Community Detection and Evaluation

Chapter 3

Community

- Community: It is formed by individuals such that those within a group interact with each other more frequently than with those outside the group
  - a.k.a. group, cluster, cohesive subgroup, module in different contexts
- Community detection: discovering groups in a network where individuals' group memberships are not explicitly given

- Why communities in social media?
  - Human beings are social
  - 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
  - Interactions between nodes can help determine communities

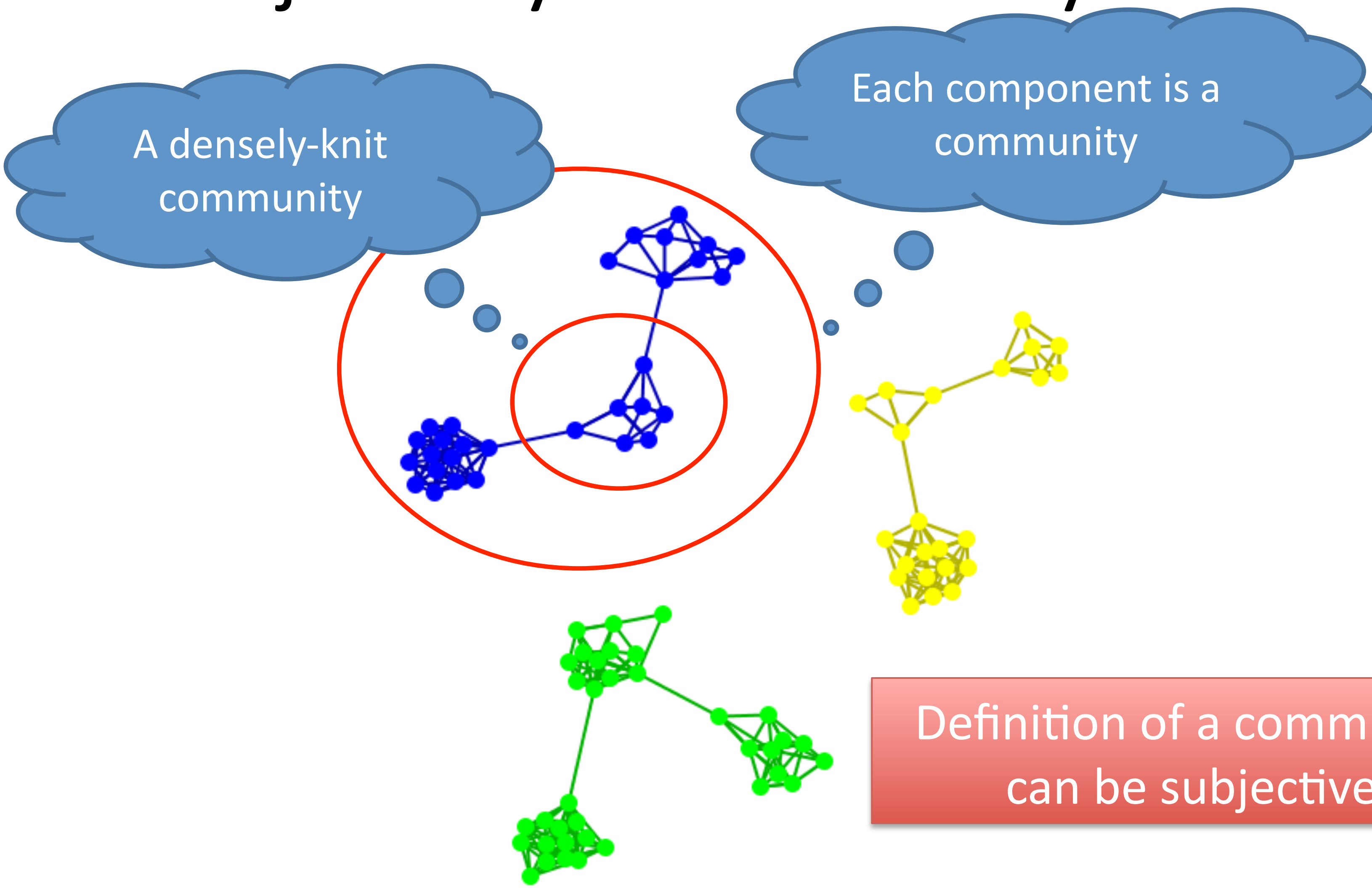
Communities in Social Media

- Two types of groups in social media
  - Explicit Groups: formed by user subscriptions
  - Implicit Groups: implicitly formed by social interactions

- Some social media sites allow people to join groups, is it necessary to extract groups based on network topology?
  - Not all sites provide community platform
  - Not all people want to make effort to join groups
  - Groups can change dynamically
- Network interaction provides rich information about the relationship between users
  - Can complement other kinds of information
  - Help network visualization and navigation
  - Provide basic information for other tasks

COMMUNITY DETECTION

Subjectivity of Community Definition



Definition of a community can be subjective.

Taxonomy of Community Criteria

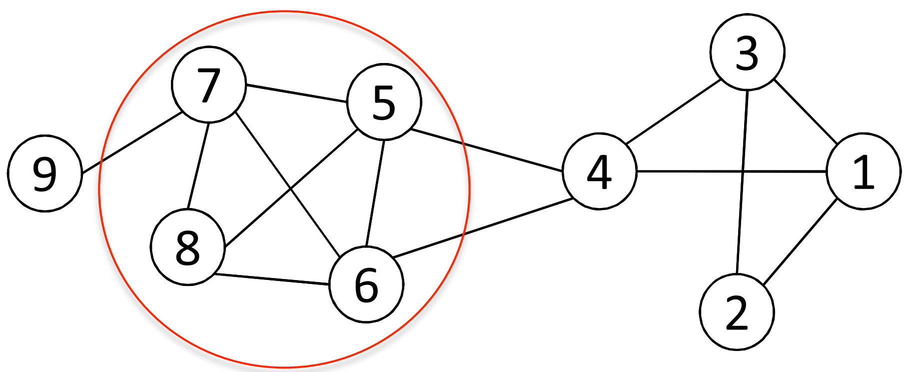
- Criteria vary depending on the tasks
- Roughly, community detection methods can be divided into 4 categories (not exclusive):
- Node-Centric Community
  - Each node in a group satisfies certain properties
- Group-Centric Community
  - Consider the connections within a group as a whole. The group has 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

- 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: Cliques

- Clique: a maximum complete subgraph in which all nodes are adjacent to each other



