Social Media Mining

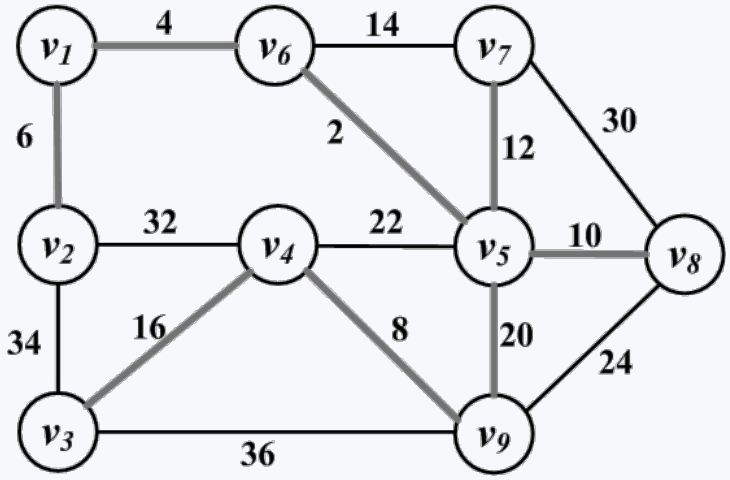
Graph Essentials

Outline

- Graph basics
- Graph representation
- Types of graphs
- Connectivity in graphs
- Special graphs

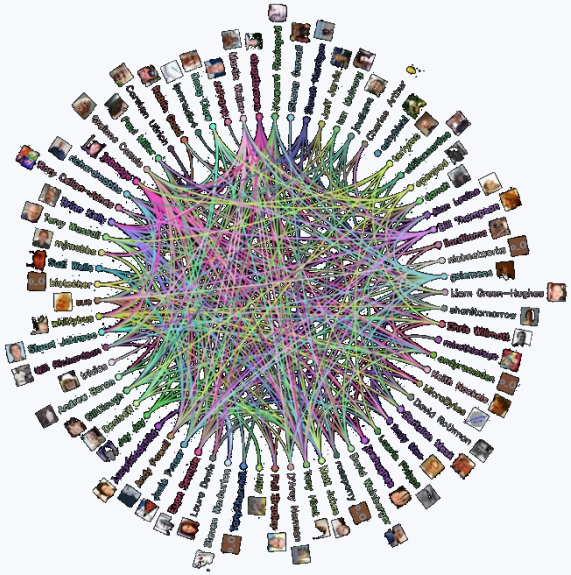
Networks <-> Graphs

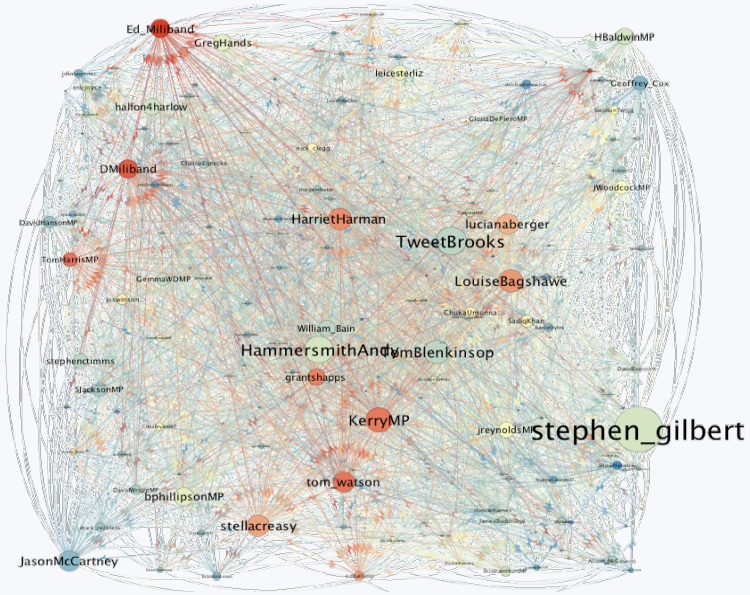
- A network is a graph.
  - Elements of the network have meanings
- Network problems can usually be represented in terms of graph problems
- Twitter example:
  - Given a piece of information, a network of individuals, and the cost to propagate information among any connected pair, find the minimum cost to disseminate the information to all individuals.



