CS 579: Online Social Network Analysis

Community Analysis

Kai Shu

Spring 2023

Read Chapter 6

Social Community

[real-world] community

A group of individuals with common *economic*, *social*, or *political* interests or characteristics, often living in *relative proximity*.

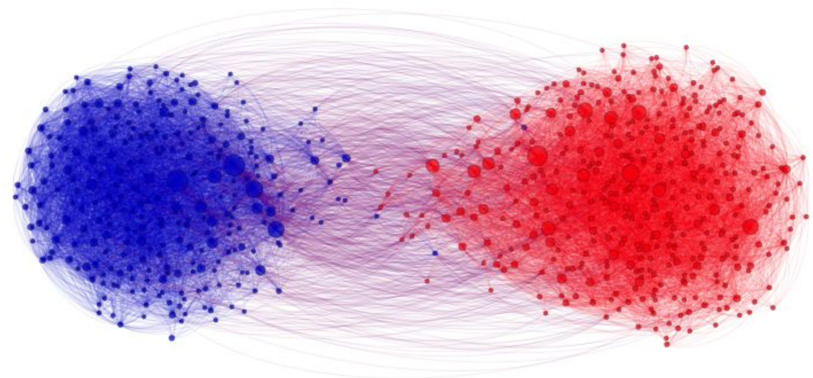
Why analyze communities?

Analyzing communities helps better understand users

– Users form groups based on their interests

Groups provide a clear global view of user interactions

- E.g., find polarization



Some behaviors are only observable in a group setting and not on an individual level

- Some republican can agree with some democrats, but their parties can disagree

Social Media Communities

- **Formation:**
  - When like-minded users on social media form a link and start interacting with each other
- **More Formal Formation**:
  1. A set of at least two nodes sharing some interest, and
  2. Interactions with respect to that interest.
- Social Media Communities
  - **Explicit**: formed by user subscriptions
  - **Implicit**: implicitly formed by social interactions
    - **Example:** individuals calling Canada from the United States
    - Phone operator considers them one community for promotional offers
- Other community names: *group*, *cluster*, *cohesive subgroup*, or *module*

Examples of Explicit Social Media Communities

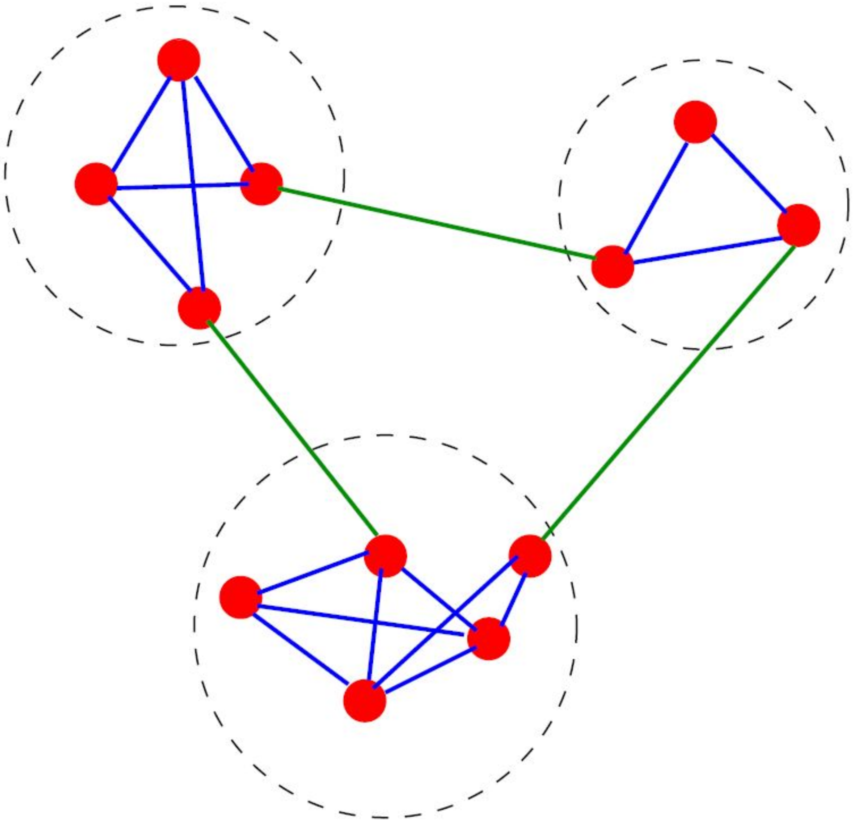
Facebook has groups and communities. Users can
- post messages and images
- can comment on other messages
- can like posts
- can view activities of other users