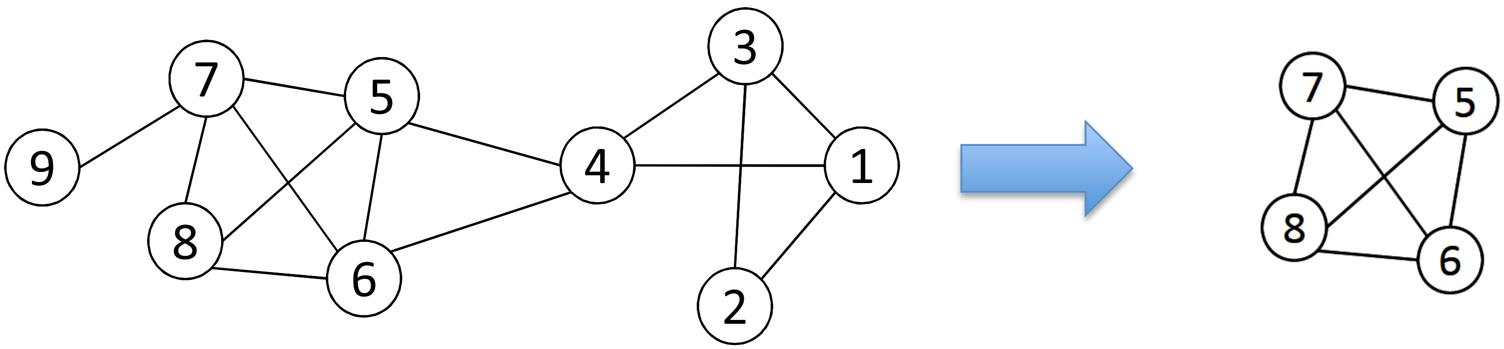
Nodes 5, 6, 7 and 8 form a clique

- NP-hard to find the maximum clique in a network
- Straightforward implementation to find cliques is very expensive in time complexity

Finding the Maximum Clique

- In a clique of size k, each node maintains degree >= k-1
- Nodes with degree < k-1 will not be included in the maximum clique
- Recursively apply the following pruning procedure
  - Sample a sub-network from the given network, and find a clique in the sub-network, say, by a greedy approach
  - Suppose the clique above is size k, in order to find out a *larger* clique, all nodes with degree <= k-1 should be removed.
- Repeat until the network is small enough
- Many nodes will be pruned as social media networks follow a power law distribution for node degrees

Maximum Clique Example



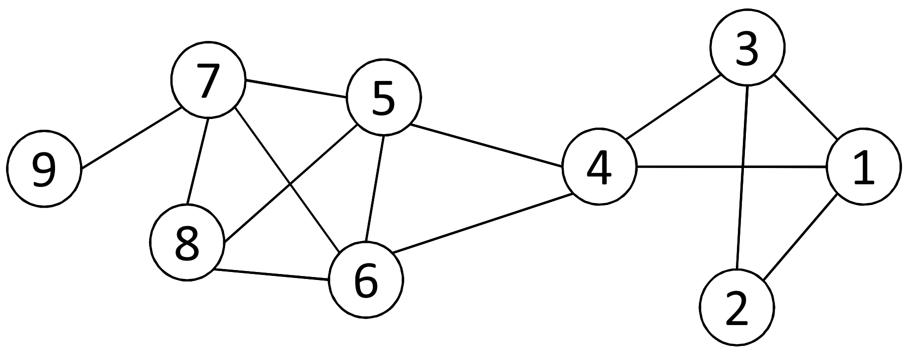
- Suppose we sample a sub-network with nodes {1-5} and find a clique {1, 2, 3} of size 3
- In order to find a clique >3, remove all nodes with degree <=3-1=2
  - Remove nodes 2 and 9
  - Remove nodes 1 and 3
  - Remove node 4

Clique Percolation Method (CPM)

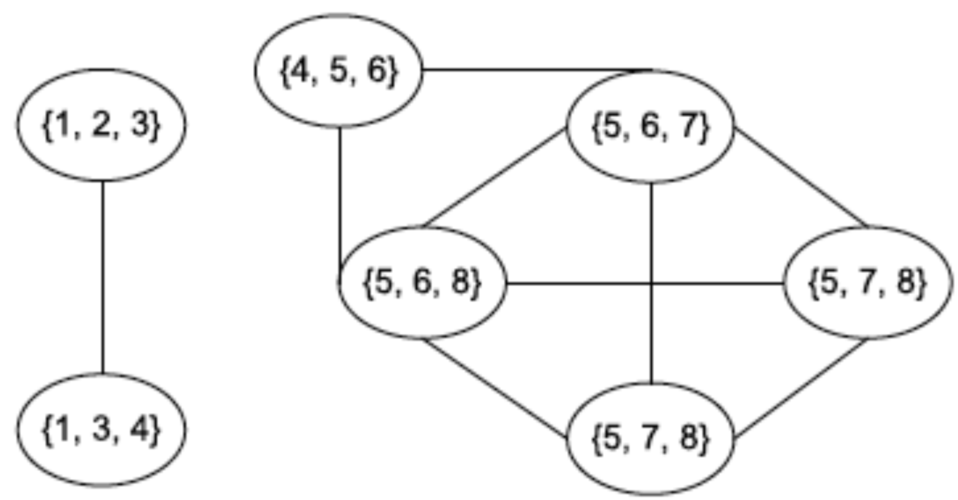
- Clique is a very strict definition, unstable
- Normally use cliques as a core or a seed to find larger communities

- CPM is such a method to find overlapping communities
  - **Input**
    - A parameter k, and a network
  - **Procedure**
    - Find out all cliques of size k in a given network
    - Construct a clique graph. Two cliques are adjacent if they share k-1 nodes
    - Each connected components in the clique graph form a community

CPM Example



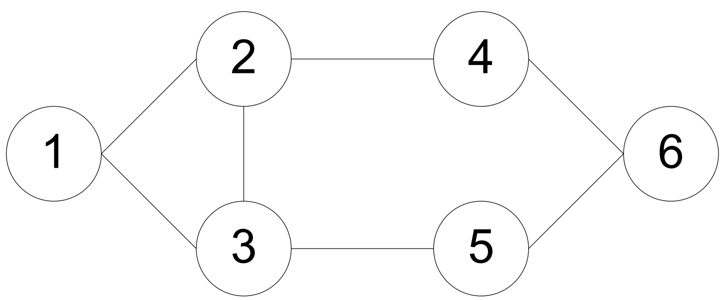
Cliques of size 3:
{1, 2, 3}, {1, 3, 4}, {4, 5, 6}, {5, 6, 7}, {5, 6, 8}, {5, 7, 8}, {6, 7, 8}



Communities:
{1, 2, 3, 4}
{4, 5, 6, 7, 8}

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 <= k
- k-club: a substructure of diameter <= 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}

- A k-clique might have diameter larger than k in the subgraph
- Commonly used in traditional SNA
- Often involves combinatorial optimization

Group-Centric Community Detection: Density-Based Groups

- The group-centric criterion requires the whole group to satisfy a certain condition
  - E.g., the group density >= a given threshold
- A subgraph $G_s(V_s, E_s)$ is a $\gamma - dense$ quasi–clique if

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

- A similar strategy to that of cliques can be used
  - Sample a subgraph, and find a maximal $\gamma - dense$ quasi–clique (say, of size k)
  - Remove nodes with degree $< k\gamma$

