

Complex Dynamical Networks:

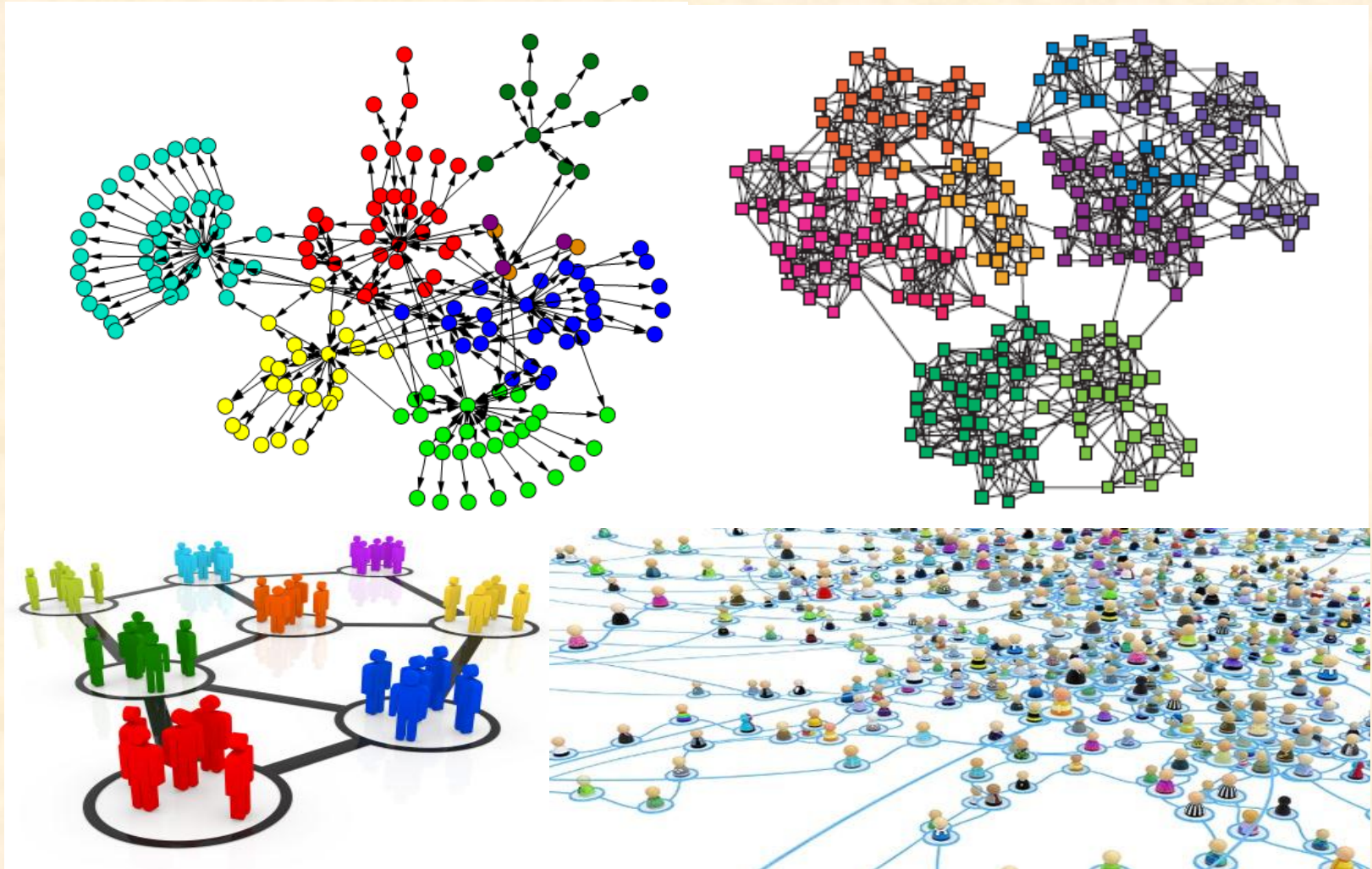
Lecture 6b: Community Structures

EE 6605

Instructor: G Ron Chen



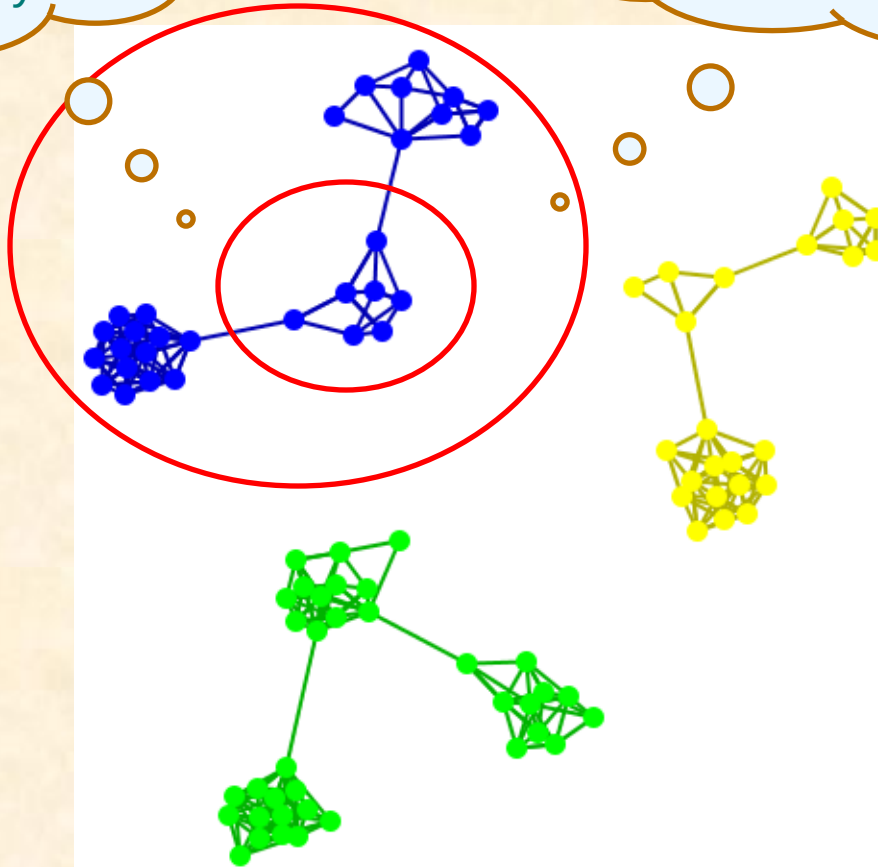
Community Structure in Complex Networks



Community

Each densely-linked group is a community

Each symmetrical group is a community



Definition of a community can be subjective



Community Detection

- ❖ Typical criteria: A **community** is a set of nodes among which the connections are relatively dense and strong, or interactions are relatively frequent
- ❖ Community Detection
(grouping, clustering) finding cohesive subgroups
Input: a complex network
Output: a community structure
- ❖ Applications
Understanding interactions between people (or systems)
Visualizing and navigating on large-scale networks
Forming bases for other tasks such as data mining
...

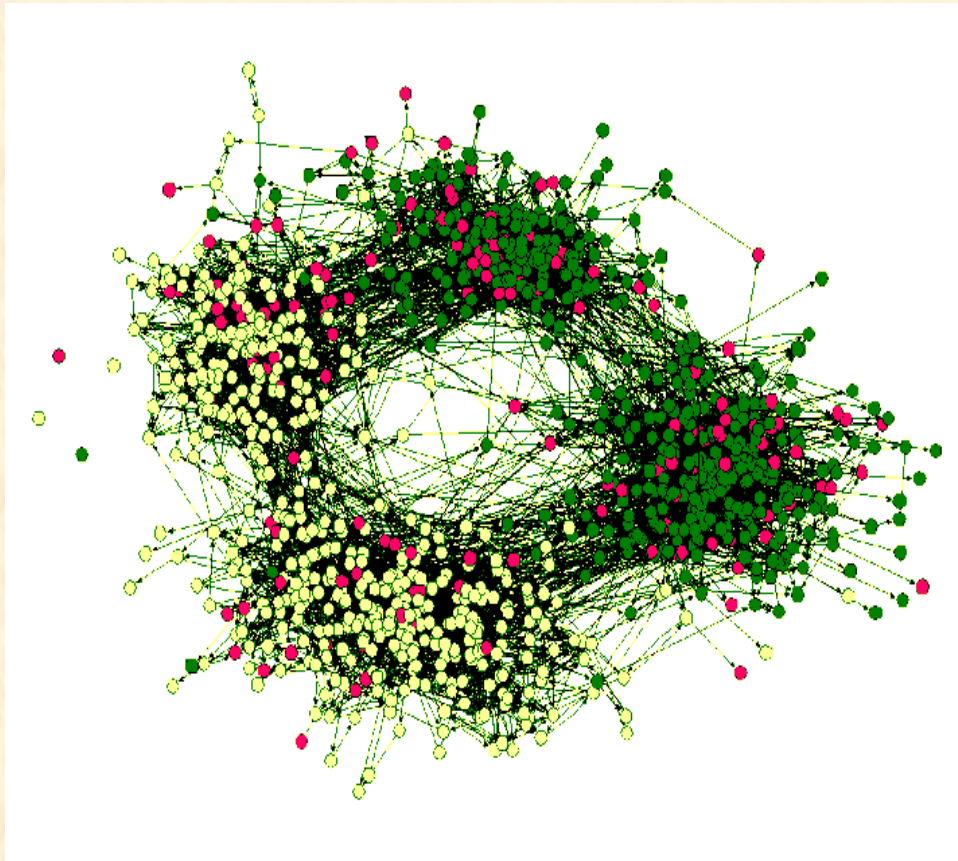
Community Structure

A network is said to have a **community structure** if the nodes of the network can be easily grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally

In the particular case of **non-overlapping community finding**, this implies that the network divides naturally into groups of nodes with dense internal connections and sparse external connections

For an **overlapping community**, node pairs are more likely to be connected if they are both members of the same community, but less likely to be connected if they do not share common communities

Example: School integration and friendship segregation in America (2001)



Race: left (white) to right (black)

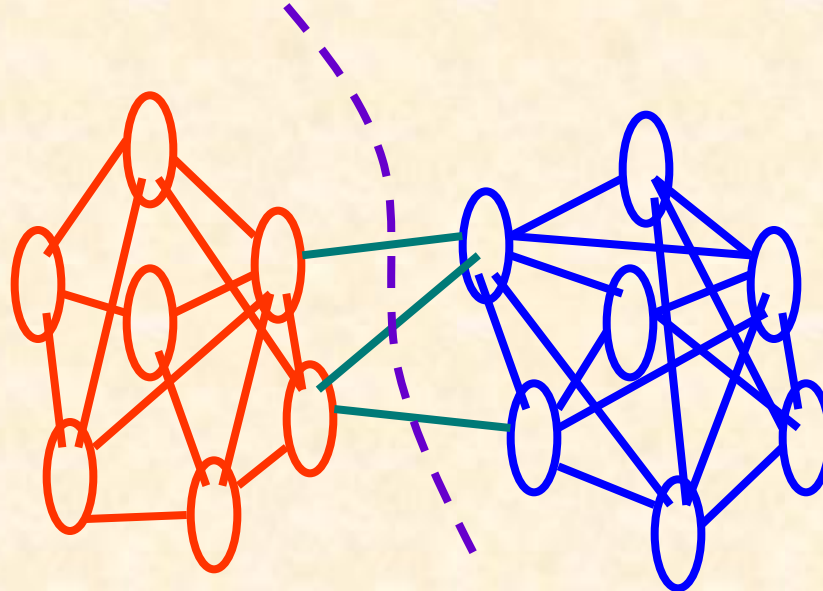
Grade: up (junior) to down (senior)

J. Moody, Amer. J. of Sociology, 2001

Community Detection

Many real networks have a natural community structure, and we want to discover this structure

For a large-scale network, how can we discover community structure in an automated way?



