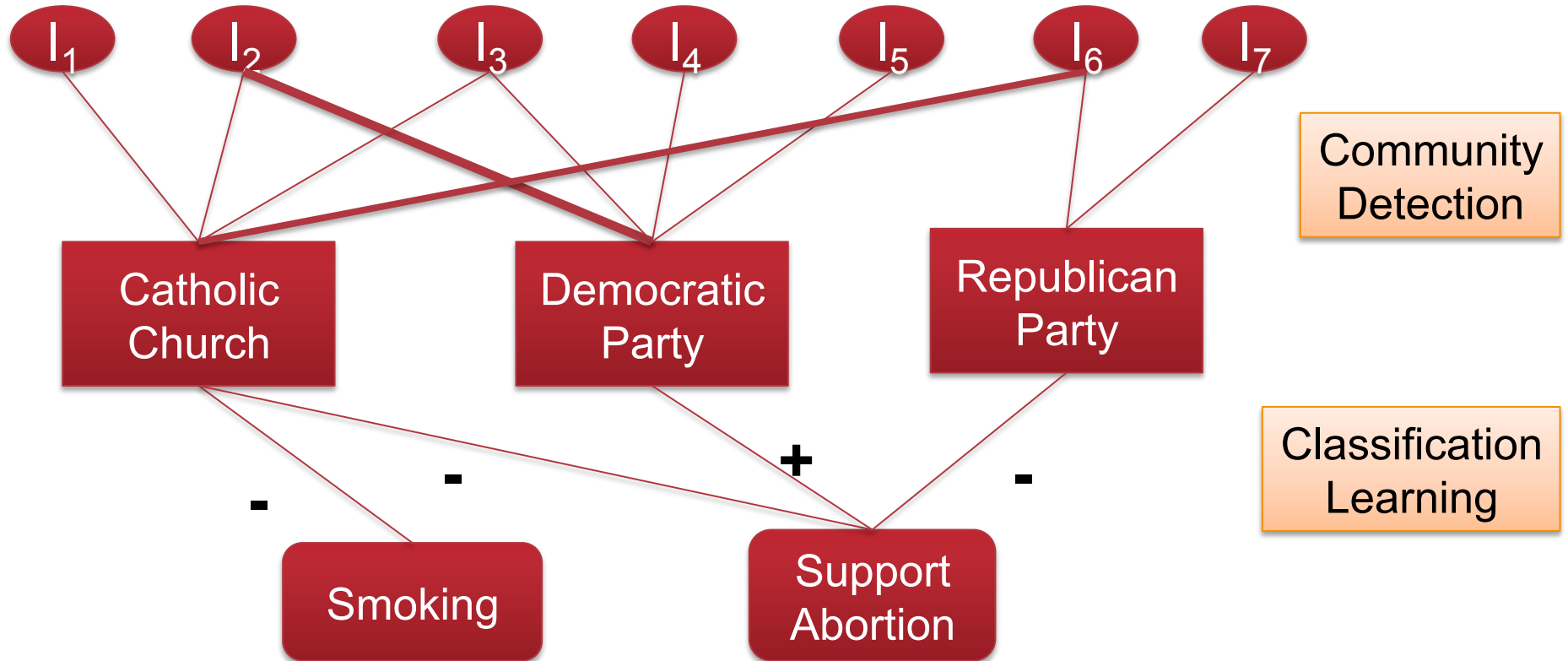
■ Training:

- Extract social dimensions to **represent potential affiliations of actors**
 - Any community detection methods is applicable (block model, spectral clustering)
- Build a classifier to **select those discriminative dimensions**
 - Any discriminative classifier is acceptable (SVM, Logistic Regression)

■ Prediction:

- Predict labels based on one actor's latent social dimensions
- No collective inference is necessary

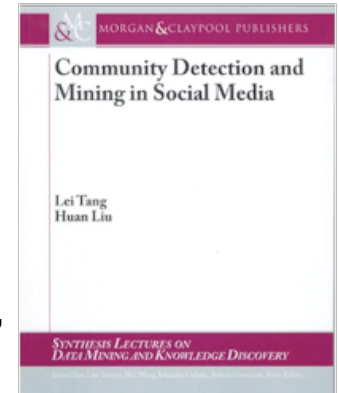
An Example of SocioDim Model



Communities are features!!

- Community detection can be used to **differentiate connections in networks**
 - One is likely to participate in multiple communities
 - Community membership of one node become **features**
 - Community-based learning outperforms collective inference, especially for social media networks
 - Enable **integration** of node features and network information
-

References



- Lei Tang and Huan Liu. *Community Detection and Mining in Social Media*, Morgan & Claypool Publishers, 2010.
- Lei Tang and Huan Liu. *Graph Mining Applications to Social Network Analysis*. In *Managing and Mining Graph Data*, Editors: Charu Aggarwal and Haixun Wang. Springer, 2010.
- Lei Tang and Huan Liu. *Leveraging Social Media Networks for Classification*. Journal of Data Mining and Knowledge Discovery (DMKD), 2011.
- Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, Edward Y. Chang, "Parallel Spectral Clustering in Distributed Systems," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 3, pp. 568-586, Mar. 2011, doi: 10.1109/TPAMI.2010.88
- Spiros Papadimitriou and Jimeng Sun. 2008. DisCo: Distributed Co-clustering with Map-Reduce: A Case Study towards Petabyte-Scale End-to-End Mining. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining (ICDM '08)*. IEEE Computer Society, Washington, DC, USA, 512-521. DOI=10.1109/ICDM.2008.142 <http://dx.doi.org/10.1109/ICDM.2008.142>

Thank You!



Please feel free to contact **Lei Tang** (L.Tang@asu.edu) if you have any questions!