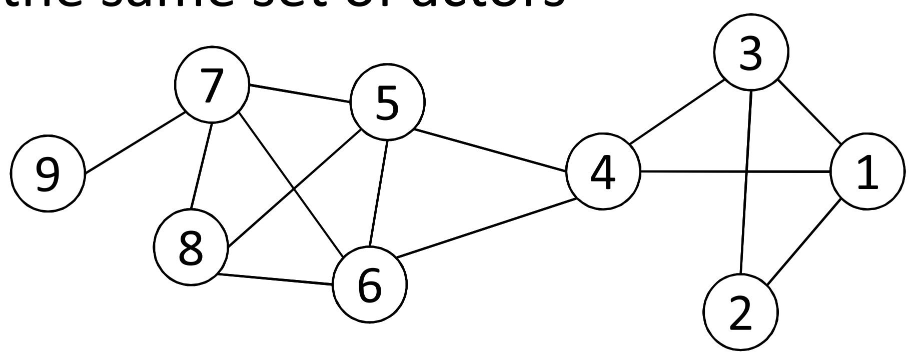
Network-Centric Community Detection

- Network-centric criterion needs to consider the connections within a network globally
- Goal: partition nodes of a network into disjoint sets
- Approaches:
  - Clustering based on vertex similarity
  - Latent space models
  - Block model approximation
  - Spectral clustering
  - Modularity maximization

Clustering based on Vertex Similarity

- Apply k-means or similarity-based clustering to nodes
- Vertex similarity is defined in terms of the similarity of their neighborhood
- Structural equivalence: two nodes are structurally equivalent iff they are connecting to the same set of actors

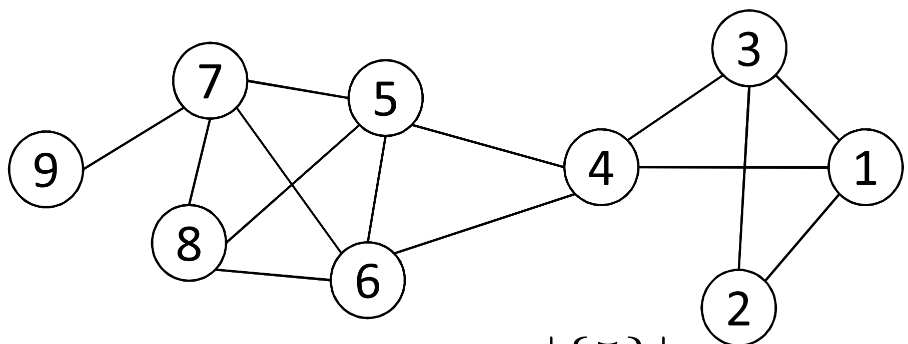
Nodes 1 and 3 are structurally equivalent; So are nodes 5 and 7.



- Structural equivalence is too restrict for practical use.

Vertex Similarity

- Jaccard Similarity $$Jaccard(v_i, v_j) = \frac{|N_i \cup N_j|}{|N_i \cap N_j|}$$
- Cosine similarity $$cosine(v_i, v_j) = \frac{\sum_k A_{ik} A_{jk}}{\sqrt{A_{is}^2 \cdot \sum_t A_{jt}^2}}$$



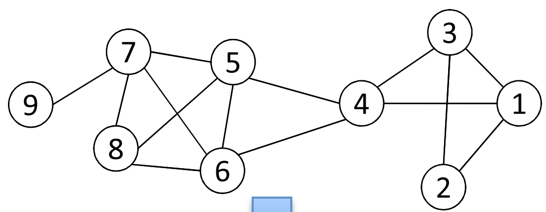
$$Jaccard(4,6) = \frac{|\{5\}|}{|\{1,3,4,5,6,7,8\}|} = \frac{1}{7}$$

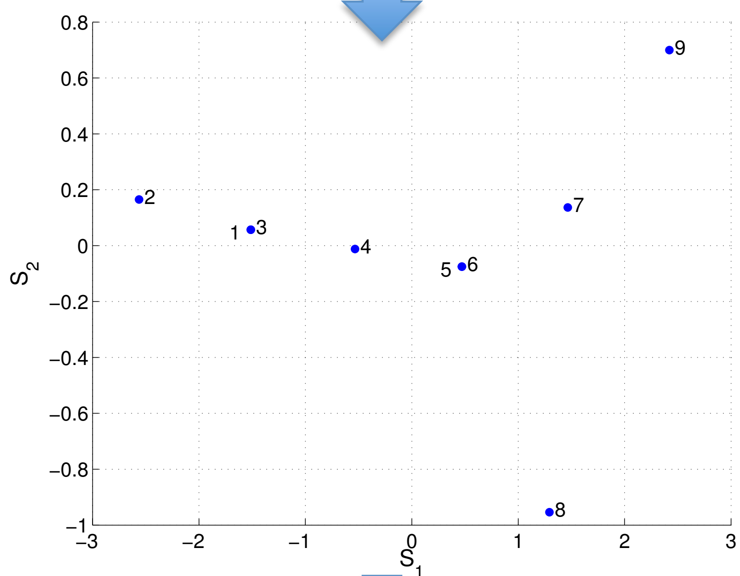
$$cosine(4,6) = \frac{1}{\sqrt{4 \cdot 4}} = \frac{1}{4}$$

Latent Space Models

- Map nodes into a low-dimensional space such that the proximity between nodes based on network connectivity is preserved in the new space, then apply k-means clustering
- Multi-dimensional scaling (MDS)
  - Given a network, construct a proximity matrix P representing the pairwise distance between nodes (e.g., geodesic distance)
  - Let $S \in R^{n \times \ell}$ denote the coordinates of nodes in the low-dimensional space $$SS^T \approx -\frac{1}{2}(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)(P \circ P)(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T) = \widetilde{P}$$
  - Objective function: $\min \|SS^T - \widetilde{P}\|_F^2$
  - Solution: $S = V\Lambda^{\frac{1}{2}}$
  - V is the top ℓ eigenvectors of $\widetilde{P}$, and Λ is a diagonal matrix of top eigenvalues $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_\ell)$

MDS Example





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

geodesic distance

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

$$\widetilde{P} = \begin{bmatrix} 2.46 & 3.96 & 1.96 & 0.85 & -0.65 & -0.65 & -2.21 & -2.04 & -3.65 \\ 3.96 & 6.46 & 3.96 & 1.35 & -1.15 & -1.15 & -3.71 & -3.54 & -6.15 \\ 1.96 & 3.96 & 2.46 & 0.85 & -0.65 & -0.65 & -2.21 & -2.04 & -3.65 \\ 0.85 & 1.35 & 0.85 & 0.23 & -0.27 & -0.27 & -0.82 & -0.65 & -1.27 \\ -0.65 & -1.15 & -0.65 & -0.27 & 0.23 & -0.27 & 0.68 & 0.85 & 1.23 \\ -0.65 & -1.15 & -0.65 & -0.27 & -0.27 & 0.23 & 0.68 & 0.85 & 1.23 \\ -2.21 & -3.71 & -2.21 & -0.82 & 0.68 & 0.68 & 2.12 & 1.79 & 3.68 \\ -2.04 & -3.54 & -2.04 & -0.65 & 0.85 & 0.85 & 1.79 & 2.46 & 2.35 \\ -3.65 & -6.15 & -3.65 & -1.27 & 1.23 & 1.23 & 3.68 & 2.35 & 6.23 \end{bmatrix}$$