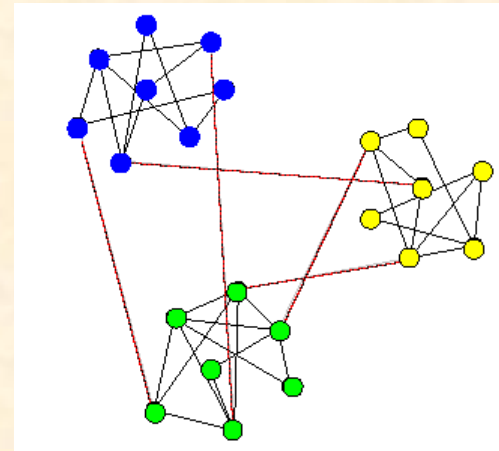
Criteria: Which partition is better ?

- **Benchmarks:**

Which algorithm is the best to characterize a given network with a known community structure?

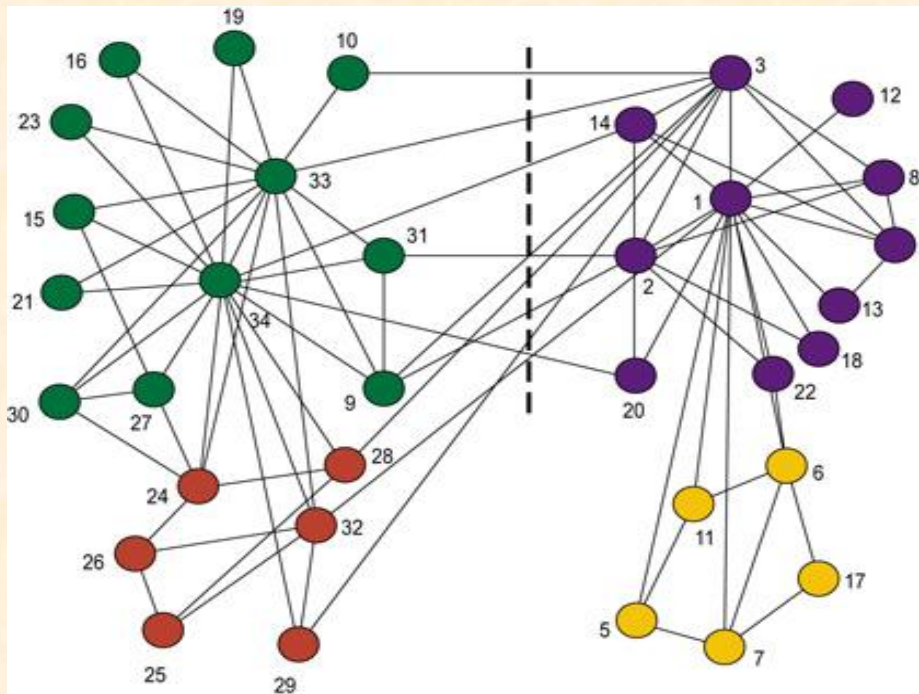
- **Quality Measures:**

How to evaluate an algorithm's performance while the community structure is unknown?