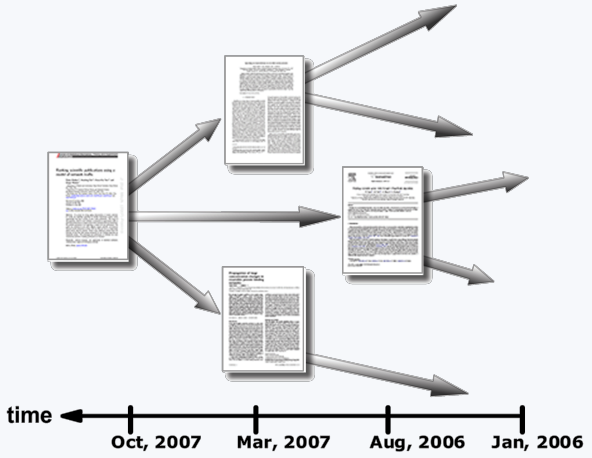
Networks are Pervasive

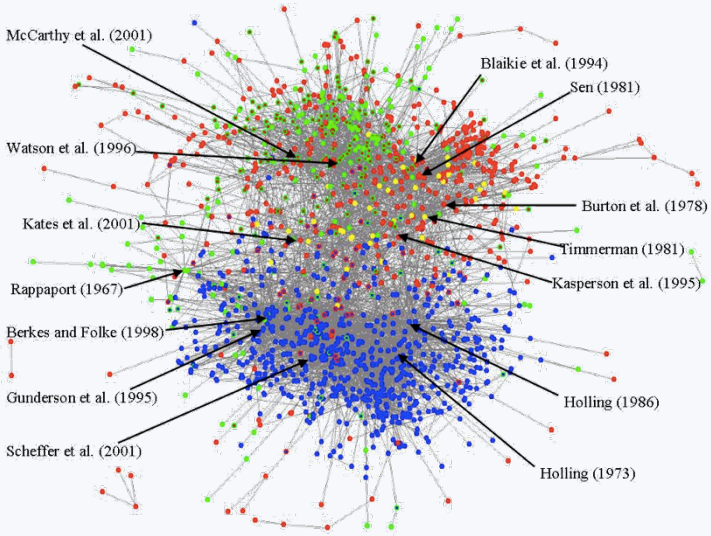
Twitter Networks



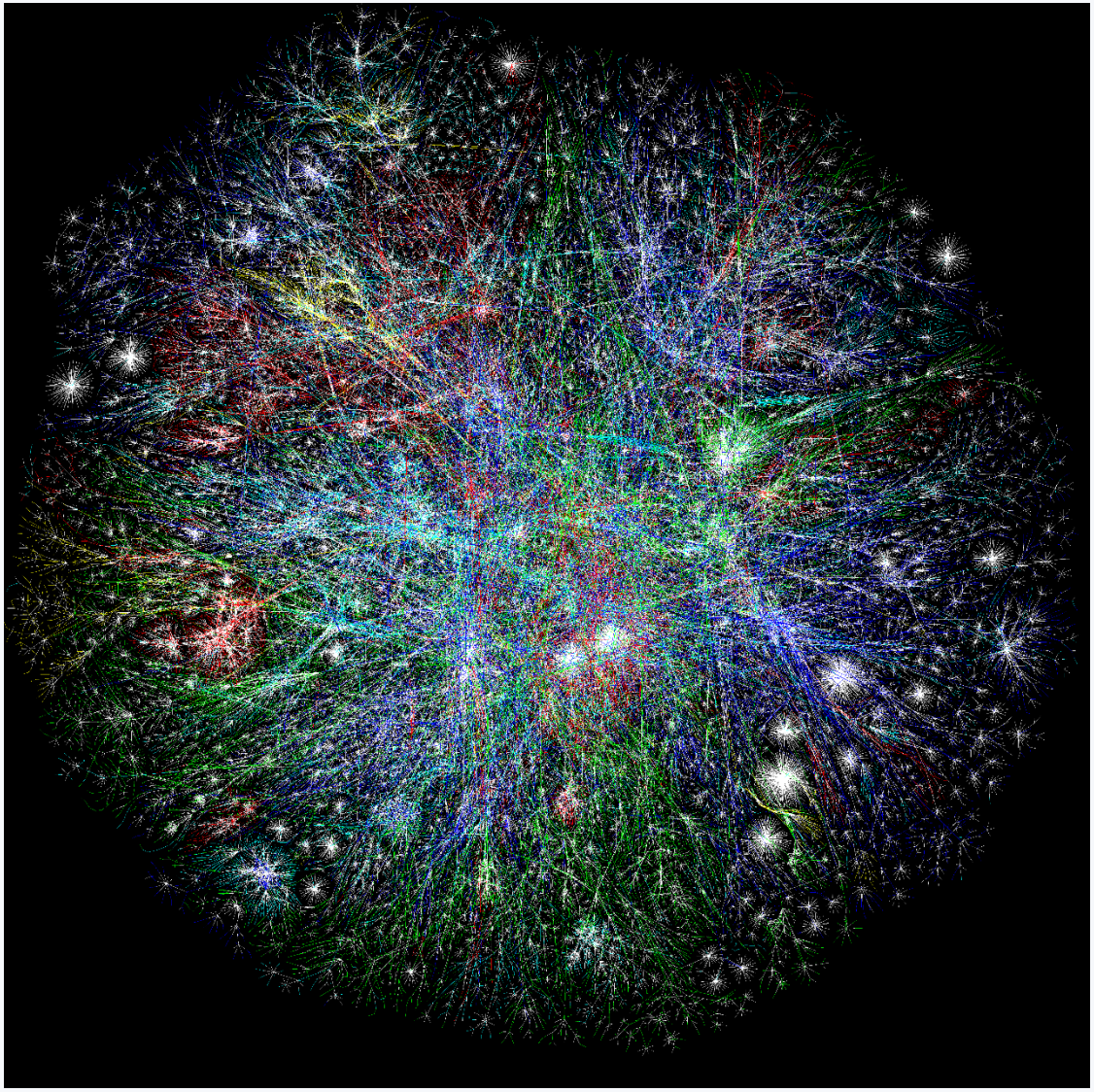


Citation Networks





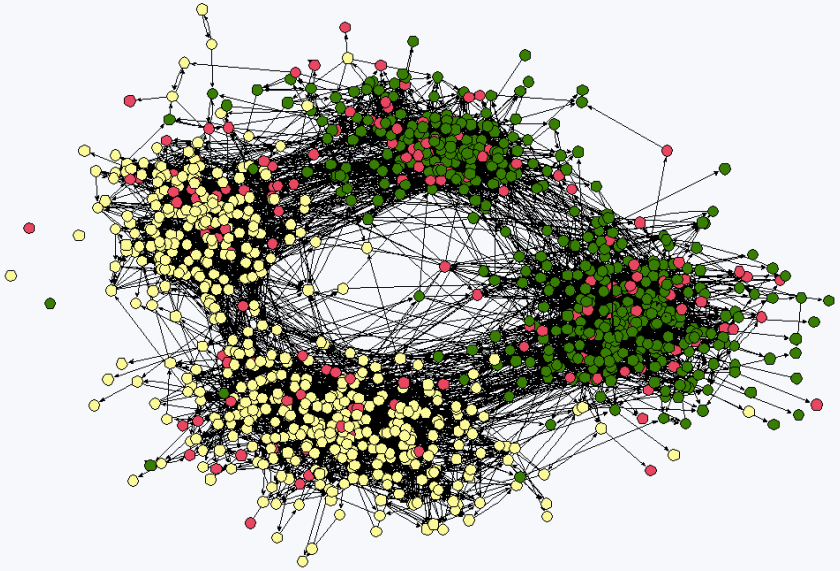
Internet



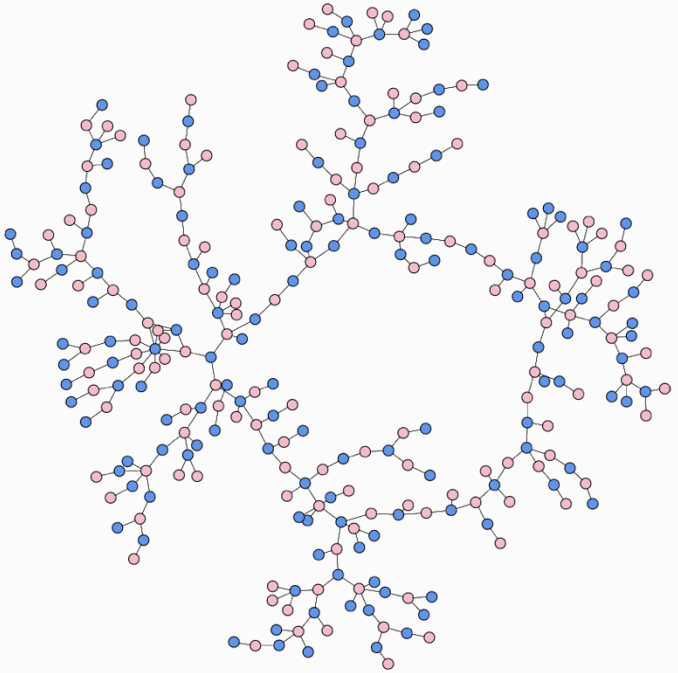
Social Networks and Social Network Analysis

- A social network
  - A network where elements have a social structure
    - A set of actors (such as individuals or organizations)
    - A set of ties (connections between individuals)
- Social networks examples:
  - your family network, your friend network, your colleagues, etc.
- To analyze these networks we can use Social Network Analysis (SNA)