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

Block Models

Table 3.1: Adjacency Matrix

-	1	1	1	0	0	0	0	0
1	-	1	0	0	0	0	0	0
1	1	-	1	0	0	0	0	0
1	0	1	-	1	1	0	0	0
0	0	0	1	-	1	1	1	0
0	0	0	1	1	-	1	1	0
0	0	0	0	1	1	-	1	1
0	0	0	0	1	1	1	-	0
0	0	0	0	0	0	1	0	-

$$\min ||A - S\Sigma S^T||_F^2$$

Table 3.2: Ideal Block Structure

1	1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0
0	0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1	1
0	0	0	0	1	1	1	1	1

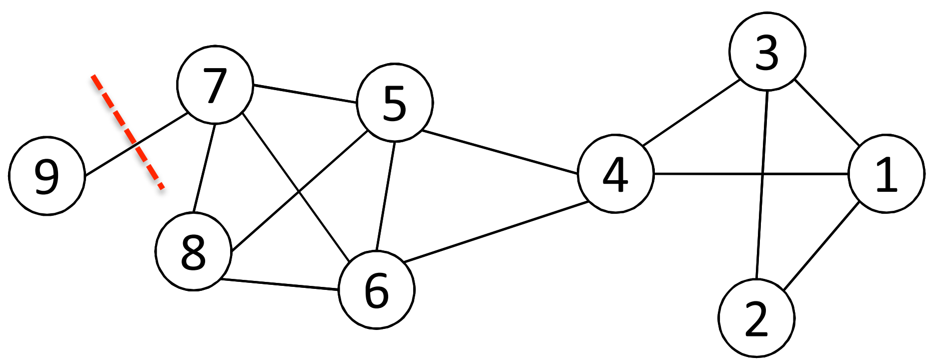
- S is the community indicator matrix
- Relax S to be numerical values, then the optimal solution corresponds to the top eigenvectors of A

$$S = \begin{bmatrix} 0.20 & -0.52 \\ 0.11 & -0.43 \\ 0.20 & -0.52 \\ 0.38 & -0.30 \\ 0.47 & 0.15 \\ 0.47 & 0.15 \\ 0.41 & 0.28 \\ 0.38 & 0.24 \\ 0.12 & 0.11 \end{bmatrix}, \Sigma = \begin{bmatrix} 3.5 & 0 \\ 0 & 2.4 \end{bmatrix}.$$

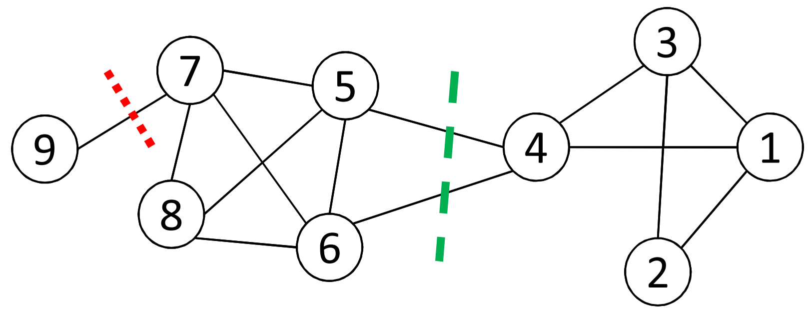
Two communities:
{1, 2, 3, 4} and {5, 6, 7, 8, 9}

Cut

- Most interactions are within group whereas interactions between groups are few
- community detection → minimum cut problem
- Cut: A partition of vertices of a graph into two disjoint sets
- Minimum cut problem: find a graph partition such that the number of edges between the two sets is minimized



Ratio Cut & Normalized Cut



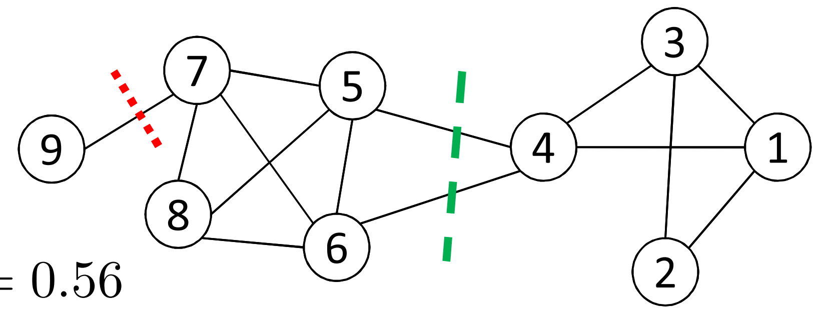
- Minimum cut often returns an imbalanced partition, with one set being a singleton
- Change the objective function to consider community size

$$\text{Ratio Cut}(\pi) = \frac{1}{k}\sum_{i=1}^{k}\frac{cut(C_i, \bar{C}_i)}{|C_i|},$$

$$\text{Normalized Cut}(\pi) = \frac{1}{k}\sum_{i=1}^{k}\frac{cut(C_i, \bar{C}_i)}{vol(C_i)}$$

C_i: a community
$|C_i|$: number of nodes in C_i
$vol(C_i)$: sum of degrees in C_i

Ratio Cut & Normalized Cut Example



For partition in red: π_1

$$\text{Ratio Cut}(\pi_1) = \frac{1}{2}\left(\frac{1}{1} + \frac{1}{8}\right) = 9/16 = 0.56$$

$$\text{Normalized Cut}(\pi_1) = \frac{1}{2}\left(\frac{1}{1} + \frac{1}{27}\right) = 14/27 = 0.52$$

For partition in green: π_2

$$\text{Ratio Cut}(\pi_2) = \frac{1}{2}\left(\frac{2}{4} + \frac{2}{5}\right) = 9/20 = 0.45 < \text{Ratio Cut}(\pi_1)$$

$$\text{Normalized Cut}(\pi_2) = \frac{1}{2}\left(\frac{2}{12} + \frac{2}{16}\right) = 7/48 = 0.15 < \text{Normalized Cut}(\pi_1)$$