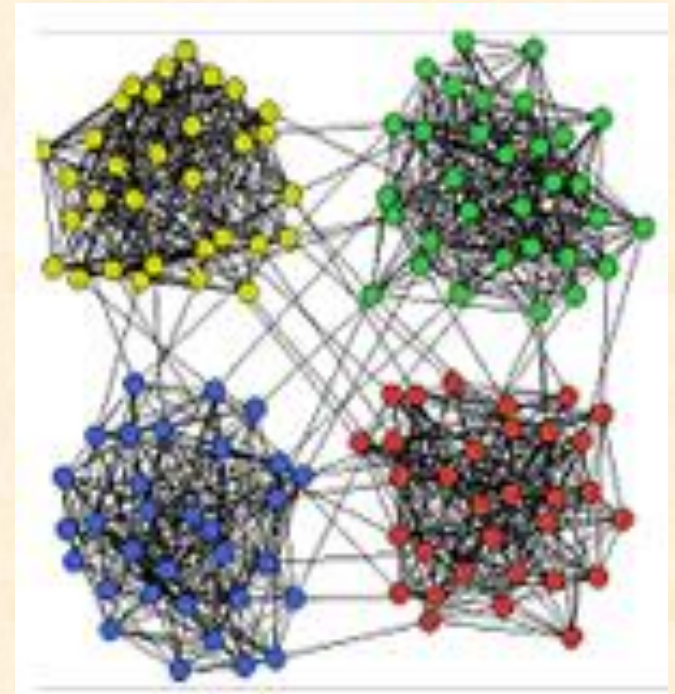


Benchmark Examples

A real network:
Zachary's Karate Network



An artificial network:
Planted *l*-partition model



W. W. Zachary, J. Anthropological Research **33**: 452-473, 1977

A. Condon and R. M. Karp, Random Structures & Algorithms **18**(2): 116-140 , 2001

Zachary's Karate Network

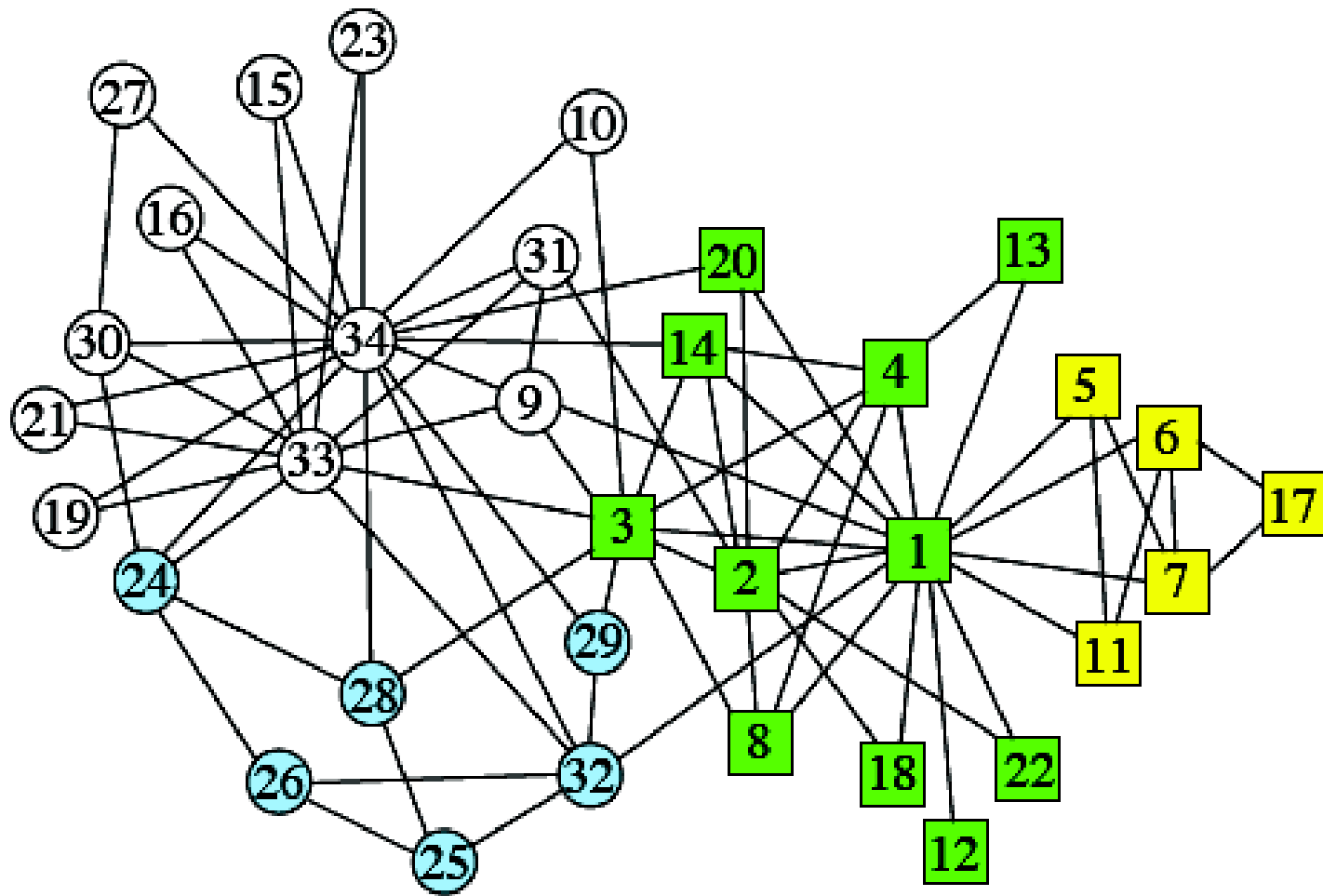


Story Background and Dataset

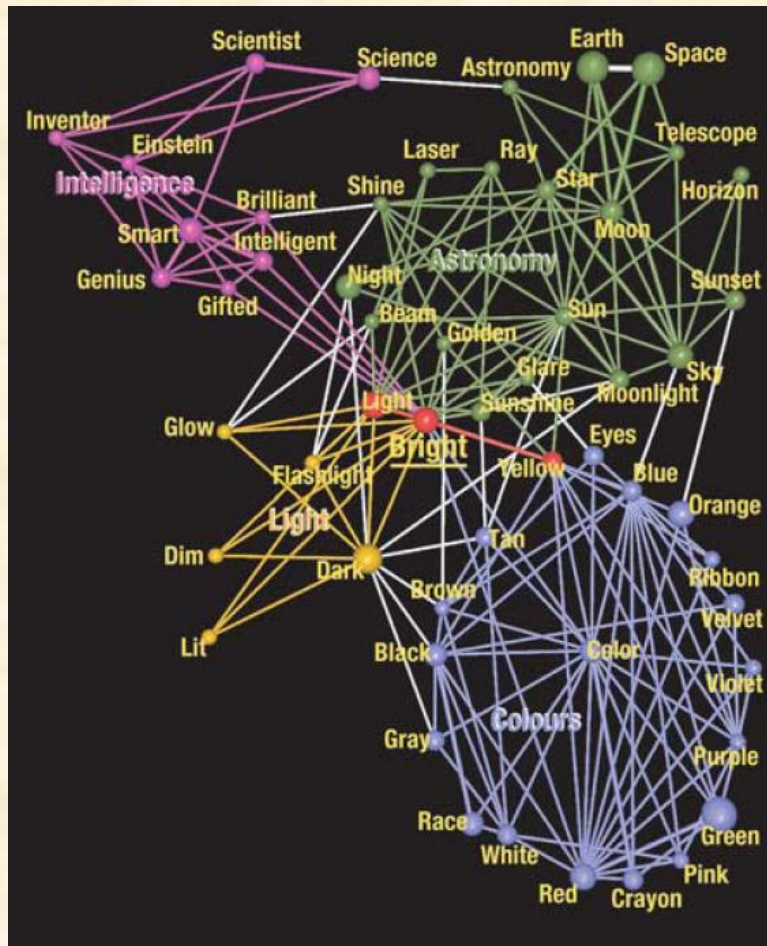
Another Dataset

Yet Another Dataset

Zachary's Karate Network

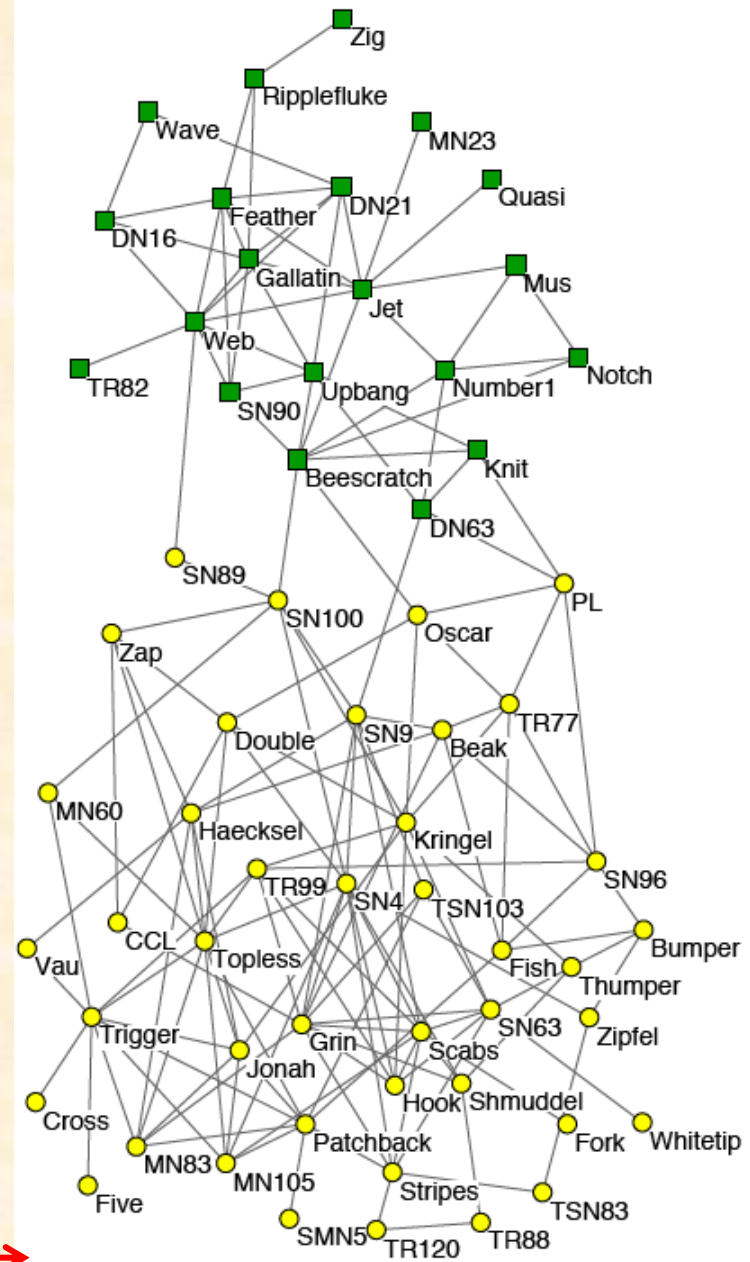


Other Benchmark Examples



word
network ↑

Lusseau's network of
bottlenose dolphins →



Planted l -Partition Model

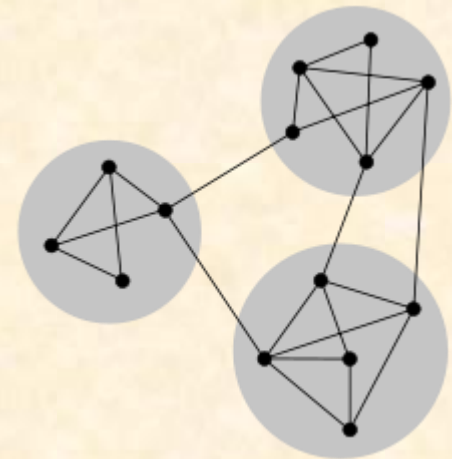
$$l = 2, 3, \dots$$

Partition the graph with N nodes into $n = N/l$ parts

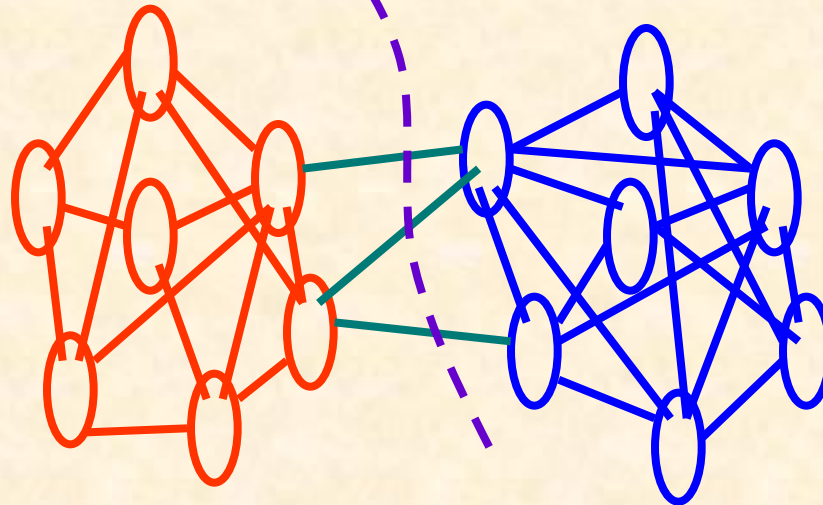
Each node has a probability p_{in} of being connected to nodes inside its group and a probability p_{out} of being connected to nodes outside its group

If $p_{in} > p_{out}$ then the graph has a community structure

Otherwise, it is a homogeneous (e.g., random) graph

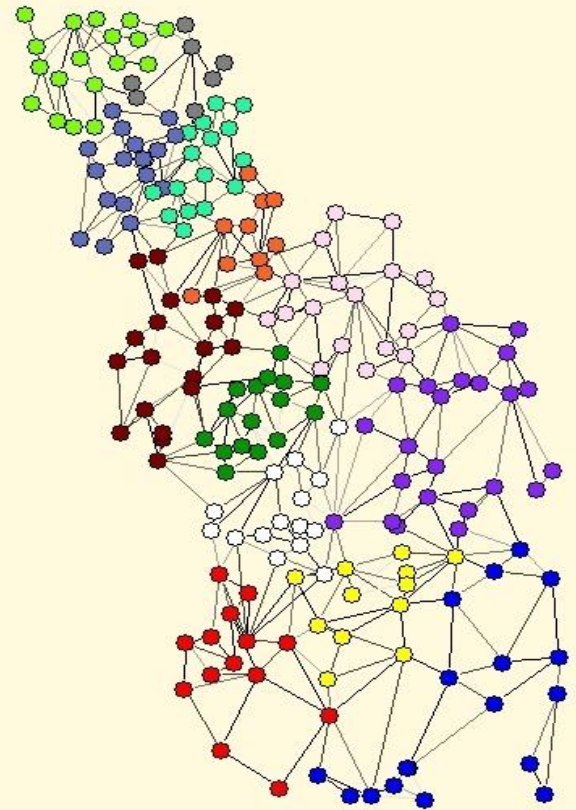


For Example: Planted 2-Partition Model



Methods and Algorithms

- 1) Minimum-cut method
- 2) Hierarchical clustering
- 3) Girvan–Newman algorithm
- 4) Clique-based methods
- 5) Modularity maximization
- 6) Information-based algorithms