In Google+, Circles represent communities

In Twitter, communities form as lists
- Users join lists to receive information in the form of tweets

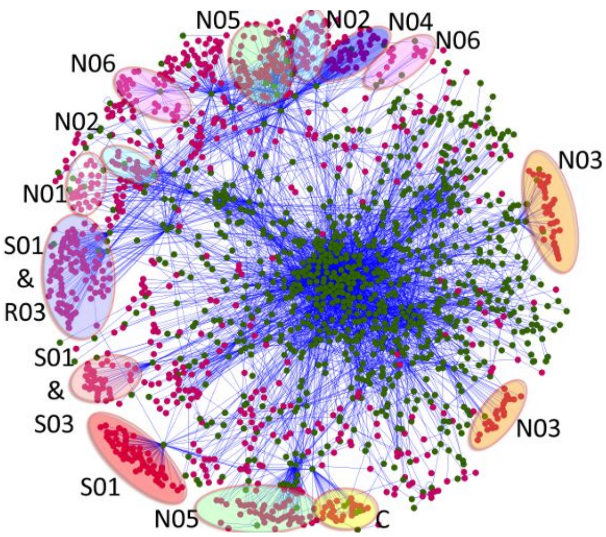
LinkedIn provides Groups and Associations
- Users can join professional groups where they can post and share information related to the group

Finding Implicit Communities: An Example



A simple graph in which **three** implicit communities are found, enclosed by the dashed circles

Implicit communities in other domains



Protein-protein interaction networks

– Communities are likely to group proteins having the same specific function within the cell

World Wide Web

– Communities may correspond to groups of pages dealing with the same or related topics