Both ratio cut and normalized cut prefer a balanced partition

Spectral Clustering

- Both ratio cut and normalized cut can be reformulated as

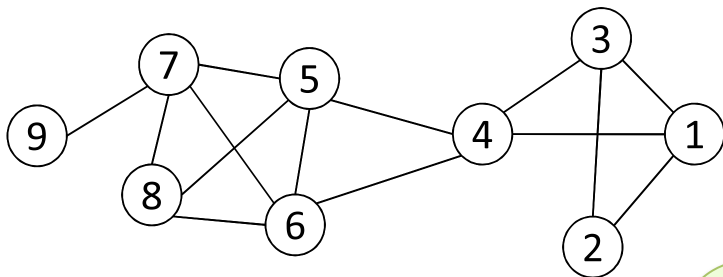
$$\min_{S \in \{0,1\}^{n \times k}} Tr(S^T \widetilde{L} S)$$

- Where $\widetilde{L} = \begin{cases} D - A & \text{graph Laplacian for ratio cut} \\ I - D^{-1/2} A D^{-1/2} & \text{normalized graph Laplacian} \end{cases}$

  $D = diag(d_1, d_2, \cdots, d_n)$ A diagonal matrix of degrees

- Spectral relaxation: $\min_{S} Tr(S^T \widetilde{L} S) \quad s.t. \ S^T S = I_k$
- Optimal solution: top eigenvectors with the smallest eigenvalues

Spectral Clustering Example



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

$$D = diag(3, 2, 3, 4, 4, 4, 4, 3, 1)$$

The 1st eigenvector means all nodes belong to the same cluster, no use

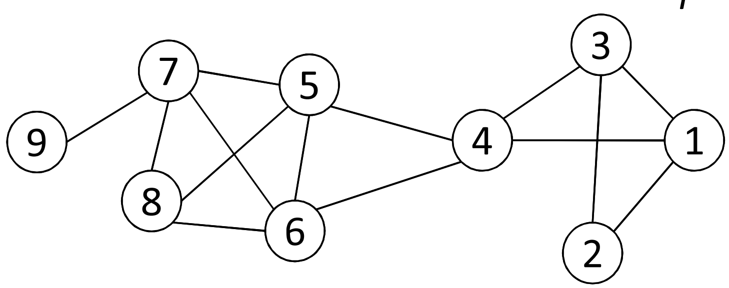
k-means

$$\widetilde{L} = D - A = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 4 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 4 & -1 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 4 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 4 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & -1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}$$

$$S = \begin{bmatrix} 0.33 & -0.38 \\ 0.33 & -0.48 \\ 0.33 & -0.38 \\ 0.33 & -0.12 \\ 0.33 & 0.16 \\ 0.33 & 0.16 \\ 0.33 & 0.30 \\ 0.33 & 0.24 \\ 0.33 & 0.51 \end{bmatrix}$$

Modularity Maximization

- Modularity measures the strength of a community partition by taking into account the degree distribution
- Given a network with *m* edges, the expected number of edges between two nodes with d_i and d_j is $d_i d_j / 2m$



The expected number of edges between nodes 1 and 2 is 3*2/ (2*14) = 3/14

- Strength of a community: $\sum_{i \in C, j \in C} A_{ij} - d_i d_j / 2m$
- Modularity: $Q = \frac{1}{2m} \sum_{\ell=1}^{k} \sum_{i \in C_\ell, j \in C_\ell} (A_{ij} - d_i d_j / 2m)$
- A larger value indicates a good community structure

Modularity Matrix

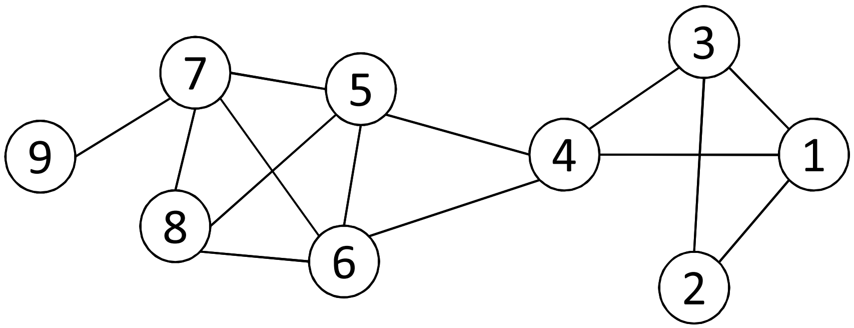
- Modularity matrix: $B = A - \mathbf{d}\mathbf{d}^T/2m \quad (B_{ij} = A_{ij} - d_i d_j/2m)$

- Similar to spectral clustering, Modularity maximization can be reformulated as

$$\max Q = \frac{1}{2m} Tr(S^T B S) \quad s.t.\ S^T S = I_k$$

- Optimal solution: top eigenvectors of the modularity matrix
- Apply k-means to S as a post-processing step to obtain community partition

Modularity Maximization Example



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

k-means

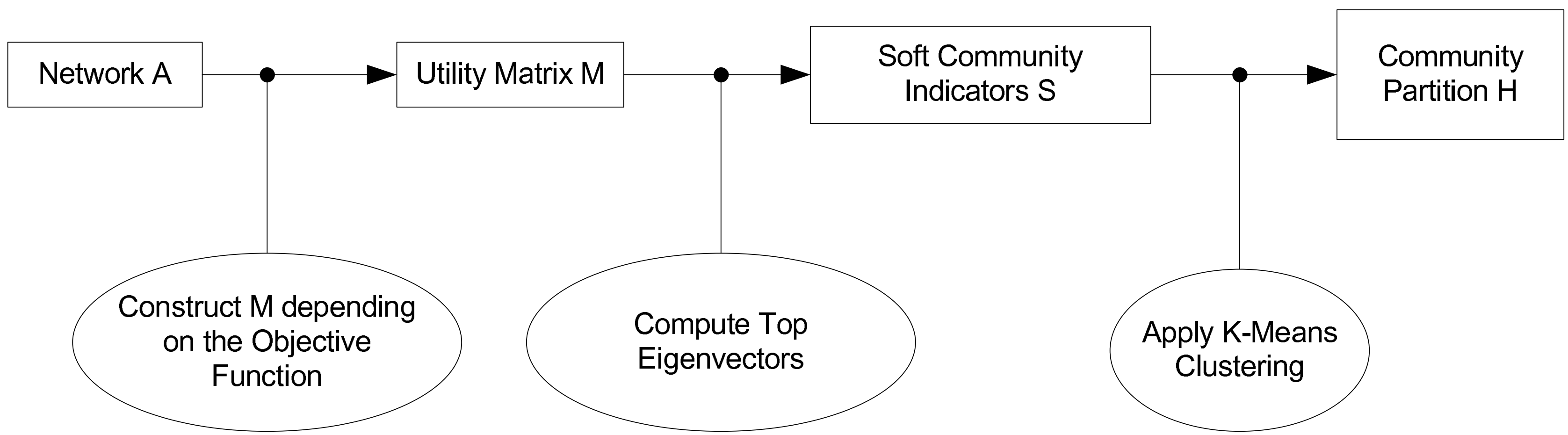
$$B = \begin{bmatrix} -0.32 & 0.79 & 0.68 & 0.57 & -0.43 & -0.43 & -0.43 & -0.32 & -0.11 \\ 0.79 & -0.14 & 0.79 & -0.29 & -0.29 & -0.29 & -0.29 & -0.21 & -0.07 \\ 0.68 & 0.79 & -0.32 & 0.57 & -0.43 & -0.43 & -0.43 & -0.32 & -0.11 \\ 0.57 & -0.29 & 0.57 & -0.57 & 0.43 & 0.43 & -0.57 & -0.43 & -0.14 \\ -0.43 & -0.29 & -0.43 & 0.43 & -0.57 & 0.43 & 0.43 & 0.57 & -0.14 \\ -0.43 & -0.29 & -0.43 & 0.43 & 0.43 & -0.57 & 0.43 & 0.57 & -0.14 \\ -0.43 & -0.29 & -0.43 & -0.57 & 0.43 & 0.43 & -0.57 & 0.57 & 0.86 \\ -0.32 & -0.21 & -0.32 & -0.43 & 0.57 & 0.57 & 0.57 & -0.32 & -0.11 \\ -0.11 & -0.07 & -0.11 & -0.14 & -0.14 & -0.14 & 0.86 & -0.11 & -0.04 \end{bmatrix}$$