- Social Network Analysis is an interdisciplinary field from social sciences, statistics, graph theory, complex networks, and computer science

Social Networks: Examples



High school friendship



High school dating

Social Network Mining tasks

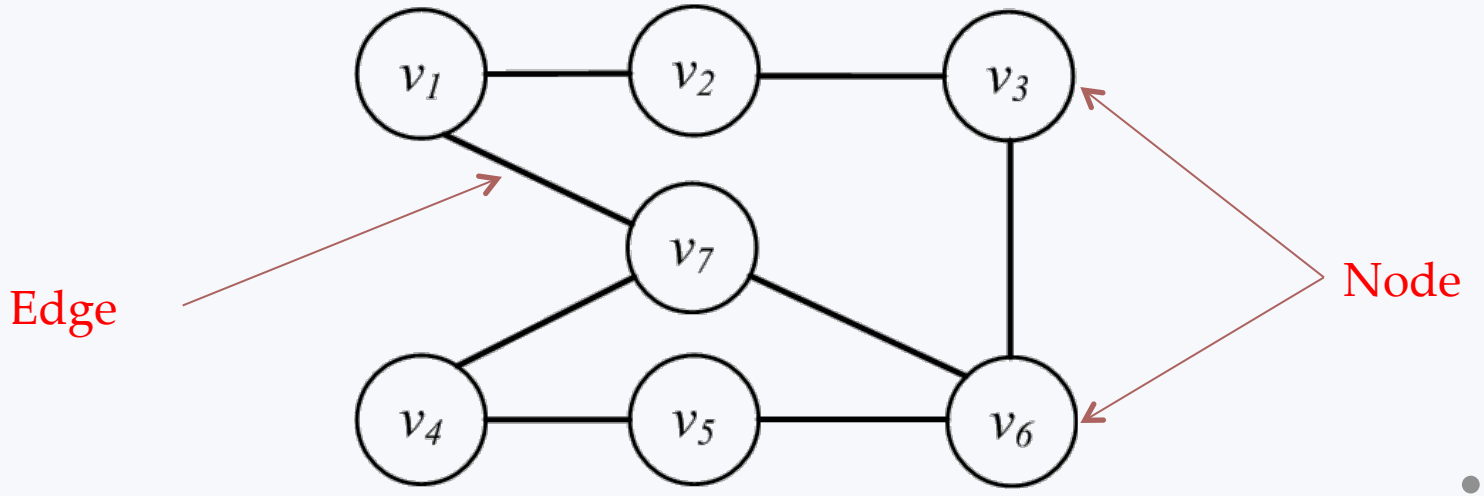
- Link analysis
  - Analyzing the network structure
  - E.g., which actors are most important/influential
- Community detection
  - Finding tightly connected sets of actors
  - E.g., a social clique or voting block
- Classification
  - E.g., is person X interested in buying product Y?
- Anomaly detection
  - E.g., which civilians are actually terrorists?