Metabolic networks

– Communities may be related to functional modules such as cycles and pathways

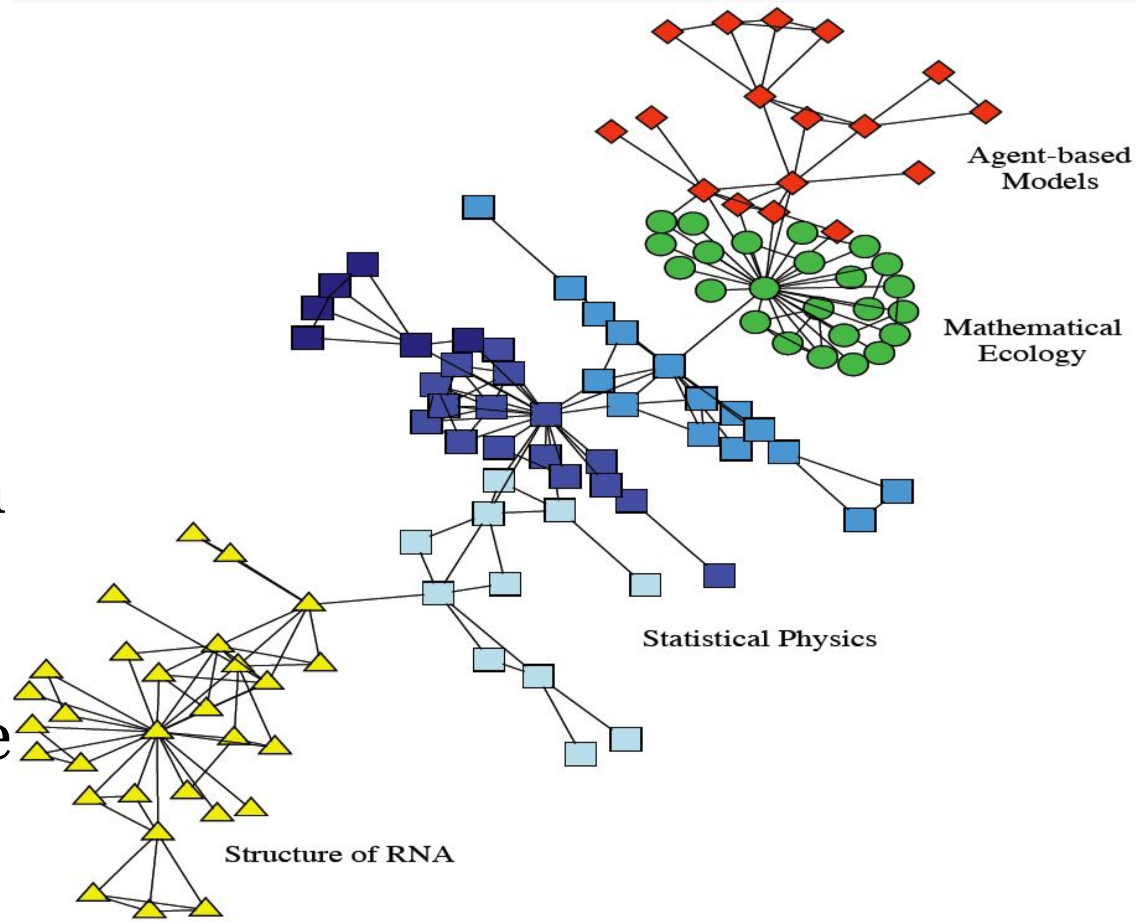
Food webs

– Communities may identify compartments

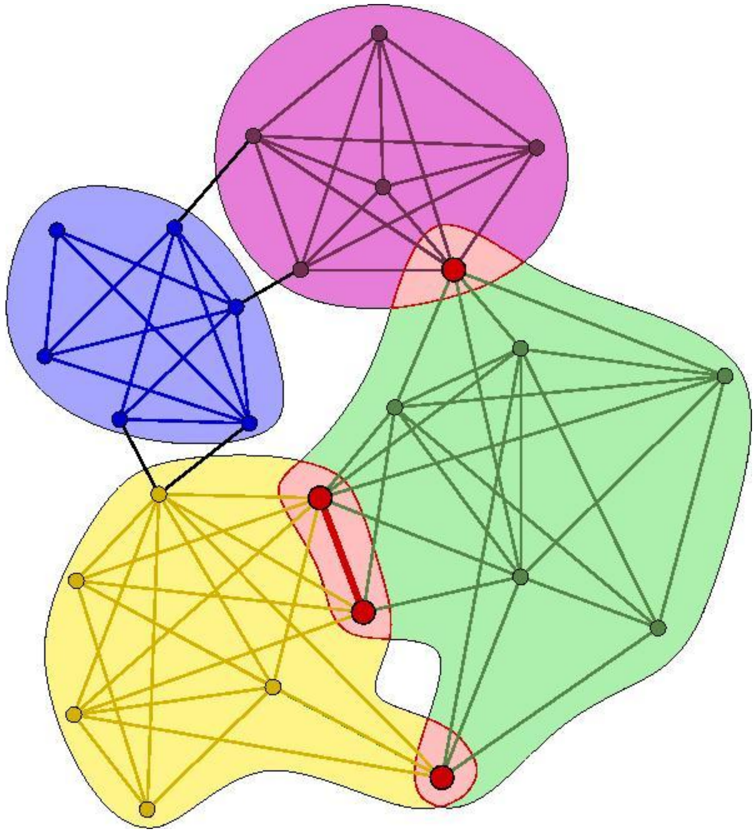
Real-world Implicit Communities

Collaboration network between scientists working at the Santa Fe Institute

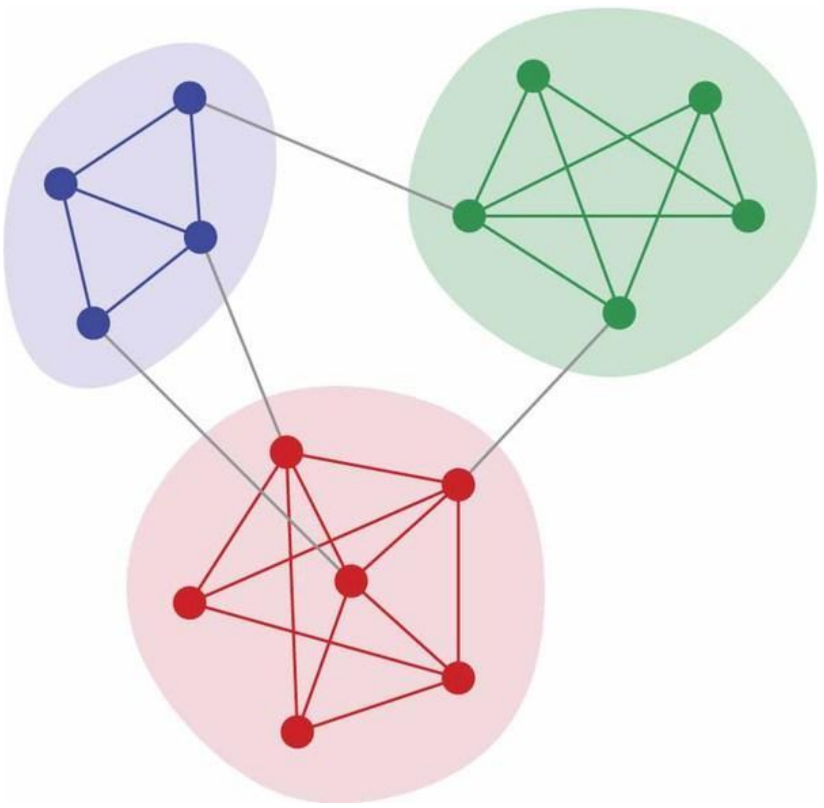
The colors indicate high level communities and correspond to research divisions of the institute



Overlapping vs. Disjoint Communities



Overlapping Communities



Disjoint Communities

What is Community Analysis?