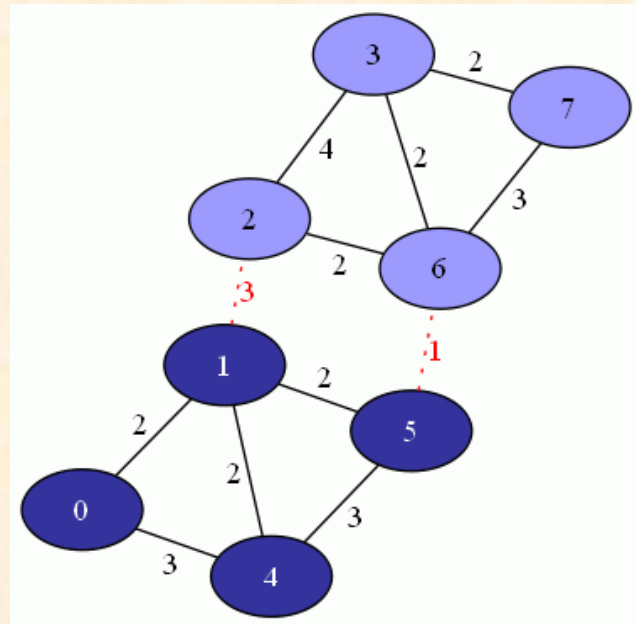
1) Minimum-Cut Method

One of the oldest algorithms for dividing networks into parts

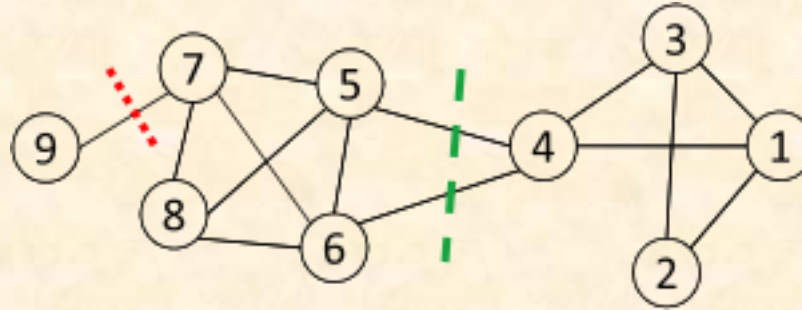
The network is divided into a pre-determined number of groups, usually of approximately the same size (“balanced”), chosen such that the number or cost (total weights) of edges between groups is minimized

Karger's algorithm

Example:



Ratio Cut and Normalized Cut



- ❖ If cost (total weights) of edges between groups are not chosen appropriately, the corresponding **minimum cut** may yield an unbalanced partition, e.g., with one set being a singleton
- ❖ Change the objective function to consider community sizes:

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

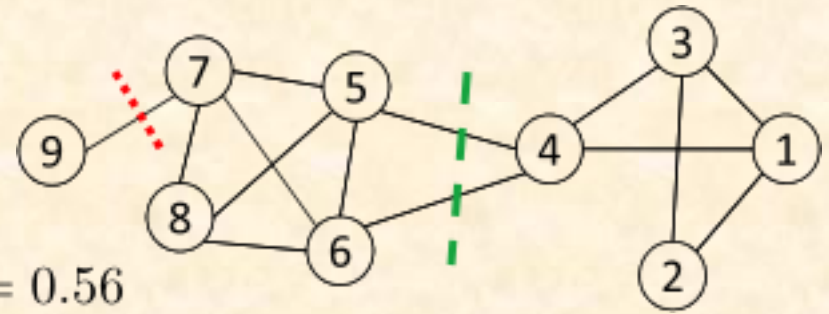
$$\text{Normalized Cut}(\pi) = \frac{1}{k} \sum_{i=1}^k \frac{\text{cut}(C_i, \bar{C}_i)}{\text{vol}(C_i)}$$

C_i : community i

$|C_i|$: number of nodes in C_i

$\text{vol}(C_i)$: sum of degrees in C_i

Example



For partition by 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 by 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 yield a balanced partition

2) Node-Similarity-Based Clustering

- ❖ For practical use with huge networks:
 - Consider the connections as features
 - Use Cosine similarity or Jaccard similarity to compute node similarity
 - Apply the 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
-

Node Similarity

		1	2	3	4	5	6	7	8	9	10	11	12	13
Adjacency matrix	5		1				1							
	8	1					1							1
Structurally equivalent	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: $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$

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

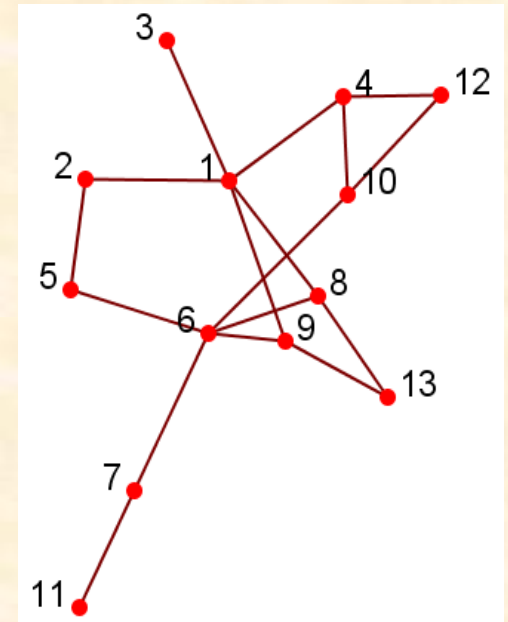
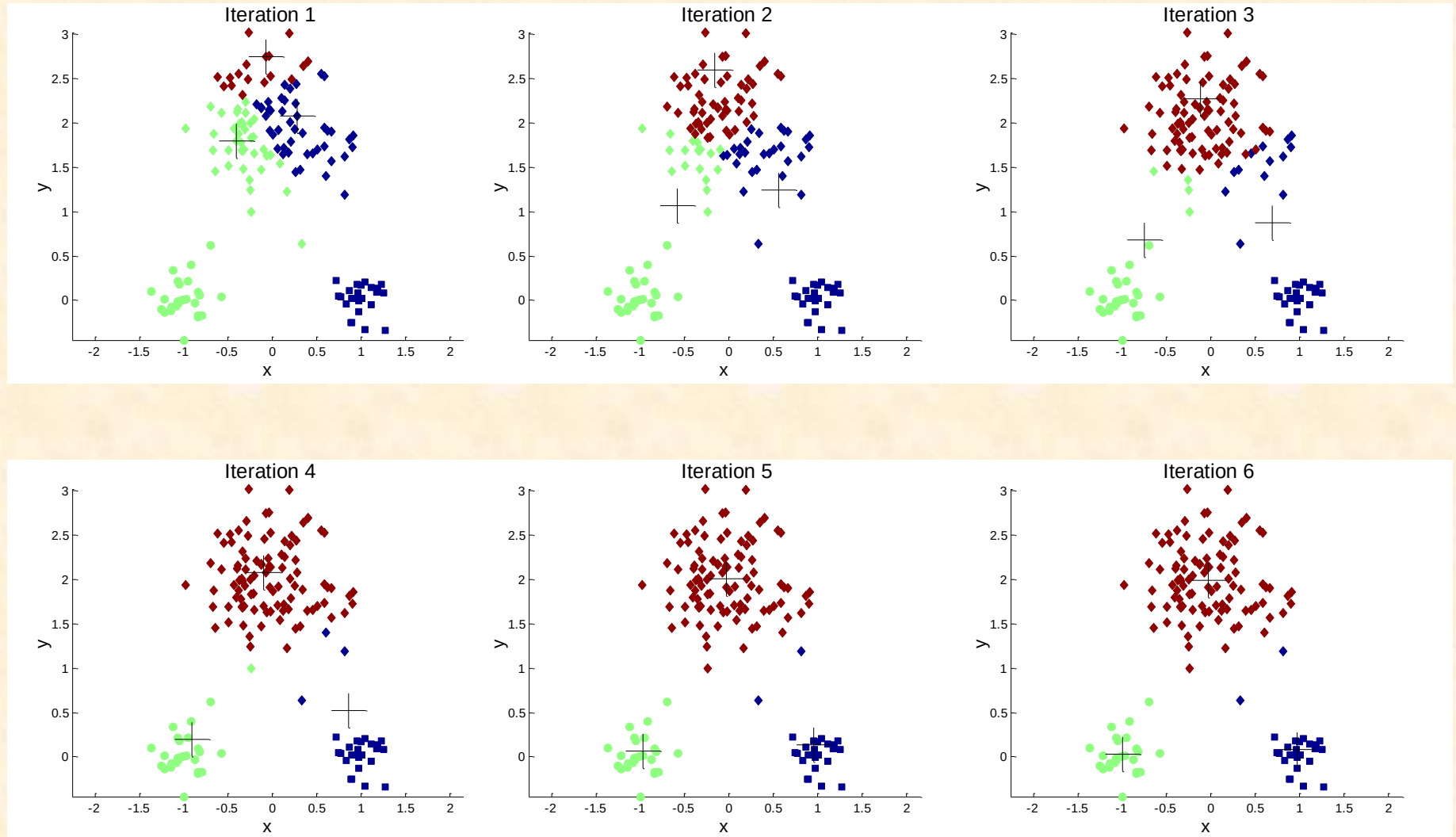


Illustration of K-means Clustering



2) Hierarchical Clustering Method

A **similarity measure** is used to quantify node pairs

Commonly used measures include: cosine similarity, Jaccard similarity, and Hamming distance between rows of the adjacency matrix, etc.

Then, according to this measure, similar nodes are grouped into communities

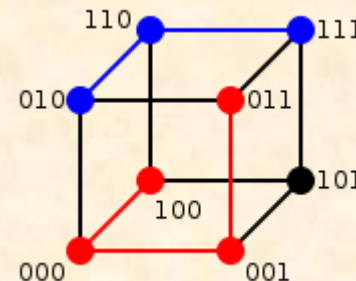
In information theory, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different