Graph Basics

Nodes and Edges

A network is a graph, or a collection of points connected by lines

- Points are referred to as **nodes**, **actors**, or **vertices** (plural of **vertex**)
- Connections are referred to as **edges** or **ties**



Nodes or Actors

- In a friendship social graph, nodes are people and any pair of people connected denotes the friendship between them
- Depending on the context, these nodes are called vertices or actors
  - In a web graph, "nodes" represent sites and the connection between nodes indicates web-links between them
  - In a social setting, these nodes are called actors

$$V = \{v_1, v_2, \ldots, v_n\}$$

  - The size of the graph is $|V| = \mathbf{n}$

Edges

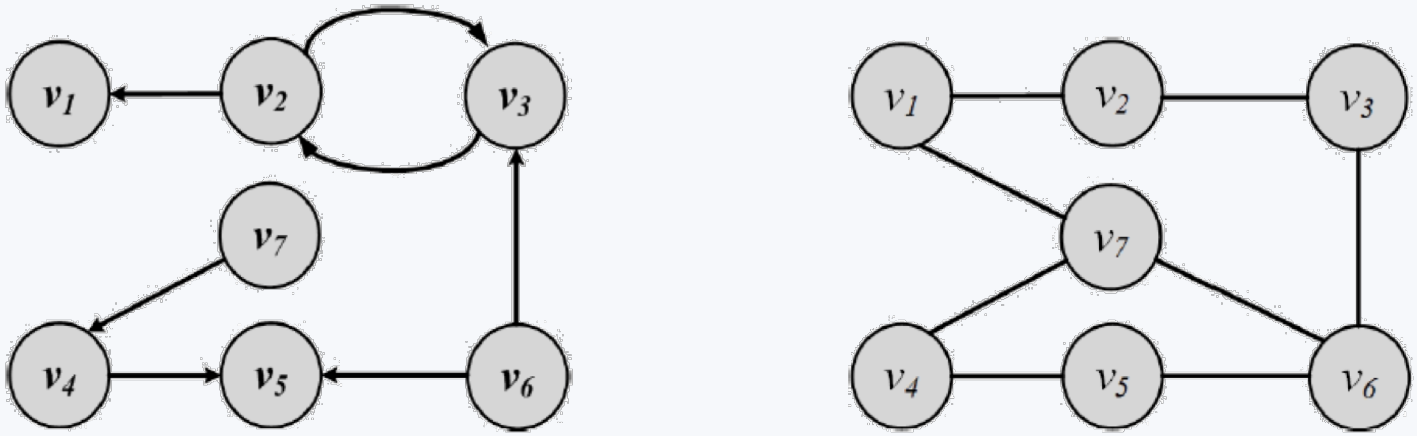
- Edges connect nodes and are also known as **ties** or **relationships**
- In a social setting, where nodes represent social entities such as people, edges indicate internode relationships and are therefore known as relationships or (social) ties

$$E = \{e_1, e_2, \ldots, e_m\}$$

- Number is edges (size of the edge-set) is denoted as $|E| = \mathbf{m}$

Directed Edges and Directed Graphs

- Edges can have directions. A directed edge is sometimes called an arc



- Edges are represented using their end-points (v_2,v_1). In undirected graphs both representations are the same

Neighborhood and Degree (In-degree, out-degree)

- For any node *v*, the set of nodes it is connected to via an edge is called its neighborhood and is represented as *N(v)*
- The number of edges connected to one node is the degree of that node (the size of its neighborhood)
  - Degree of a node *i* is usually presented using notation d_i
  - In case of directed graphs

d_i^{in}
- In-degrees is the number of edges pointing towards a node

d_i^{out}
- Out-degree is the number of edges pointing away from a node

Degree Distribution

When dealing with very large graphs, how nodes' degrees are distributed is an important concept to analyze and is called ***Degree Distribution***

p_d

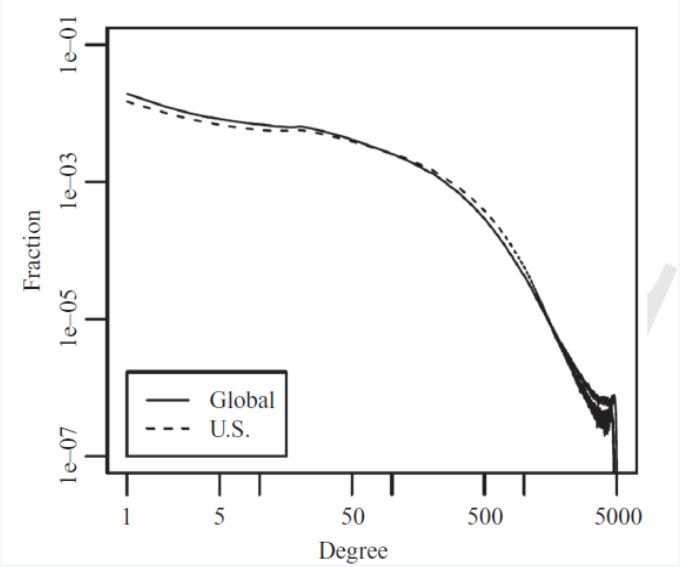
$$p_d = \frac{n_d}{n}$$

- Where n_d is the number of nodes with degree d
- Degree distribution can be computed from **degree sequence:**