- **Community detection**
  - Discovering implicit communities
- **Community evolution**
  - Studying temporal evolution of communities
- **Community evaluation**
  - Evaluating detected communities

Community Detection

What is community detection?

- The process of finding clusters of nodes ("*communities*")
  - With **Strong** internal connections and
  - **Weak** connections between different communities

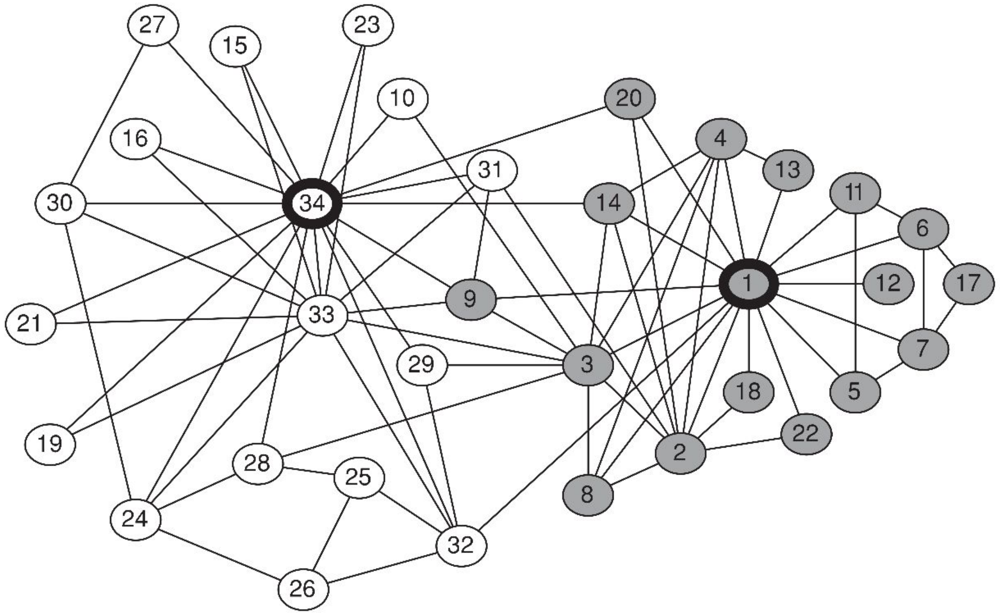
- Ideal decomposition of a large graph
  - Completely disjoint communities
  - There are no interactions between different communities

- In practice,
  - find community partitions that are maximally decoupled

Why Detecting Communities is Important?

Zachary's karate club

Interactions between 34 members of a karate club for over two years



- The club members split into two groups (gray and white)
- Disagreement between the administrator of the club (node 34) and the club's instructor (node 1),
- The members of one group left to start their own club