Hamming distance between:

toned and **roses** is 3

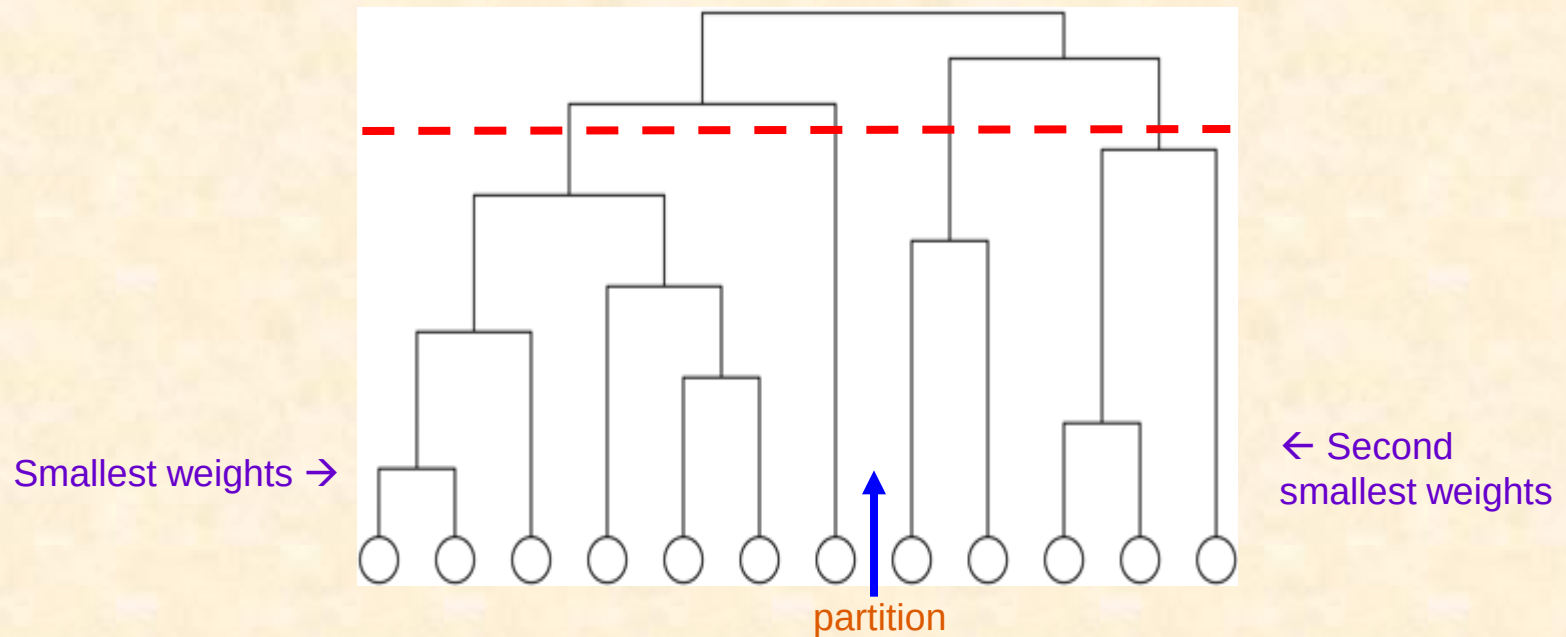
100 → **011** has distance 3 (red path)

010 → **111** has distance 2 (blue path)



Hierarchical Clustering Method

- Calculate a “weight” (e.g., similarity value) for every pair of nodes, which represents how closely connected the nodes are
- Starting with all disconnected nodes, add edges between pairs, one by one, in decreasing order of their weights
- **Result:** Some nested components, where one can take a “slice” at any level of the tree



3) Girvan–Newman Algorithm

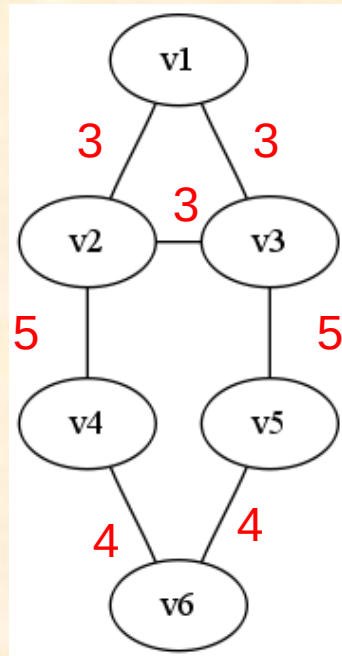
The GN algorithm identifies edges in a network that lie between communities and then removes them, leaving behind just the communities themselves

This is a **betweenness-based clustering** method: Identification is performed by employing the **edge-betweenness**, yielding results of reasonable quality

Drawbacks: High computational complexity

GN Algorithm

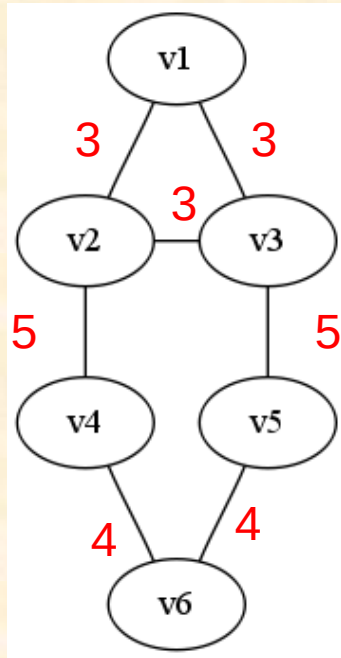
- Calculate all the edge-betweenness in the network
- Remove the edge with the highest betweenness
- Re-calculate all the edge-betweennesses for the resulting (smaller) network
- Repeat the above, until no edge is left



Remove v2-v4, v3-v5

Remove v4-v6, v5-v6

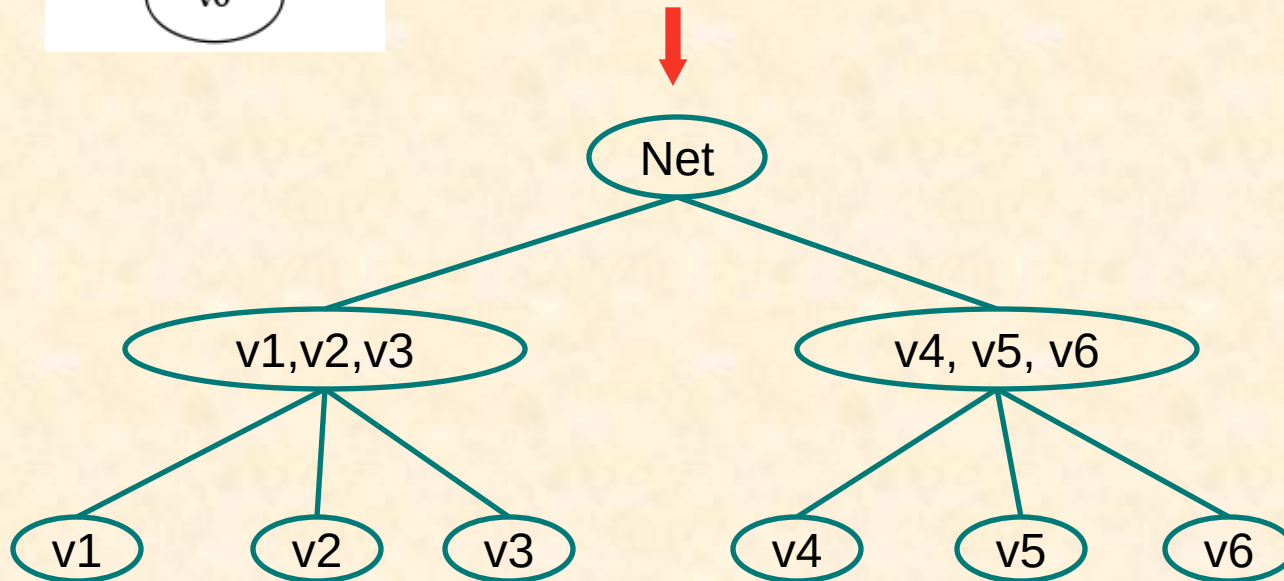
Remove v1-v2, v2-v3, v3-v1



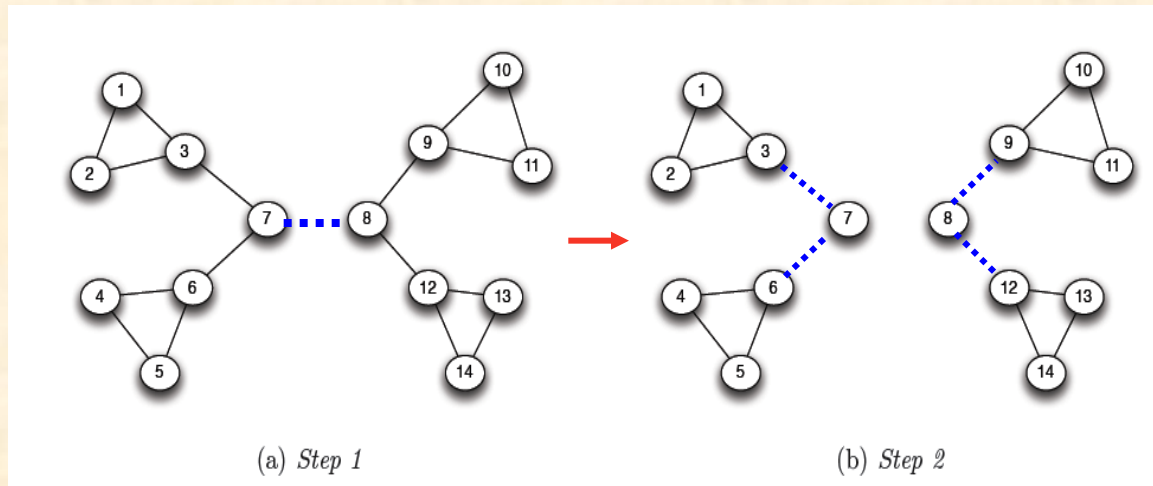
Remove v2-v4, v3-v5

Remove v4-v6, v5-v6

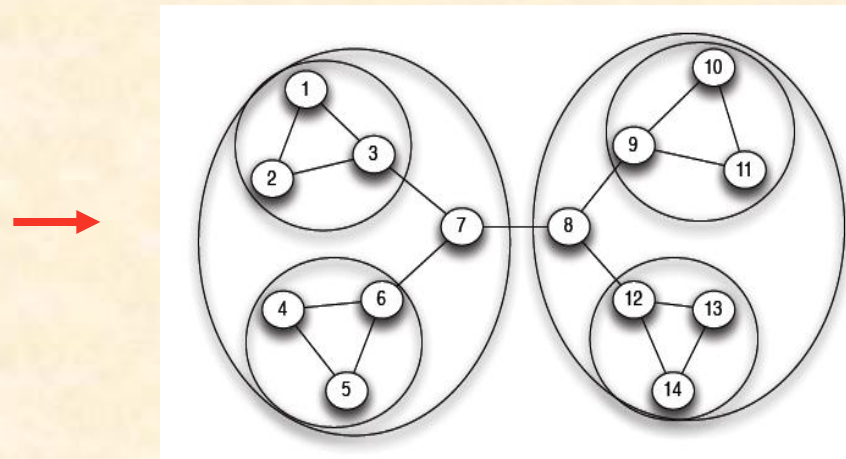
Remove v1-v2, v2-v3, v3-v1



Example:



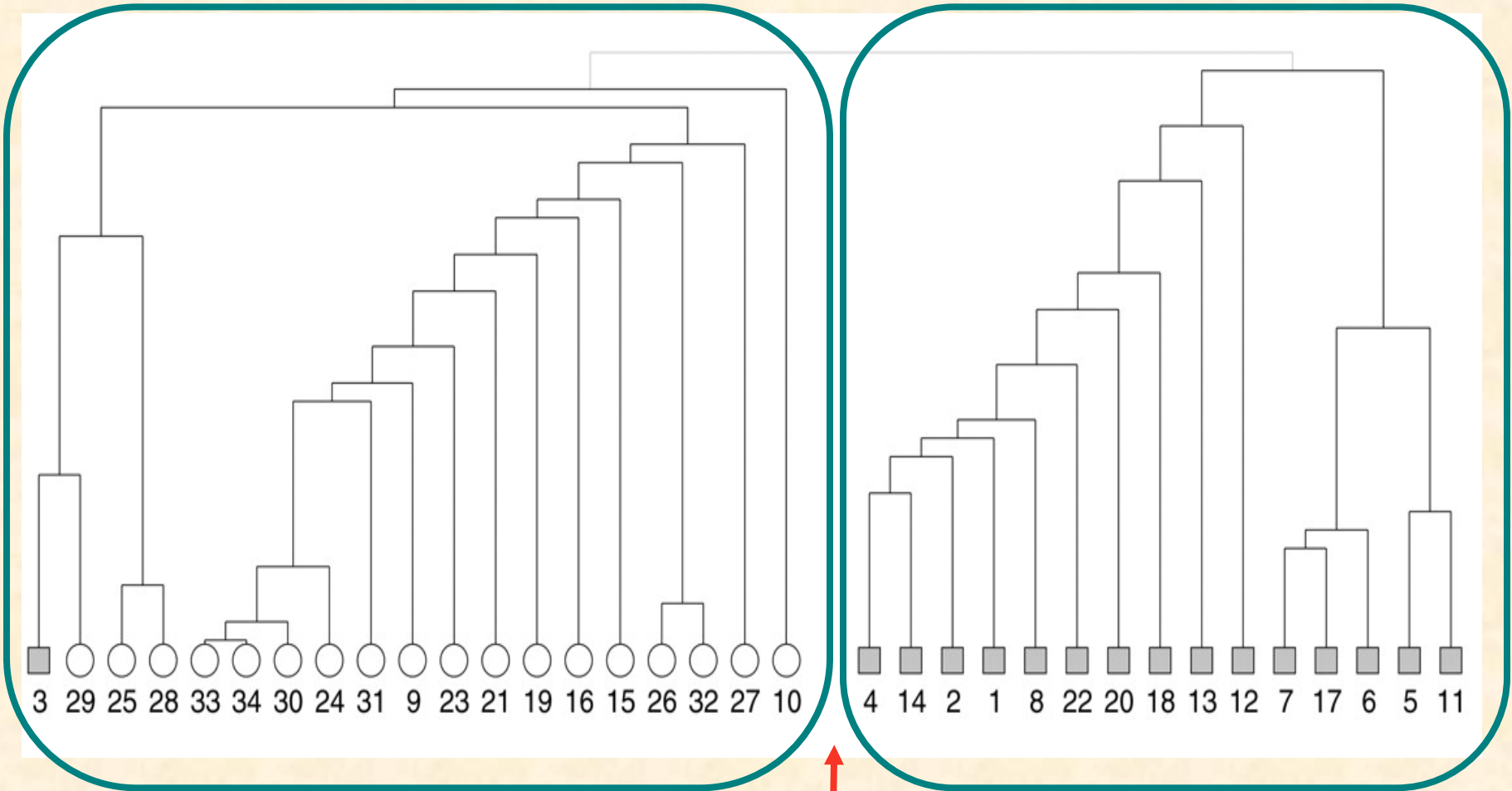
Step 1: remove 7-8 Step 2: remove 3-7, 6-7; 8-9, 8-12



Result

GN Algorithm

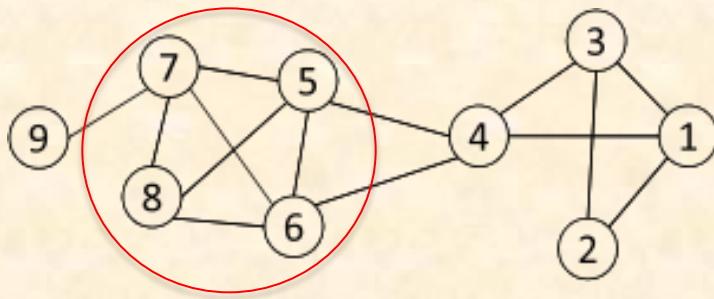
Applied to the Karate Club Dataset



Partition

4) Clique-Based Methods

- A **clique** is a fully-connected subgraph
- There are several **clique-based community detection algorithms** (computationally NP-hard)
- Since a node can be a member of more than one clique, these methods usually yield **overlapping** community structures



For example:

Nodes 5, 6, 7, 8 form a clique

Clique-Based Methods

- One approach is to find all the maximal cliques, which are cliques that are not the subgraph of any other clique
- To find maximum cliques: Bron-Kerbosch algorithm

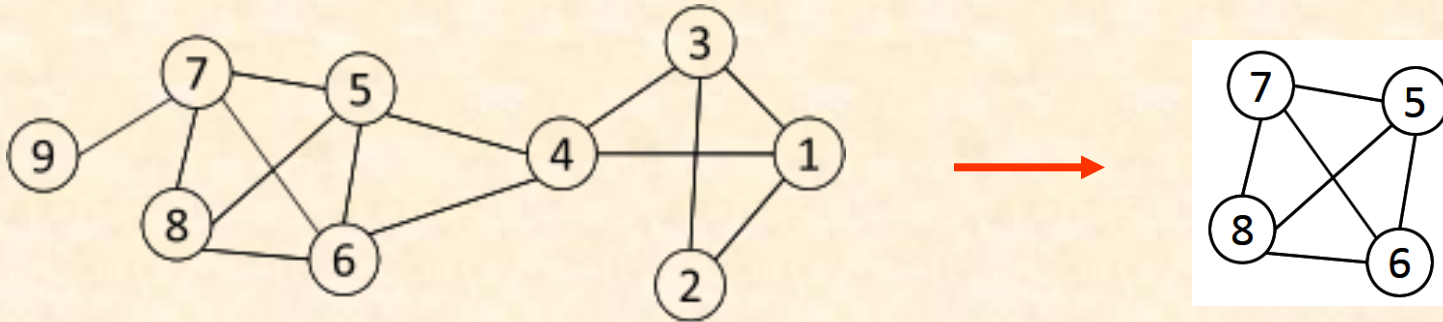
Recursively apply the following pruning procedure:

Sample a subgraph of size k from the given network, and find a clique in it (say, by a greedy algorithm)

To find a larger clique, all nodes with degree $\leq k - 1$ should be removed

Repeat the above, until the network is small enough

Example



- ❖ First, suppose we sampled a sub-network with nodes $\{1, 2, 3, 4, 5\}$ and found a clique $\{1, 2, 3\}$ of size 3
- ❖ Next, to find a clique of size > 3 , remove all nodes with degrees 1 and 2
 - Remove nodes 2, 9
 - Remove nodes 1, 3, 4
- ❖ Result is a clique $\{5, 6, 7, 8\}$, which is bigger

Clique Percolation Method

- ❖ Since a node can be a member of more than one clique, these methods usually yield overlapping community structures
- ❖ CPM is a method to find overlapping communities

Input: A network, and a parameter k

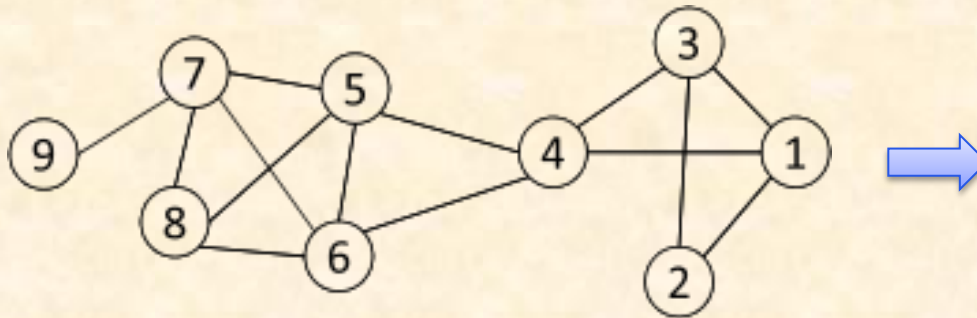
Procedure:

- Find all cliques of size k in the given network
- Construct a clique graph:

Two cliques are adjacent if they share $k - 1$ nodes

- Each connected component in the clique graph is a community

Example



Cliques of size 3:

$\{1, 2, 3\}$, $\{1, 3, 4\}$,
 $\{4, 5, 6\}$, $\{5, 6, 7\}$,
 $\{5, 6, 8\}$, $\{5, 7, 8\}$,
 $\{6, 7, 8\}$

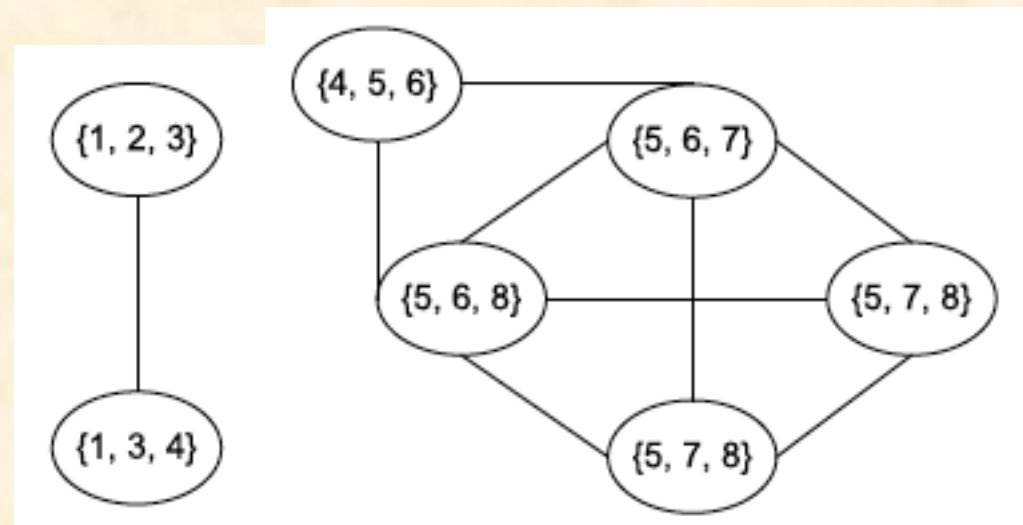
$k = 3$

$k - 1 = 2$ shared nodes

Communities:

$\{1, 2, 3, 4\}$
 $\{4, 5, 6, 7, 8\}$

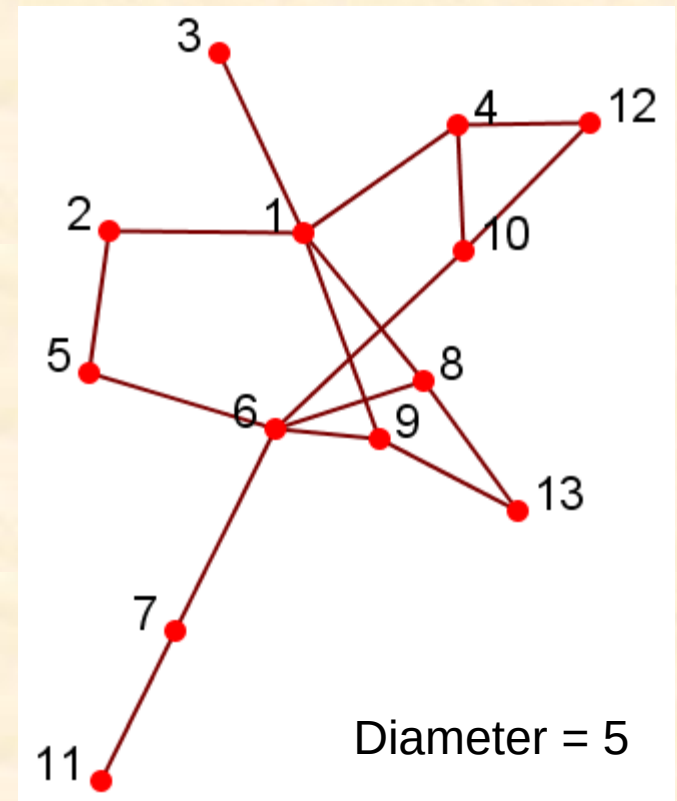
overlapping: 4



Geodesic Between Members

- ❖ Reachability is measured by the geodesic distance (shortest path length)
- ❖ 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: Number of hops in geodesic between two nodes

For example: $d(12, 6) = 2$, $d(3, 11) = 5$
- ❖ Diameter: the maximal geodesic distance for any 2 nodes in a network
= number of hops in longest shortest path



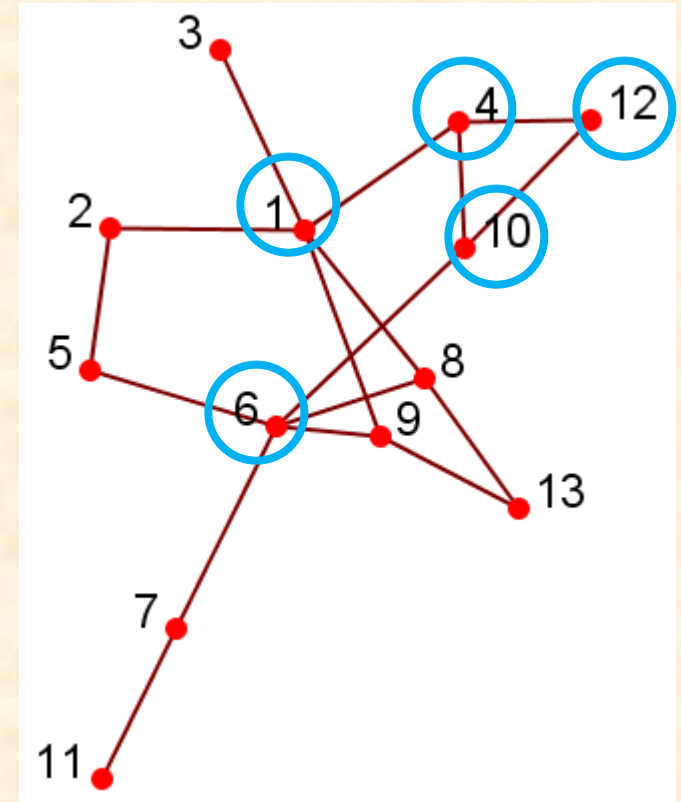
Reachability of Members: k-clique, k-club

- ❖ 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 $\leq k$
- ❖ Note: **k-clique** can have diameter larger than k within the subgraph

For example: **2-clique** {12, 4, 10, 1, 6}
within the whole network: $d(1, 6) = 2$
within this subgraph: $d(1, 6) = 3$

- ❖ **k-club**: A subgraph of diameter $\leq k$

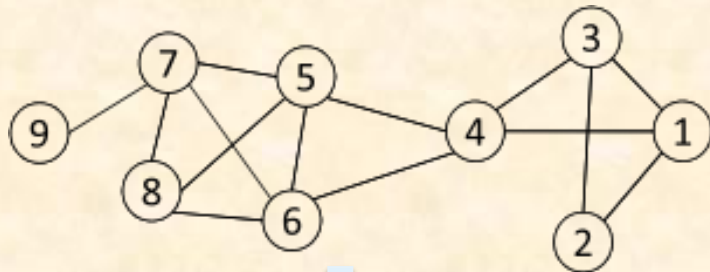
For example: {1, 2, 5, 6, 8, 9}, {12, 4, 10, 1} are **2-clubs**



Latent Space Models

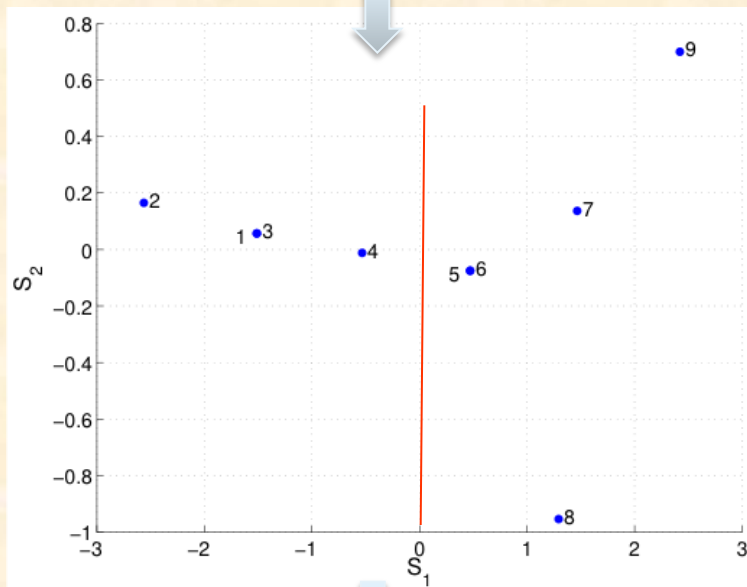
- ❖ Map nodes into a low-dimensional space, such that the proximity (i.e., closeness) between nodes based on network connectivity is preserved in the new space. Then, apply the *k-means clustering* algorithm
- ❖ Multi-Dimensional Scaling (MDS)
 - For a given network, construct a proximity matrix P representing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

MDS Example



geodesic
distances

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



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

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

Two communities:

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

5) Modularity Maximization

- It is one of the most widely used methods
- Modularity is a **benefit function** that measures the quality of a particular division of a network into communities
- The modularity maximization method detects communities by searching over possible divisions of a network for those that have particularly **high modularity**
- A popular modularity maximization approach is the **Louvain method** which iteratively optimizes local communities until global modularity can not be further improved

Modularity Maximization

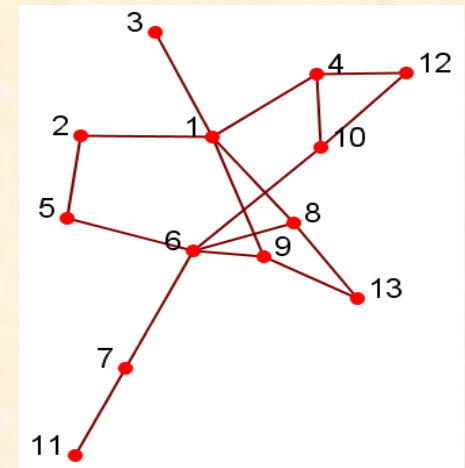
- ❖ **Modularity** measures the group interactions compared with the expected number of random connections in the group
- ❖ In a network with m edges, for two nodes with degree d_i and d_j , expected number of random connections between them are $d_i d_j / 2m$

- ❖ The interaction utility in a group, C , is

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

- ❖ To partition the group into multiple groups, the objective is to maximize

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



Expected number of edges between 6 and 9:
 $5 \times 3 / (2 \times 17) = 15/34$

Information-based algorithms

They use ideas from information theory to identify communities, for example to find an optimal compression of the graph topology

In a network of N nodes, let the number of nodes in community k be n_k . Then, the probability of a node falls into community k is $P(k) = n_k / N$ and the Shannon entropy of partition A , which contains Q_A communities, is

$$H(A) = -\sum_{i=1}^{Q_A} P(i) \log P(i) = -\sum_{i=1}^{Q_A} \frac{n_i}{N} \log \frac{n_i}{N}$$

P. Ronhovde and Z. Nussinov, Physical Review E 80, 016109 (2009)

Mutual information $I(A,B)$ between two partitions A and B is used to evaluate how similar two data sets are, which is defined as