$$\pi(d) = \{d_1, d_2, \ldots, d_n\}$$

Degree distribution histogram

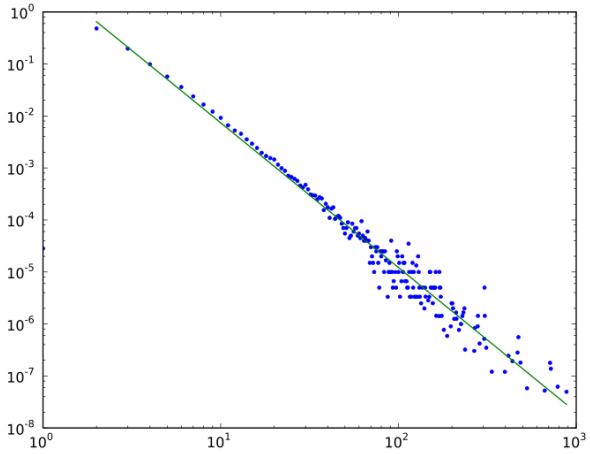
– The x-axis represents the degree and the y-axis represents the number of nodes (frequency) having that degree



Power law degree distribution

- Many social networks have a *power law degree distribution*
  - The number of actors with degree *d* is proportional to e^{-d}
- Few *hub* nodes with many ties
  - E.g., celebrities or politicians

	Twitter users	Followers	Following
1	KATY PERRY @katyperry	85,416,234	159
2	Justin Bieber @justinbieber	78,104,944	261,401
3	Taylor Swift @taylorswift13	73,838,674	245
4	Barack Obama @BarackObama	71,970,752	637,021
5	YouTube @YouTube	60,940,798	922
6	Rihanna @rihanna	57,977,699	1,137

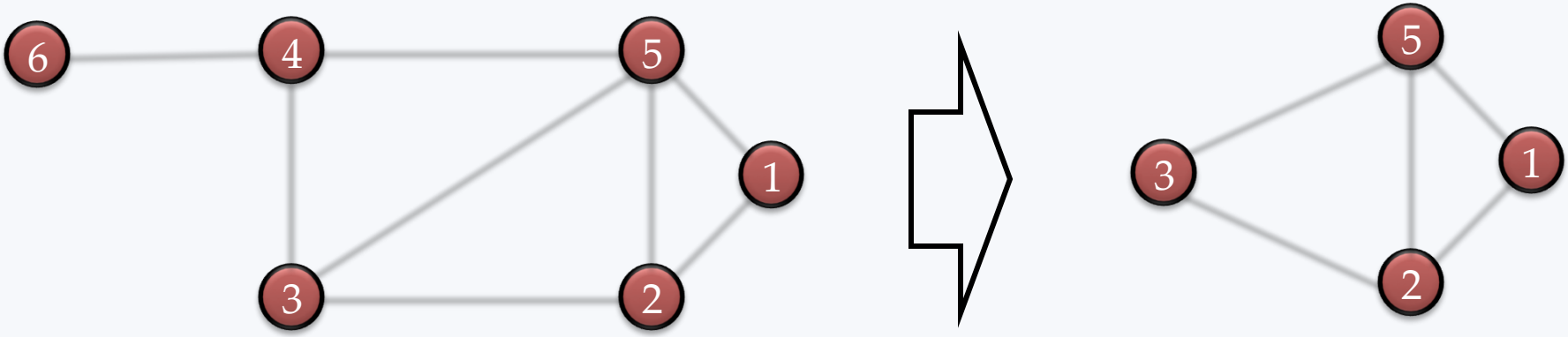


Subgraph

- Graph G can be represented as a pair G(V; E), where V is the node set and E is the edge set
- G'(V',E') is a subgraph of G(V,E)