The same communities can be found using community detection

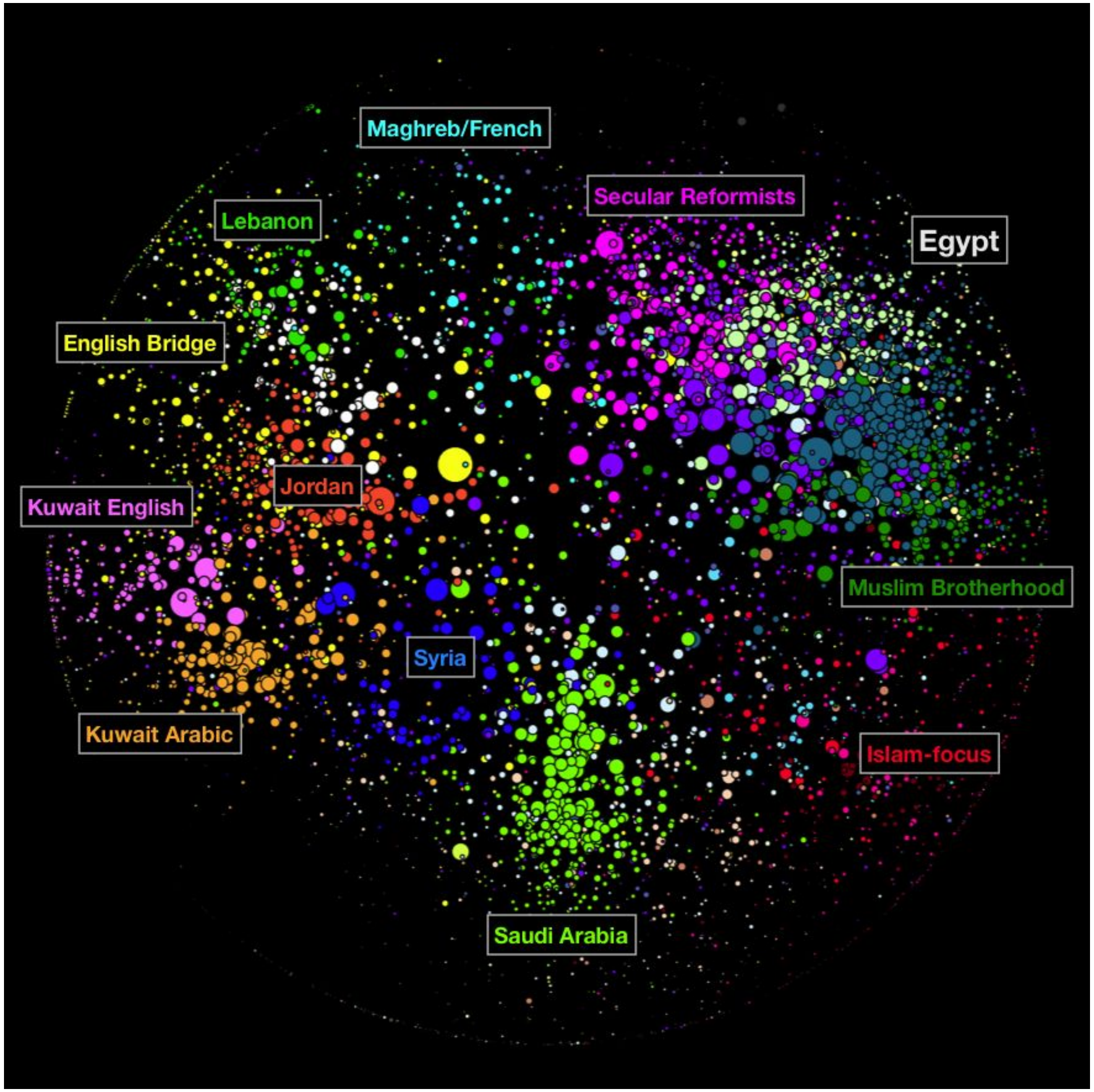
Why Community Detection?

Network Summarization

- A community can be considered as a summary of the whole network
- Easier to visualize and understand

Preserve Privacy

- [<u>Sometimes</u>] a community can reveal some properties without releasing the individuals' privacy information.



