Community Detection

Community

In social sciences:

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

Taxonomy of Community Criteria

- 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

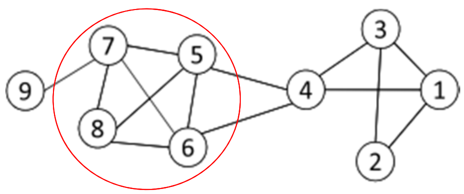
Node-Centric Community Detection

- 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

53

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
Cliques of size 3:
- 1,2, and 3
- 1,3, and 4
- 4,5, and 6

- NP-hard to find the maximum clique in a network
  - Hard to approx within $n^{1-\varepsilon}$ [Håstad, Acta Mathematica, 1999]
- Straightforward implementation to find cliques is very expensive in time complexity

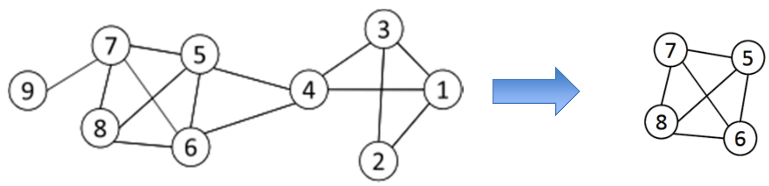
54

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

55

Maximum Clique Example



- Suppose we sample a sub-network with nodes {1-9} 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

56

GreedyMaxClique

- Works well for B-A like graphs
- A greedy algorithms:
  - Start with the highest degree node
  - Iteratively examine nodes in decreasing degree order
  - If node connects tp all nodes in the group - add it to the group
- Complexity O(|E|) or O(d^2)

[Siganos et al., J. of Communications and Networks, 2006]

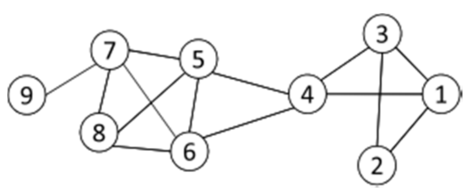
Clique Percolation Method (CPM)

- 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

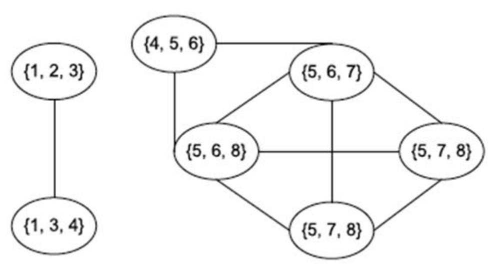
58

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}

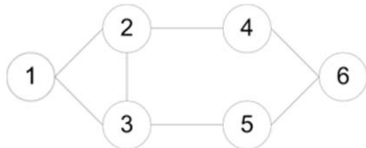


59

Notation hazard

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 <u>geodesic distance</u> between any two nodes <= k
- k-club: a substructure of <u>diameter</u> <= 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
  - E.g. {1, 2, 3, 4, 5}
- Commonly used in traditional SNA
- Often involves combinatorial optimization

60

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

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

where the denominator is the maximum number of degrees.

- A similar strategy to that of cliques can be used
  - Sample a subgraph, and find a maximal $\gamma - dense$ quasi-clique (say, of size $|V_s|$)
  - Remove nodes with degree <u>less than</u> the average degree

< $|V_s|\gamma \leq \frac{2|E_s|}{|V_s|-1}$

61

A Sub-linear Algorithm

- Given a "B-A like graph"
- Find a dense quasi-clique in sublinear time
  - *(k,ε)-dense-core*
  - $\tilde{O}(n^{1-\frac{\beta}{2}})$, where $\beta \leq 2/5$, $k = O(\log n)$

[Gonen *et al.*, Comp. Net., 2008]

Definitions

Definition 1. Closeness to a clique: *Let C^k denote the k-vertex clique. Denote by $dist(G, C^k)$ the distance (as a fraction of $\binom{k}{2}$) between a graph G over k vertices and C^k. Namely, if $dist(G, C^k) = \epsilon$ then $\epsilon\binom{k}{2}$ edges should be added in order to make G into a clique. A graph G over k vertices is ϵ-close to being a clique if $dist(G, C^k) \leq \epsilon$.*

Definition 2. (k, ϵ)-dense-core: *consider a graph G. A subset of k vertices in the graph is a* (k, ϵ)-dense-core *if the subgraph induced by this set is ϵ-close to a clique.*

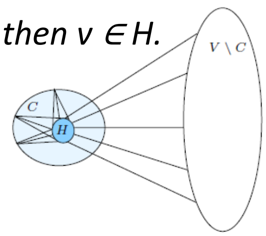
Definition 3. *Let C be a subset of vertices of a graph G. The* d-nucleus of C, *denoted by H, is the subset of vertices of C with degree* (not *induced degree) at least d.*

For a set of vertices X, let $\Gamma(X)$ denote the set of vertices that neighbor at least one vertex in X, and let $\Gamma_\delta(X)$ denote the set of vertices that neighbor all but at most $\delta|X|$ vertices in X. We next introduce our main definition.