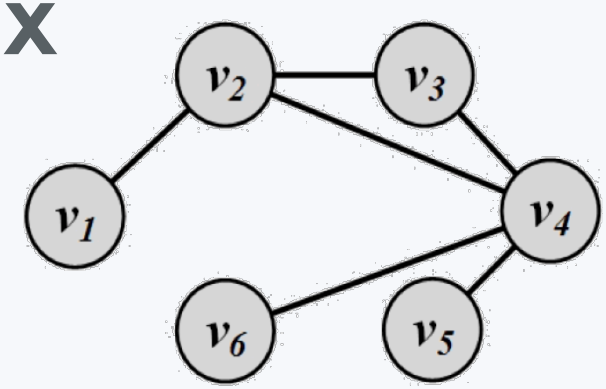
$$V' \subseteq V,$$

$$E' \subseteq (V' \times V') \cap E$$



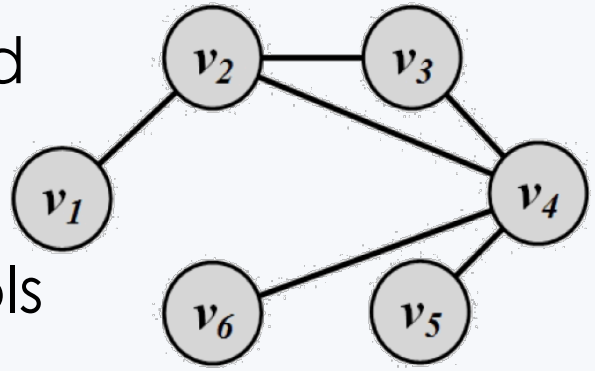
Graph Representation

- Adjacency Matrix
- Adjacency List
- Edge List



Graph Representation

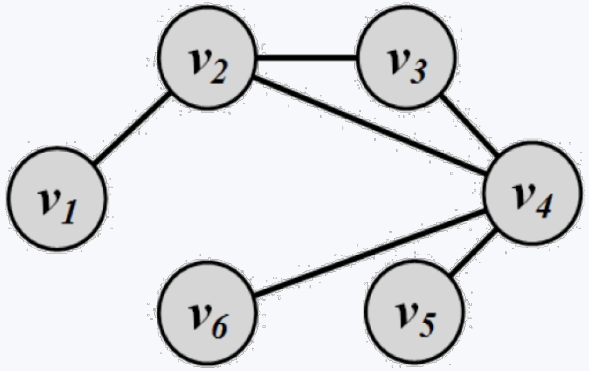
- Graph representation is straightforward and intuitive, but it cannot be effectively manipulated using mathematical and computational tools



- We are seeking representations that can store these two sets in a way such that
  - Does not lose information
  - Can be manipulated easily by computers
  - Can have mathematical methods applied easily

Adjacency Matrix

$$A_{ij} = \begin{cases} 1, \text{ if there is an edge between nodes } vi \text{ and } vj \\ 0, \text{ otherwise} \end{cases}$$



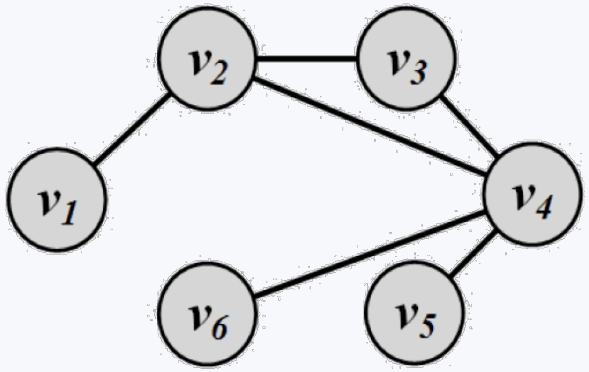
	v_1	v_2	v_3	v_4	v_5	v_6
v_1	0	1	0	0	0	0
v_2	1	0	1	1	0	0
v_3	0	1	0	1	0	0
v_4	0	1	1	0	1	1
v_5	0	0	0	1	0	0
v_6	0	0	0	1	0	0

- Diagonal Entries are self-links or loops

Social media networks have very sparse Adjacency matrices

Adjacency List

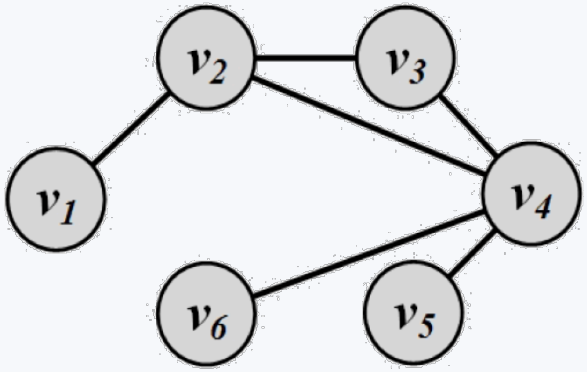
- In an adjacency list for every node, we maintain a list of all the nodes that it is connected to
- The list is usually sorted based on the node order or other preferences



Node	Connected To
v_1	v_2
v_2	v_1, v_3, v_4
v_3	v_2, v_4
v_4	v_2, v_3, v_5, v_6
v_5	v_4
v_6	v_4

Edge List

- In this representation, each element is an edge and is usually represented as *(u, v)*, denoting that node *u* is connected to node *v* via an edge



(v_1, v_2)
(v_2, v_3)
(v_2, v_4)
(v_3, v_4)
(v_4, v_5)
(v_4, v_6)

Types of Graphs

- Null, Empty, Directed/ Undirected, Simple/Multigraph, Weighted

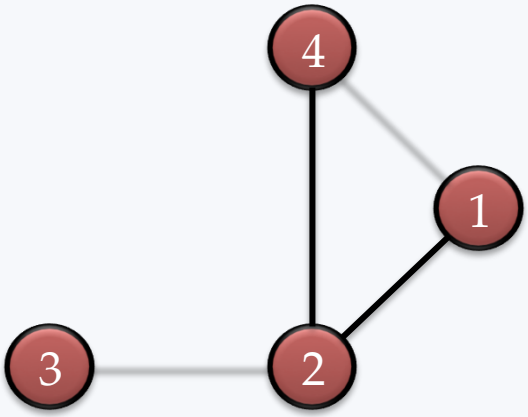
Null Graph and Empty Graph

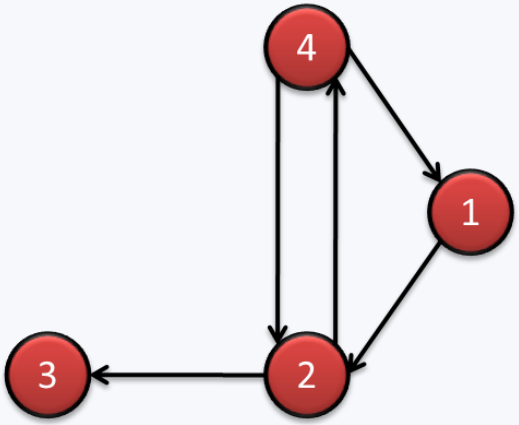
- A **null graph** is one where the node set is empty (there are no nodes)
  - Since there are no nodes, there are also no edges

$$G(V, E),\ V = E = \varnothing$$

- An **empty graph** or **edge-less graph** is one where the edge set is empty, $E = \varnothing$
  - The node set can be non-empty.
  - A null-graph is an empty graph.