Community Detection vs. Clustering

Clustering

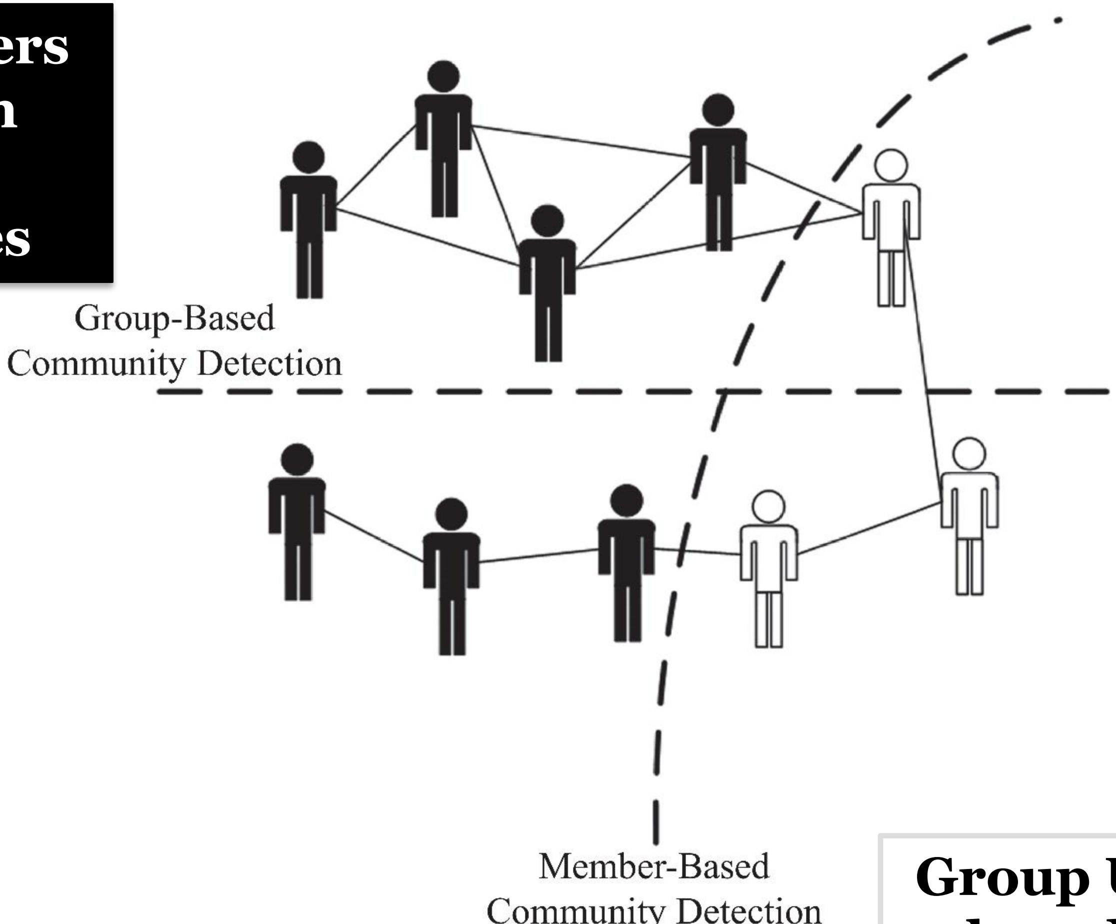
- Data is often non-linked (matrix rows)
- Clustering works on the distance or similarity matrix, e.g., k-means.
- If you use k-means with adjacency matrix rows, you are only considering the ego-centric network

Community detection

- Data is linked (a graph)
- Network data tends to be “discrete”, leading to algorithms using the graph property directly
  - k-clique, quasi-clique, or edge-betweenness

Community Detection Algorithms

Group Users based on Group attributes



Group Users based on Member attributes

Member-Based Community Detection

Member-Based Community Detection

- Look at node characteristics; and
- Identify nodes with similar characteristics and consider them a community

Node Characteristics

A. Degree
- Nodes with same (or similar) degrees are in one community
- Example: cliques

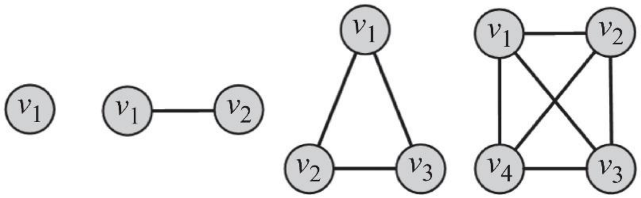
B. Reachability
- Nodes that are close (small shortest paths) are in one community
- Example: k-cliques, k-clubs, and k-clans

C. Similarity
- Similar nodes are in the same community

A. Node Degree

Most common subgraph searched for:

- **Clique**: a maximum complete subgraph in which all nodes inside the subgraph adjacent to each other



Find communities by searching for

1. **The maximum clique**: the one with the largest number of vertices, or
2. **All maximal cliques**: cliques that are not subgraphs of a larger clique; i.e., cannot be further expanded

To overcome this, we can

I. Brute Force
II. Relax cliques
III. Use cliques as the core for larger communities

Both problems are NP-hard

I. Brute-Force Method

Can find all the maximal cliques in the graph

For each vertex v_x, we find the maximal clique that contains node v_x

Algorithm 1 Brute-Force Clique Identification

Require: Adjacency Matrix A, Vertex v_x

return Maximal Clique C containing v_x
CliqueStack = $\{\{v_x\}\}$, Processed = $\{\}$;
while CliqueStack not empty **do**
  C=pop(CliqueStack); push(Processed,C);
  v_{last} = Last node added to C;
  $N(v_{last}) = \{v_i | A_{v_{last}, v_i} = 1\}$.
  for all $v_{temp} \in N(v_{last})$ **do**
    if $C \bigcup \{v_{temp}\}$ is a clique **then**
      push(CliqueStack, $C \bigcup \{v_{temp}\}$);
    end if
  end for
end while
Return the largest clique from Processed