(k, d, c, ε)-Jellyfish subgraph

A graph G contains a (k, d, c, ε)-Jellyfish subgraph if it contains a subset C of vertices, with $|C| = k$, that is a (k,ε)-dense-core, which has a non-empty d-nucleus H, s.t., the following conditions hold:

1. *For all $v \in C$, v neighbors at least $(1 - \varepsilon)|H|$ vertices in H.*
2. *For all but $\varepsilon\,|\Gamma_{3\varepsilon}(H)|$ vertices, if a vertex $v \in V$ neighbors at least $(1 - \varepsilon)|H|$ vertices in H then v has at least $(1-\varepsilon)|C|$ neighbors in C.*
3. *For all but $|H|$ vertices in G, if $deg(v) \geq d$ then $v \in H$.*
4. $|\Gamma_{3\varepsilon}(H)|/|C| \leq c$.



A short pause

- We looked at finding max cliques and quasi-cliques
- This will give us the largest community
  - The core of the network
- What about the other communities?
  - Need an algorithms for all cliques

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:
  - (1) Clustering based on vertex similarity
  - **(2) Latent space models (multi-dimensional scaling)**
  - (3) Block model approximation
  - **(4) Spectral clustering**
  - **(5) Modularity maximization**

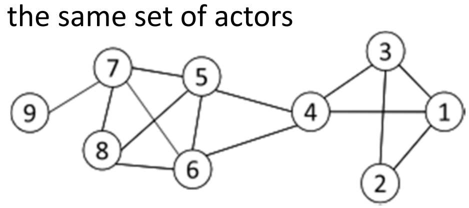
66

(1) Clustering based on vertex similarity

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 6.



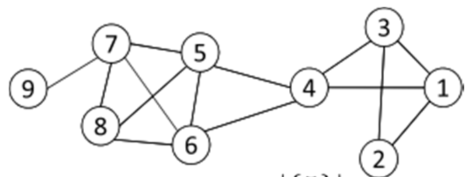
- Structural equivalence is too restrict for practical use.

67

(1) Clustering based on vertex similarity

Vertex Similarity

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



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

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

68

(2) Latent space models

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 <u>pairwise distance</u> between nodes (e.g., geodesic distance)
  - Let $S \in R^{n \times l}$ denote the coordinates of nodes in the low-dimensional space $SS^T \approx -\frac{1}{2}(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)(P \circ P)(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T) = \widetilde{P}$

Centered matrix

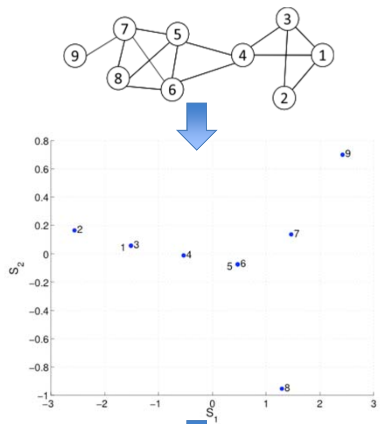
  - Objective function: $\min \|SS^T - \widetilde{P}\|_F^2$
  - Solution: $S = V\Lambda^{\frac{1}{2}}$
  - V is the top ℓ eigenvectors of $\widetilde{P}$, and Λ is a diagonal matrix of top eigenvalues $\Lambda = diag(\lambda_1, \lambda_2, \cdots, \lambda_\ell)$

Reference: http://www.cse.ust.hk/~weikep/notes/MDS.pdf

69

(2) Latent space models

MDS Example



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

geodesic distance

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

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

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

70

(3) Block model approximation

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 (group memberships)
- Relax S to be numerical values, then the optimal solution corresponds to the top eigenvectors of A