Modularity Matrix

$$S = \begin{bmatrix} 0.44 & -0.00 \\ 0.38 & 0.23 \\ 0.44 & -0.00 \\ 0.17 & -0.48 \\ -0.29 & -0.32 \\ -0.29 & -0.32 \\ -0.38 & 0.34 \\ -0.34 & -0.08 \\ -0.14 & 0.63 \end{bmatrix}$$

A Unified View for Community Partition

- Latent space models, block models, spectral clustering, and modularity maximization can be unified as



$$\text{Utility Matrix } M = \begin{cases} \text{modified proximity matrix } \widetilde{P} & \textit{if} \text{ latent space models} \\ \text{adjacency matrix } A & \textit{if} \text{ block models} \\ \text{graph Laplacian } \widetilde{L} & \textit{if} \text{ spectral clustering} \\ \text{modularity maximization } \; B & \textit{if} \text{ modularity maximization} \end{cases}$$

Hierarchy-Centric Community Detection

- Goal: build a hierarchical structure of communities based on network topology
- Allow the analysis of a network at different resolutions
- Representative approaches:
  - Divisive Hierarchical Clustering
  - Agglomerative Hierarchical clustering

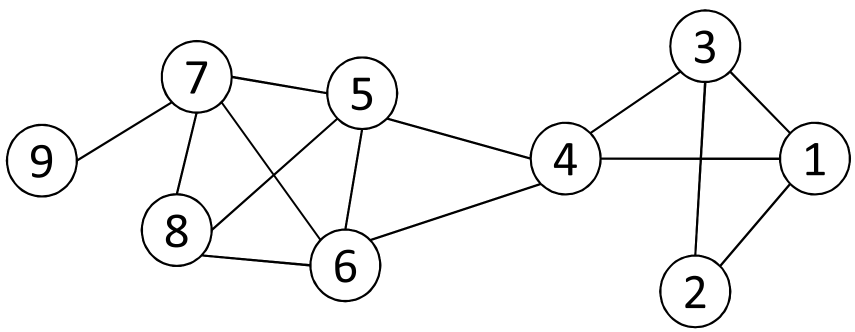
Divisive Hierarchical Clustering

- Divisive clustering
  - Partition nodes into several sets
  - Each set is further divided into smaller ones
  - Network-centric partition can be applied for the partition
- One particular example: recursively remove the “weakest” tie
  - Find the edge with the least strength
  - Remove the edge and update the corresponding strength of each edge
- Recursively apply the above two steps until a network is discomposed into desired number of connected components.
- Each component forms a community

Edge Betweenness

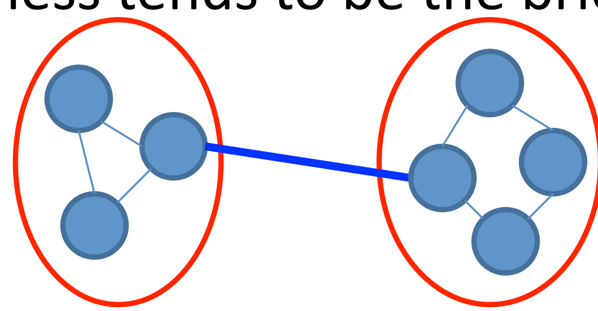
- The strength of a tie can be measured by edge betweenness
- Edge betweenness: the number of shortest paths that pass along with the edge

$$\text{edge-betweenness(e)} = \Sigma_{s<t} \frac{\sigma_{st}(e)}{\sigma_{s,t}}$$

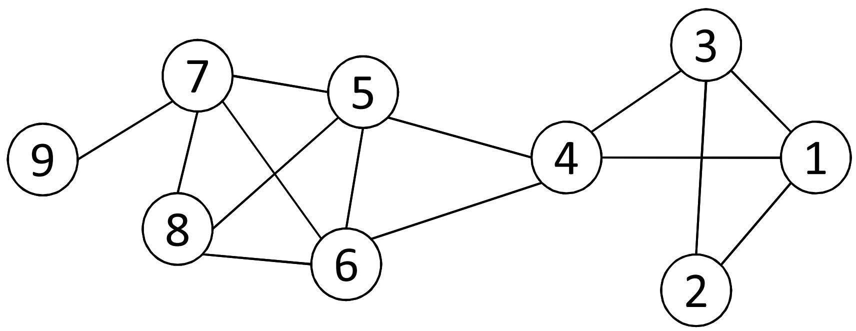


The edge betweenness of e(1, 2) is 4, as all the shortest paths from 2 to {4, 5, 6, 7, 8, 9} have to either pass e(1, 2) or e(2, 3), and e(1,2) is the shortest path between 1 and 2

- The edge with higher betweenness tends to be the bridge between two communities.