Impractical for large networks:

- For a complete graph of only 100 nodes, the algorithm will generate at least $2^{99} - 1$ different cliques starting from any node in the graph

Enhancing the Brute-Force Performance

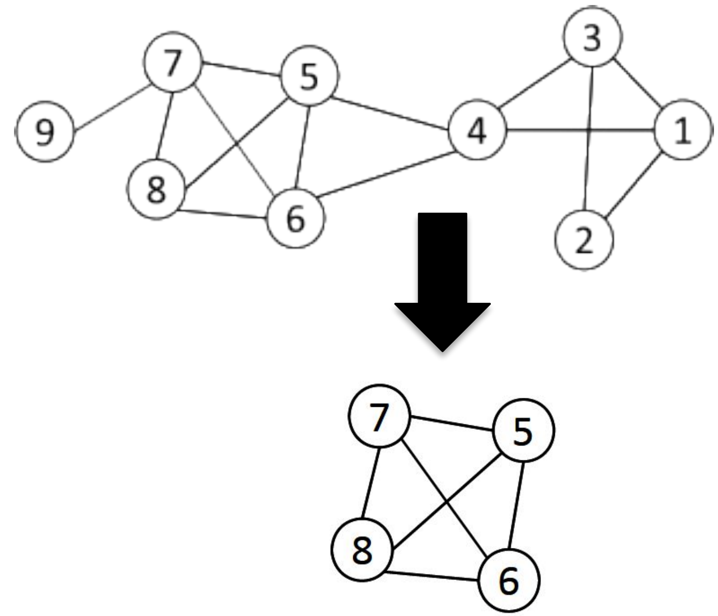
[Systematic] Pruning can help:

- When searching for cliques of size k or larger
- If the clique is found, each node should have a degree equal to or more than $k-1$
- We can first prune all nodes (and edges connected to them) with degrees less than $k-1$
  - More nodes will have degrees less than $k-1$
  - Prune them recursively
- For large k, many nodes are pruned as social media networks follow a power-law degree distribution

Maximum Clique: Pruning…

Example. to find a clique ≥ 4, remove all nodes with degree $\leq (4-1)-1=2$

- Remove nodes 2 and 9
- Remove nodes 1 and 3
- Remove node 4



Even with pruning, cliques are less desirable

- Cliques are rare
- A clique of 1000 nodes, has 999x1000/2 edges
- A single edge removal destroys the clique
- That is less than 0.0002% of the edges!

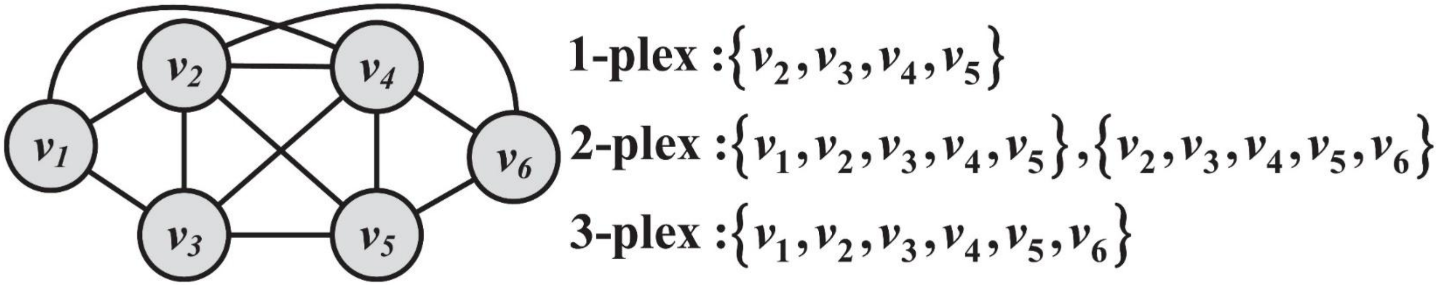
II. Relaxing Cliques

- **k-plex**: a set of vertices V in which we have

$$d_v \geq |V| - k, \forall v \in V$$

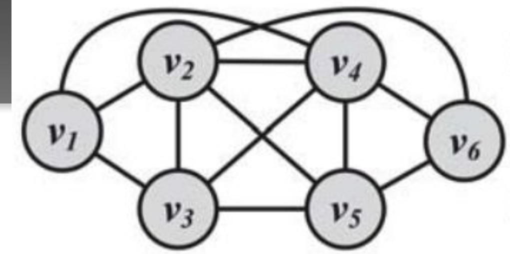
- d_v is the degree of v in the induced subgraph
  - Number of nodes from V that are connected to v

- Clique of size k is a 1-plex
- Finding the maximum k-plex: **NP-hard**
  - In practice, relatively easier due to smaller search space.