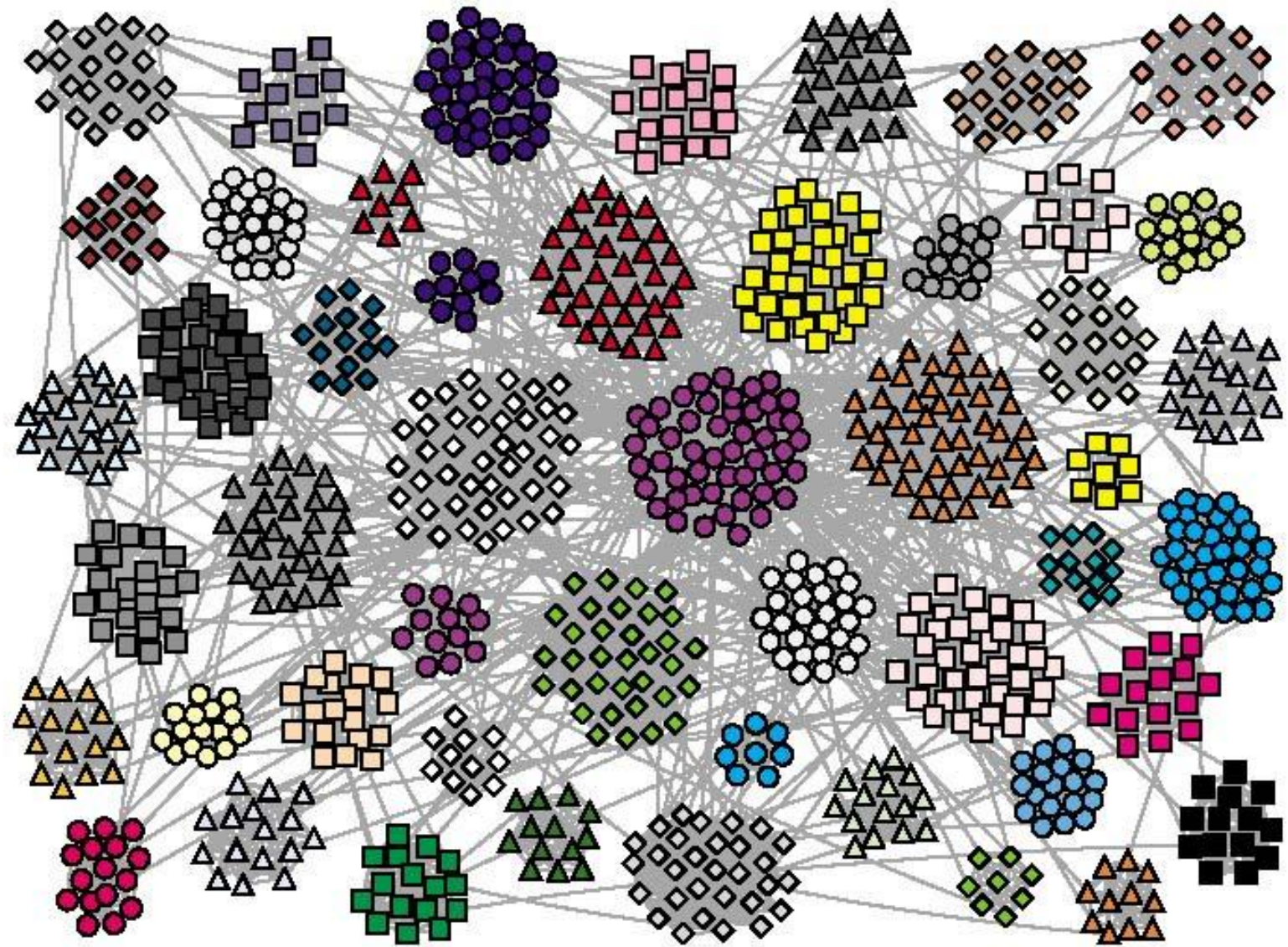
$$I(A,B) = \sum_{i=1}^{Q_A} \sum_{j=1}^{Q_B} \frac{n_{ij}}{N} \log \left(\frac{n_{ij}}{n_i n_j} N \right)$$

where n_i is the number of nodes in community i of partition A and n_{ij} is the number of nodes in community i which also belong to community j of partition B .

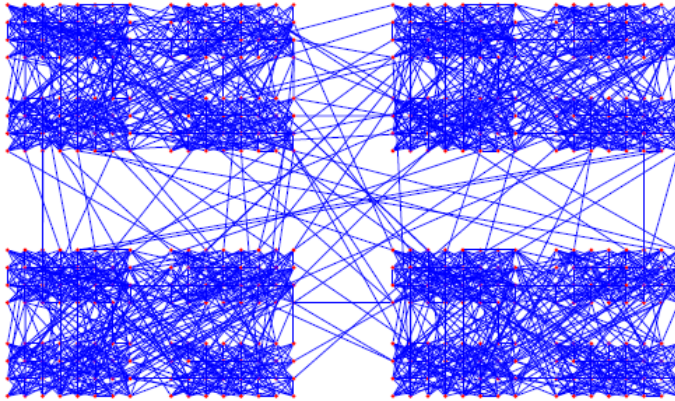
The variation $V(A,B)$ is a measure of the “distance” between partition A and partition B , defined by

$$V(A,B) = H(A) + H(B) - 2I(A,B)$$

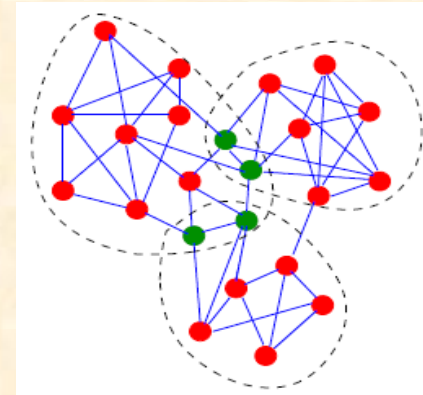
A smaller value of $V(A,B)$ means a better agreement between the two partitions.



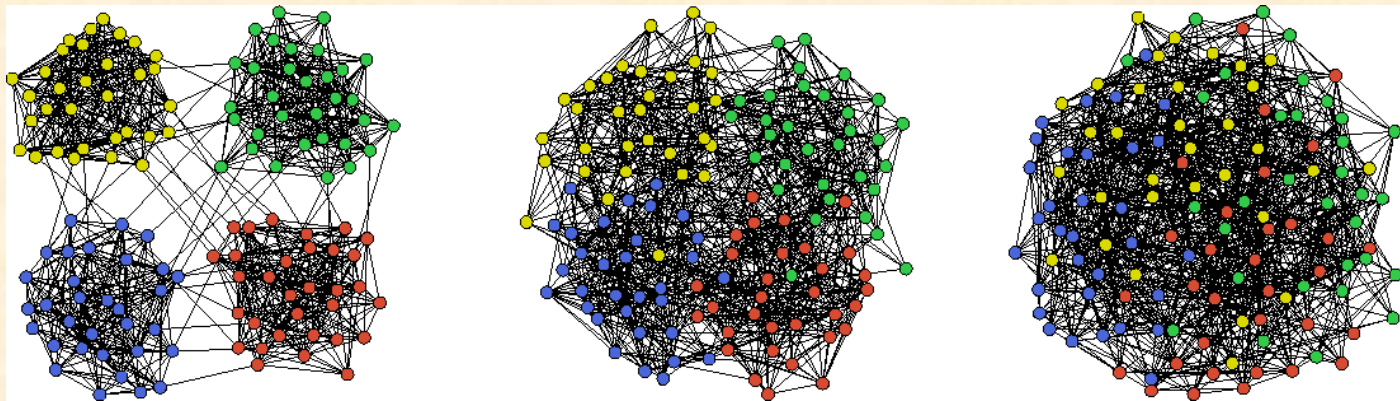
Detecting Community Structure: Challenges



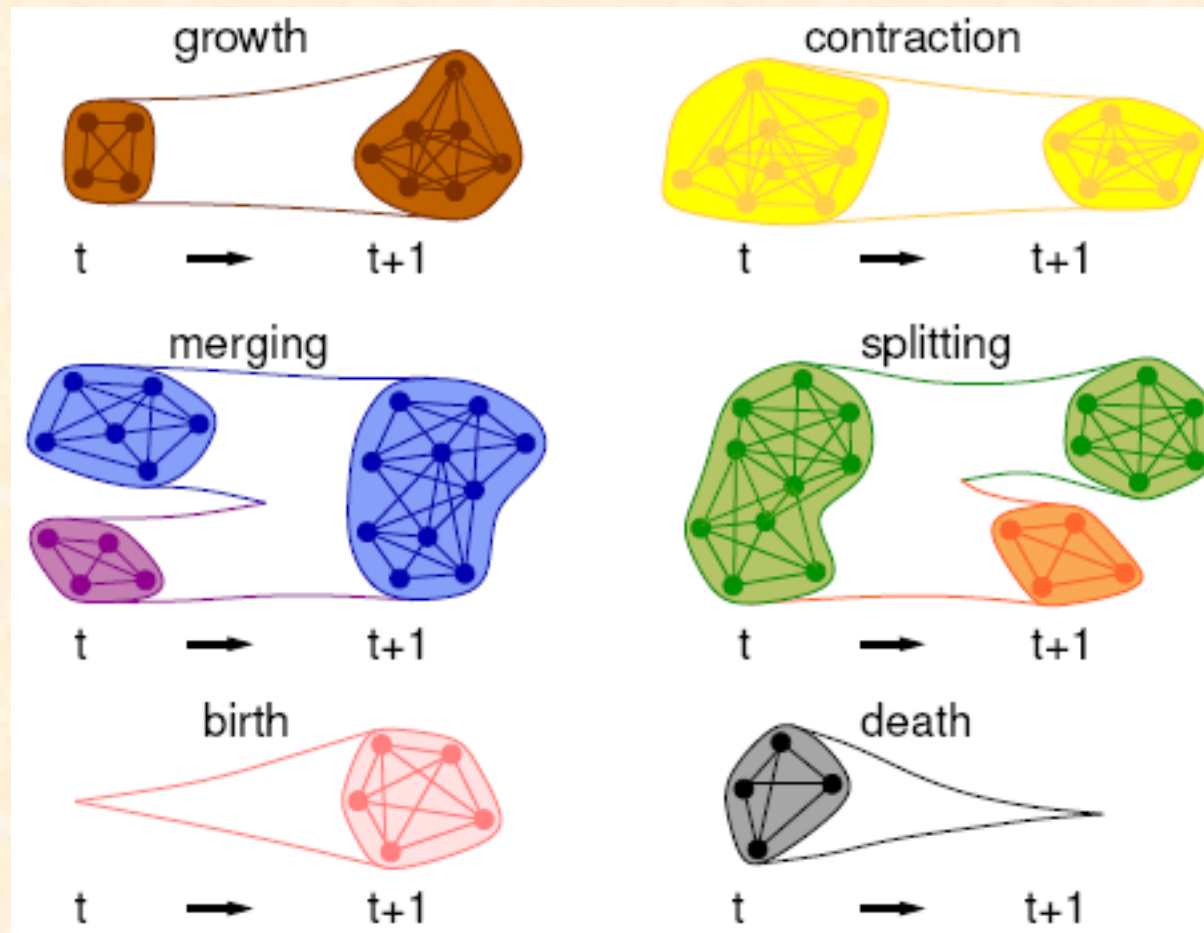
■ Hierarchical



■ Overlapping



Detecting Community Structure: Challenges



■ Evolution, Emergence

End