Directed/Undirected Graphs

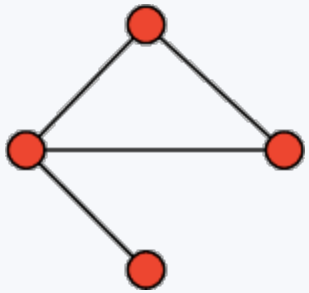




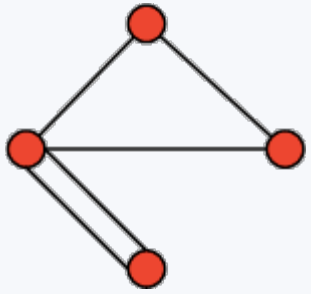
- The adjacency matrix for directed graphs is not symmetric ($A \neq A^T$)
  - E.g., Twitter followers
- The adjacency matrix for undirected graphs is symmetric ($A = A^T$)
  - E.g., Facebook friends

Simple Graphs and Multigraphs

- Simple graphs are graphs where only a single edge can be between any pair of nodes
- Multigraphs are graphs where you can have multiple edges between two nodes and loops



Simple graph

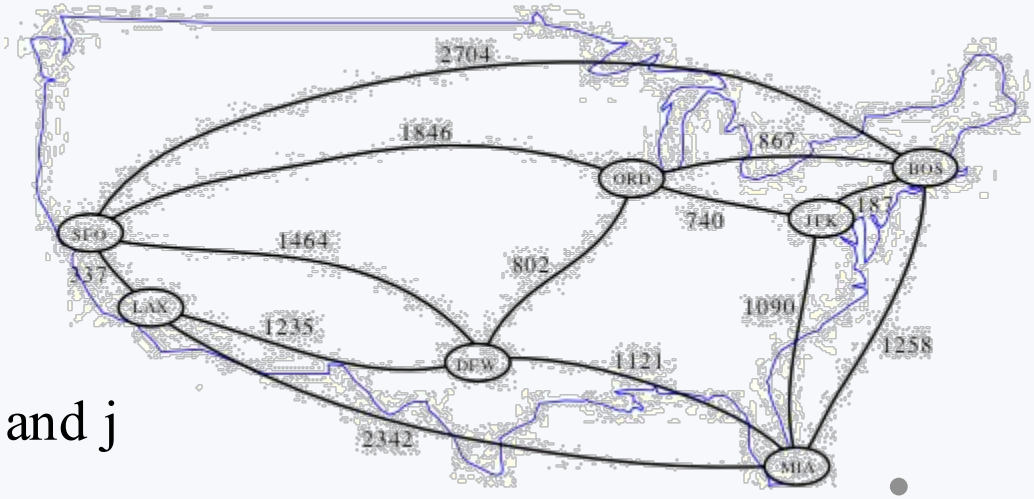


Multigraph

- The adjacency matrix for multigraphs can include numbers larger than one, indicating multiple edges between nodes
  - Web graph is a directed multigraph

Weighted Graph

- A weighted graph is one where edges are associated with weights
  - For example, a graph could represent a map where nodes are cities and edges are routes between them
    - The weight associated with each edge could represent the distance between these cities



G(V, E, W)

$$A_{ij} = \begin{cases} w, w \in R \\ 0, \text{There is no edge between i and j} \end{cases}$$

Connectivity in Graphs

- Adjacent nodes, Incident edges, Walks, Paths, Cycles

Adjacent nodes and Incident Edges

Two nodes are adjacent if they are connected via an edge.

Two edges are incident, if they share on end-point

When the graph is directed, edge directions must match for edges to be incident

An edge in a graph can be traversed when one starts at one of its end-nodes, moves along the edge, and stops at its other end-node.

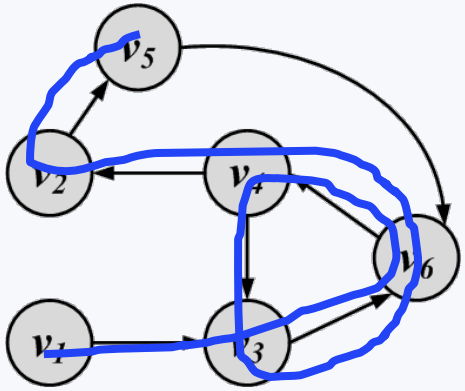
Walks and Paths

Walk: A walk is a sequence of incident edges visited one after another

- **Open walk**: A walk does not end where it starts
- **Closed walk**: A walk returns to where it starts

- Representing a walk:
  - A sequence of nodes: $v_1, v_2, \ldots, v_n$
  - A sequence of edges: $e_1, e_2, \ldots, e_n$
- Length of walk: the number of visited edges