Maximal k-plexes

K-plex Continued



- Compute All the 1-plex
    a. One node: {v1}, {v2},…,{v6}
    b. Two nodes: {v1,v2}, {v1,v3},…,{v5,v6}
    c. Three nodes: {v1,v2,v3}, {v2,v3,v4},…,{v3,v4,v5}
    d. Four nodes: **{v1,v2,v3,v4}**, **{v2,v3,v4,v5}**, **{v2,v4,v5,v6}**
    e. Five nodes: None
- 2-plex
    a. all the 1-plex subgraphs with at least two nodes are 2-plex
    b. Five nodes: **{v1,v2,v3,v4,v5}**, **{v2,v3,v4,v5,v6}**
    c. Six nodes: None
- 3-plex
    a. all the 2-plex subgraphs with at least three nodes are 3-plex
    b. Six nodes: **{v1,v2,v3,v4,v5,v6}**

The bold k-plex are the maximum k-plexes

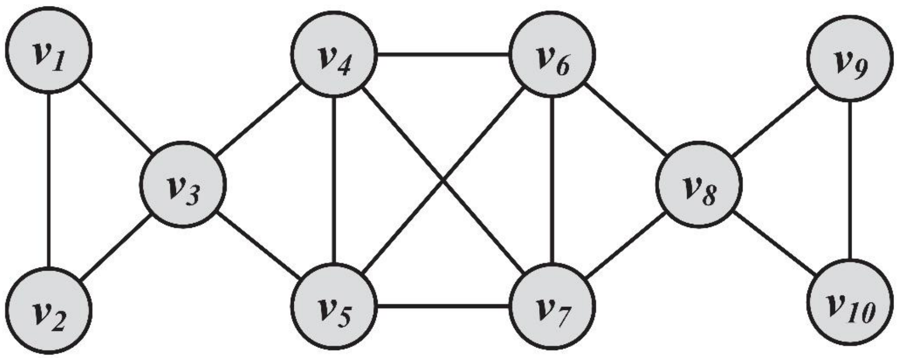
III. Using Cliques as a Seed of a Community

Clique Percolation Method (CPM)

- Uses cliques as seeds to find larger communities
- CPM finds overlapping communities

- **Input**
  - A parameter k, and a network
- Procedure
  - Find out 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 components in the clique graph form a community

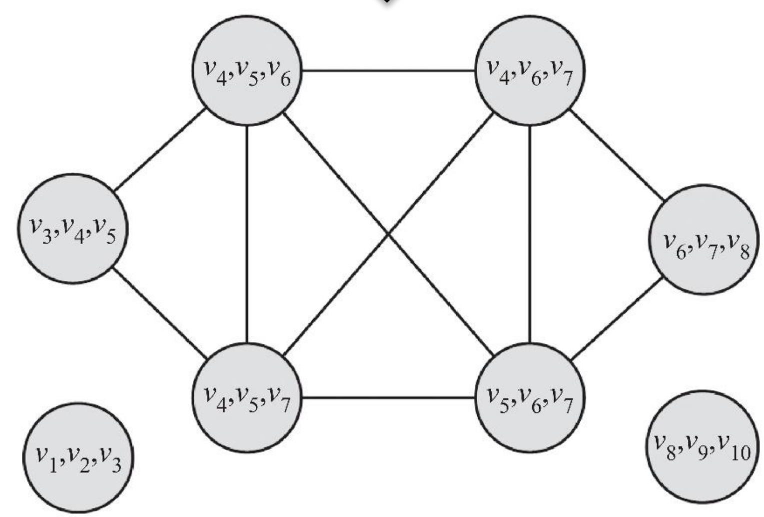
Clique Percolation Method: Example



(a) Graph

Cliques of size 3:

$\{v_1, v_2, v_3\}$, $\{v_3, v_4, v_5\}$,
$\{v_4, v_5, v_6\}$, $\{v_4, v_5, v_7\}$,
$\{v_4, v_6, v_7\}$, $\{v_5, v_6, v_7\}$,
$\{v_6, v_7, v_8\}$, $\{v_8, v_9, v_{10}\}$



(b) CPM Clique Graph

Communities:

$\{v_1, v_2, v_3\}$,
$\{v_8, v_9, v_{10}\}$,
$\{v_3, v_4, v_5, v_6, v_7, v_8\}$