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

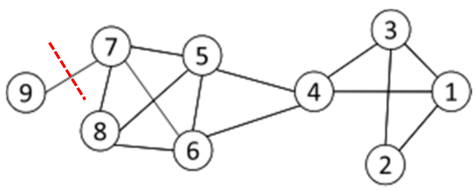
Two communities:
{1, 2, 3, 4} and {5, 6, 7, 8, 9}

71

(4) Spectral clustering

Cut

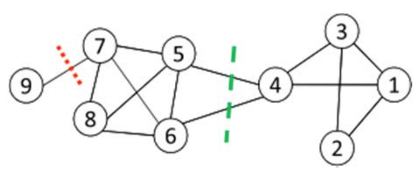
- Most interactions are within group whereas interactions between groups are few
- community detection → minimum cut problem
- Cut: A partition of vertices of a graph into two disjoint sets
- Minimum cut problem: find a graph partition such that the number of edges between the two sets is minimized



72

(4) Spectral clustering

Ratio Cut & Normalized Cut



- Minimum cut often returns an imbalanced partition, with one set being a singleton, e.g. node 9
- Change the objective function to consider community size

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

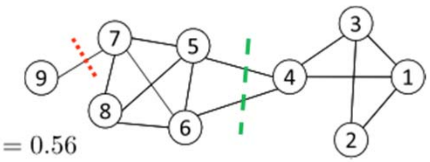
$$\text{Normalized Cut}(\pi) = \frac{1}{k}\sum_{i=1}^{k}\frac{cut(C_i, \bar{C}_i)}{vol(C_i)}$$

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

73

(4) Spectral clustering

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

74

(4) Spectral clustering

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

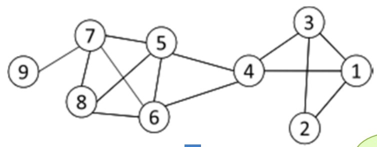
- Spectral relaxation: $\min_S Tr(S^T \widetilde{L} S) \quad s.t.\ S^T S = I_k$
- Optimal solution: top eigenvectors with the smallest eigenvalues

Reference: http://www.cse.ust.hk/~weikep/notes/clustering.pdf

75

(4) Spectral clustering

Spectral Clustering Example



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

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

k-means

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

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

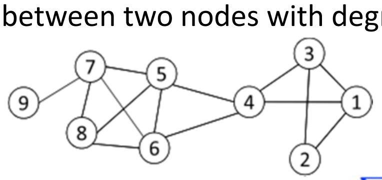
Centered matrix

76

(5) Modularity maximization

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

Given the degree distribution

- 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

77

(5) Modularity maximization

Modularity Matrix

Centered 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

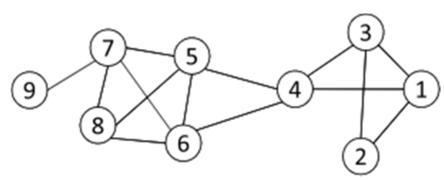
$$\max Q = \frac{1}{2m} Tr(S^T B S) \qquad s.t.\ S^T S = I_k$$

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

78

(5) Modularity maximization

Modularity Maximization Example



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

k-means

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

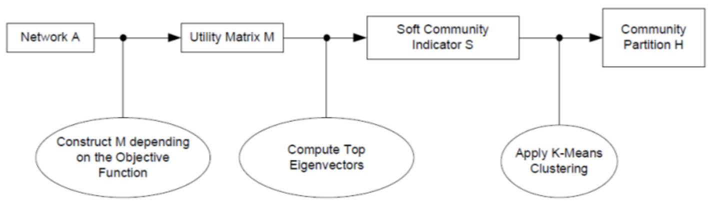
Modularity Matrix

$$\begin{matrix} 0.4384 & -0.2709 \\ 0.3809 & 0.2671 \\ 0.4384 & -0.2709 \\ 0.1716 & 0.6063 \\ -0.2861 & -0.3487 \\ -0.2861 & -0.3487 \\ -0.3754 & 0.3355 \\ -0.3421 & 0.1855 \\ -0.1396 & -0.1552 \end{matrix}$$

79

A Unified View for Community Partition

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



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

Reference: http://www.cse.ust.hk/~weikep/notes/Script_community_detection.m

80

Hierarchy-Centric Community Detection

- Goal: build a <u>hierarchical structure</u> of communities based on network topology
- Allow the analysis of a network <u>at different resolutions</u>
- Representative approaches:
  - Divisive Hierarchical Clustering (top-down)
  - Agglomerative Hierarchical clustering (bottom-up)

81