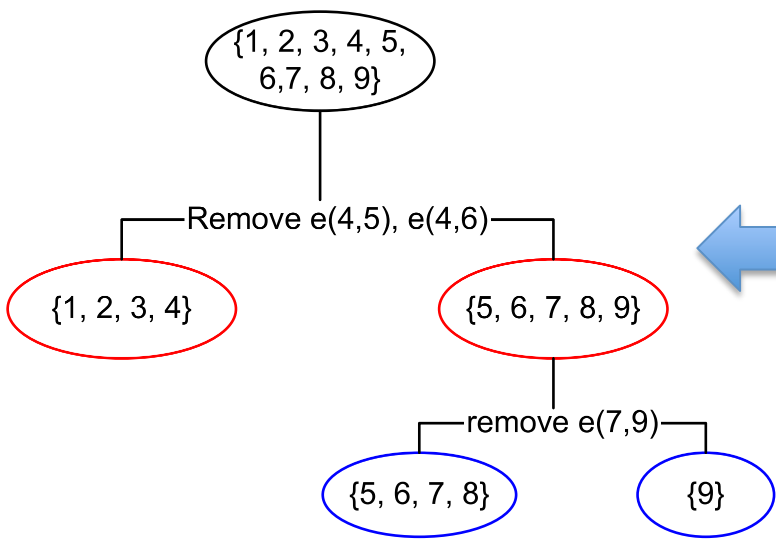
Divisive clustering based on edge betweenness



Initial betweenness value

Table 3.3: Edge Betweenness

	1	2	3	4	5	6	7	8	9
1	0	4	1	9	0	0	0	0	0
2	4	0	4	0	0	0	0	0	0
3	1	4	0	9	0	0	0	0	0
4	9	0	9	0	10	10	0	0	0
5	0	0	0	10	0	1	6	3	0
6	0	0	0	10	1	0	6	3	0
7	0	0	0	0	6	6	0	2	8
8	0	0	0	0	3	3	2	0	0
9	0	0	0	0	0	0	8	0	0

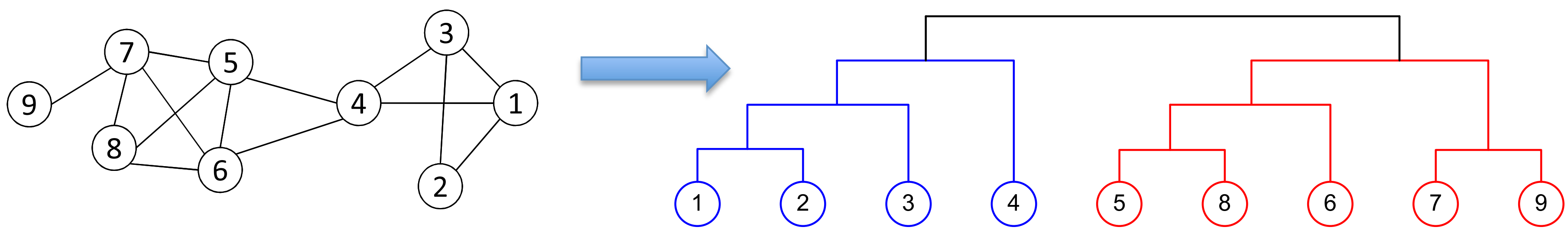


After remove e(4,5), the betweenness of e(4, 6) becomes 20, which is the highest;

After remove e(4,6), the edge e(7,9) has the highest betweenness value 4, and should be removed.

Agglomerative Hierarchical Clustering

- Initialize each node as a community
- Merge communities successively into larger communities following a certain criterion
  - E.g., based on modularity increase



Summary of Community Detection

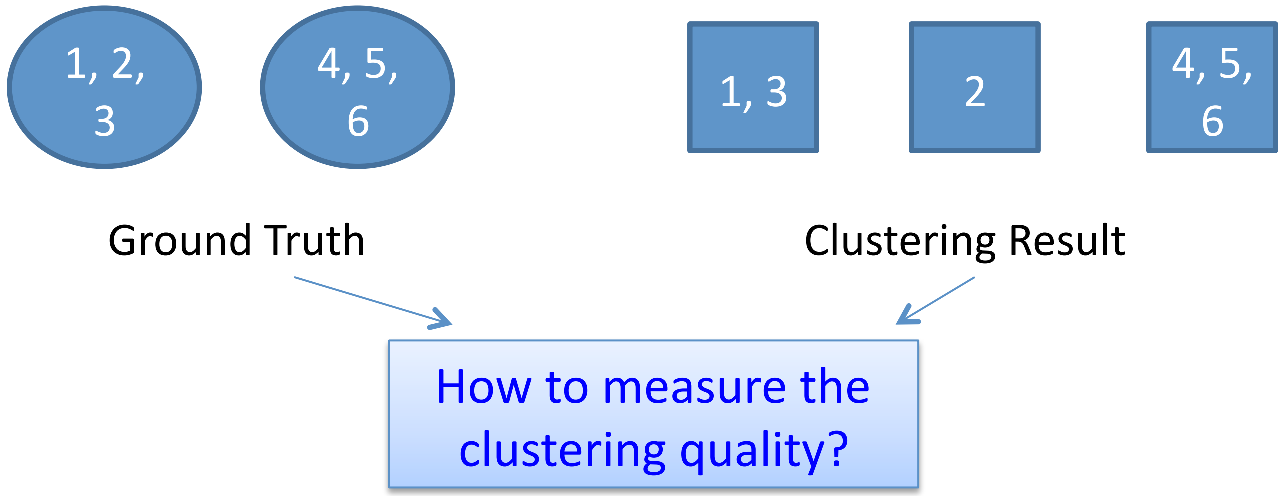
- Node-Centric Community Detection
  - *cliques, k-cliques, k-clubs*
- Group-Centric Community Detection
  - *quasi-cliques*
- Network-Centric Community Detection
  - *Clustering based on vertex similarity*
  - *Latent space models, block models, spectral clustering, modularity maximization*
- Hierarchy-Centric Community Detection
  - *Divisive clustering*
  - *Agglomerative clustering*

COMMUNITY EVALUATION

Evaluating Community Detection (1)

- For groups with clear definitions
  - E.g., Cliques, k-cliques, k-clubs, quasi-cliques
  - Verify whether extracted communities satisfy the definition
- For networks with ground truth information
  - Normalized mutual information
  - Accuracy of pairwise community memberships

Measuring a Clustering Result



- The number of communities after grouping can be different from the ground truth
- No clear community correspondence between clustering result and the ground truth
- Normalized Mutual Information can be used

Normalized Mutual Information

- Entropy: the information contained in a distribution

$$H(X) = \sum_{x \in X} p(x) \log p(x)$$

- Mutual Information: the shared information between two distributions

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p_1(x) p_2(y)} \right)$$

- Normalized Mutual Information (between 0 and 1)

$$NMI(X;Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}$$

- Consider a partition as a distribution (probability of one node falling into one community), we can compute the matching between two clusterings

NMI

$$H(X) = \sum_{x \in X} p(x) \log p(x)$$

$$H(\pi^a) = \sum_{h}^{k^{(a)}} \frac{n_h^a}{n} \log(\frac{n_h^a}{n})$$

$$H(\pi^b) = \sum_{\ell}^{k^{(b)}} \frac{n_\ell^b}{n} \log(\frac{n_\ell^b}{n})$$

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p_1(x) p_2(y)} \right)$$

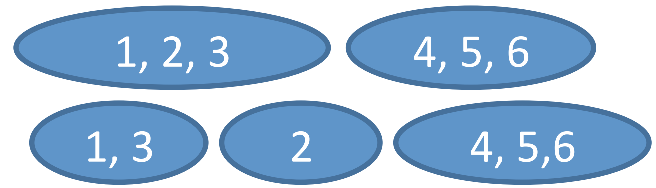
$$I(\pi^a, \pi^b) = \sum_{h} \sum_{\ell} \frac{n_{h,\ell}}{n} \log \left(\frac{\frac{n_{h,\ell}}{n}}{\frac{n_h^a}{n} \frac{n_\ell^b}{n}} \right)$$

$$NMI(X;Y) = \frac{I(X;Y)}{\sqrt{H(X)H(Y)}}$$

$$NMI(\pi^a, \pi^b) = \frac{\sum_{h=1}^{k^{(a)}} \sum_{\ell=1}^{k^{(b)}} n_{h,\ell} \log \left(\frac{n \cdot n_{h,l}}{n_h^{(a)} \cdot n_\ell^{(b)}} \right)}{\sqrt{\left(\sum_{h=1}^{k^{(a)}} n_h^{(a)} \log \frac{n_h^a}{n} \right) \left(\sum_{\ell=1}^{k^{(b)}} n_\ell^{(b)} \log \frac{n_\ell^b}{n} \right)}}$$

NMI-Example

- Partition a: [1, 1, 1, 2, 2, 2]
- Partition b: [1, 2, 1, 3, 3, 3]



$$n = 6$$
$$k^{(a)} = 2$$
$$k^{(b)} = 3$$

	n_h^a
h=1	3
h=2	3

	n_l^b
l=1	2
l=2	1
l=3	3

$n_{h,l}$	l=1	l=2	l=3
h=1	2	1	0
h=2	0	0	3

$$NMI(\pi^a, \pi^b) = \frac{\sum_{h=1}^{k^{(a)}} \sum_{\ell=1}^{k^{(b)}} n_{h,\ell} \log\left(\frac{n \cdot n_{h,l}}{n_h^{(a)} \cdot n_\ell^{(b)}}\right)}{\sqrt{\left(\sum_{h=1}^{k^{(a)}} n_h^{(a)} \log \frac{n_h^a}{n}\right)\left(\sum_{\ell=1}^{k^{(b)}} n_\ell^{(b)} \log \frac{n_\ell^b}{n}\right)}}$$

=0.8278

Accuracy of Pairwise Community Memberships

- Consider all the possible pairs of nodes and check whether they reside in the same community
- An error occurs *if*
  - Two nodes belonging to the same community are assigned to different communities after clustering
  - Two nodes belonging to different communities are assigned to the same community
- Construct a contingency table

		Ground Truth	
		$C(v_i) = C(v_j)$	$C(v_i) \neq C(v_j)$
Clustering Result	$C(v_i) = C(v_j)$	a	b
	$C(v_i) \neq C(v_j)$	c	d

$$accuracy = \frac{a+d}{a+b+c+d} = \frac{a+d}{n(n-1)/2}$$

Accuracy Example

Ground Truth Clustering Result

		Ground Truth	
		$C(v_i) = C(v_j)$	$C(v_i) != C(v_j)$
Clustering Result	$C(v_i) = C(v_j)$	4	0
	$C(v_i) != C(v_j)$	2	9

Accuracy = (4+9)/ (4+2+9+0) = 13/15

Evaluation using Semantics

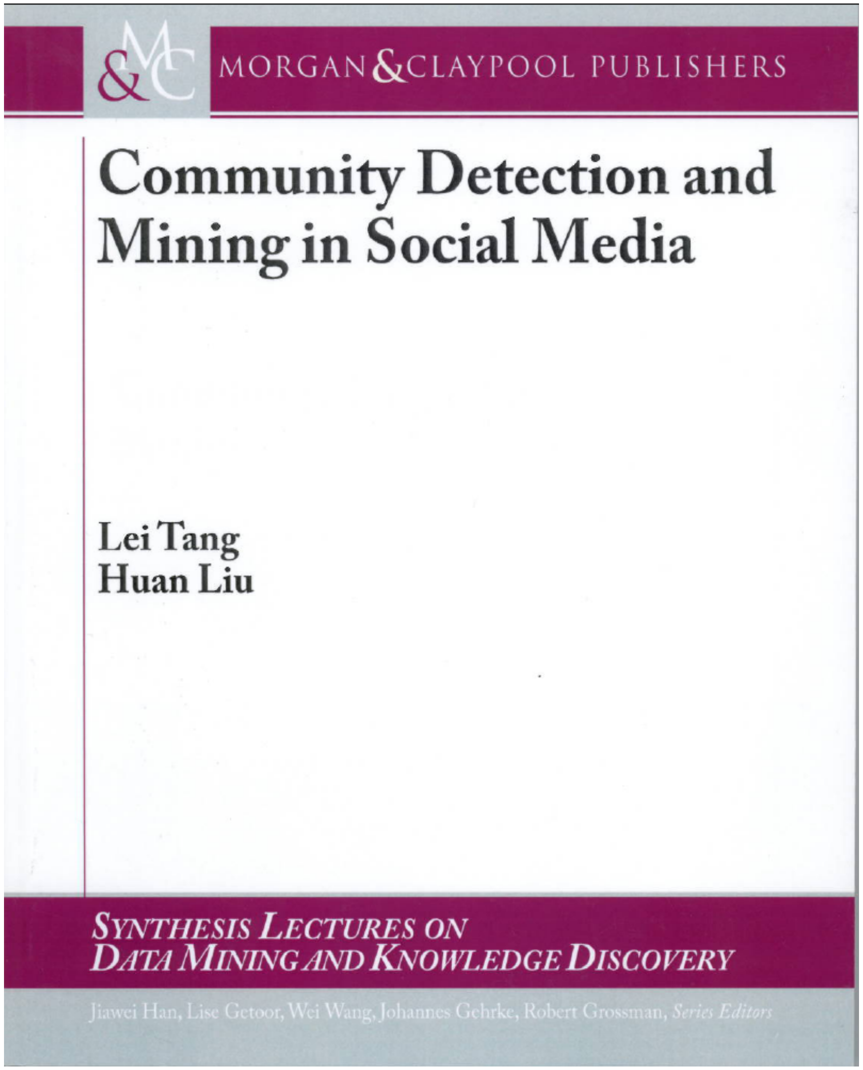
- For networks with semantics
  – Networks come with semantic or attribute information of nodes or connections
  – Human subjects can verify whether the extracted communities are coherent
- Evaluation is qualitative
- It is also intuitive and helps understand a community

An *animal* community

A *health* community

Evaluation without Ground Truth

- For networks without ground truth or semantic information
- This is the most common situation
- An option is to resort to cross-validation
  - Extract communities from a (training) network
  - Evaluate the quality of the community structure on a network constructed from a different date or based on a related type of interaction
- Quantitative evaluation functions
  - modularity
  - block model approximation error



Book Available at

- Morgan & claypool Publishers
- Amazon

If you have any comments, please feel free to contact:

- **Lei Tang**, Yahoo! Labs, ltang@yahoo-inc.com
- **Huan Liu**, ASU huanliu@asu.edu