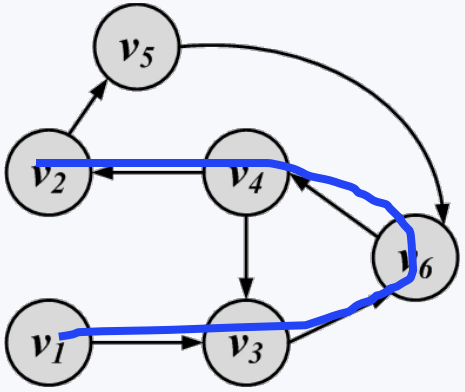
Length of walk= 8



Path

- A walk where **nodes and edges are distinct** is called a **path** and a closed path is called a **cycle**
- The length of a path or cycle is the number of edges visited in the path or cycle

Length of path= 4



Random walk

- A walk that in each step the next node is selected randomly among the neighbors
  - The weight of an edge can be used to define the probability of visiting it
  - For all edges that start at v_i the following equation holds

$$\sum_x w_{i,x} = 1, \forall i, j w_{i,j} \geq 0.$$

Random walk: example

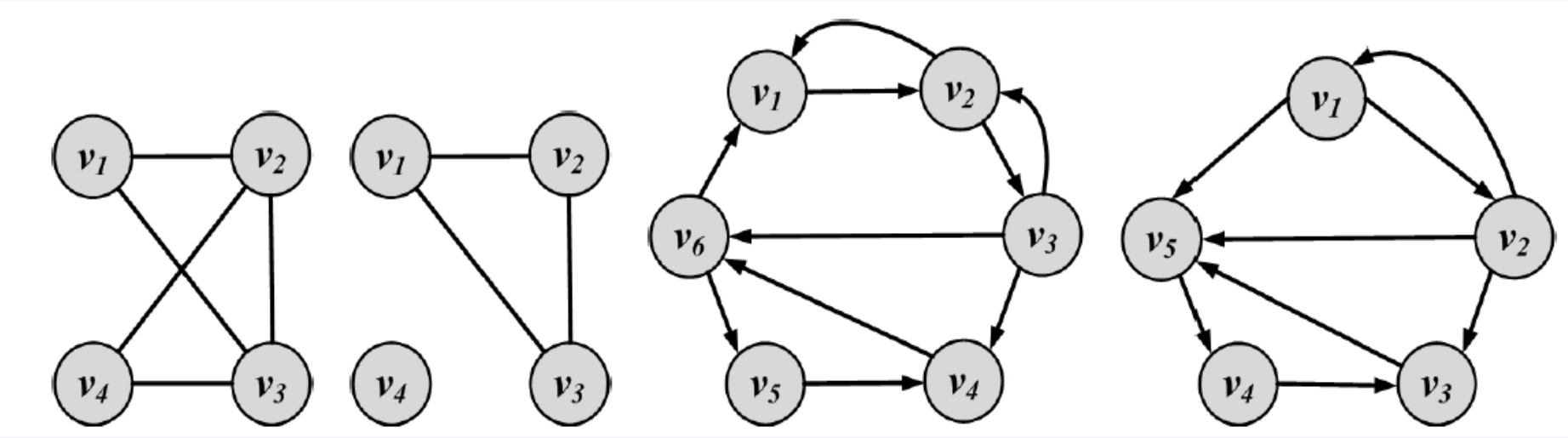
- Mark a spot on the ground
  - Stand on the spot and flip the coin (or more than one coin depending on the number of choices such as left, right, forward, and backward)
  - If the coin comes up heads, turn to the right and take a step
  - If the coin comes up tails, turn to the left and take a step
  - Keep doing this many times and see where you end up
- Random walks can be used for link analysis or community detection
  - RWs tend to visit important nodes many times
  - RWs tend to cluster in communities

Connectivity

- **A node v_i is connected** to node v_j (or reachable from v_j) if it is adjacent to it or there exists a path from v_i to v_j.

- **A graph is connected**, if there exists a path between any pair of nodes in it
  - In a directed graph, **a graph is strongly connected** if there exists a directed path between any pair of nodes
  - In a directed graph, **a graph is weakly connected** if there exists a path between any pair of nodes, without following the edge directions
- A graph is **disconnected**, if it not connected.

Connectivity: Example



Connected Disconnected Strongly connected Weakly connected

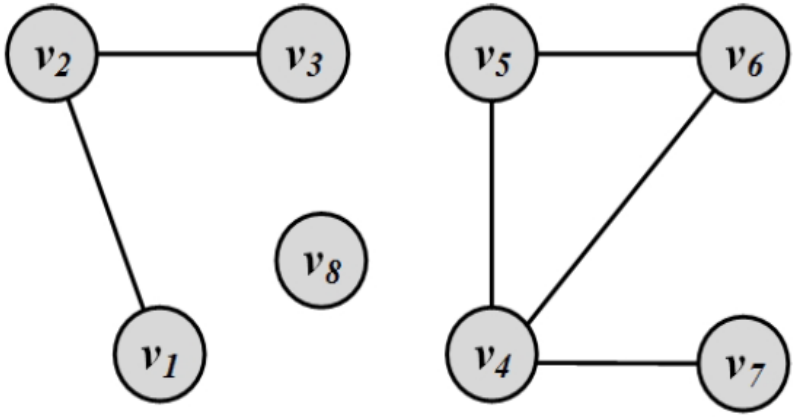
Component

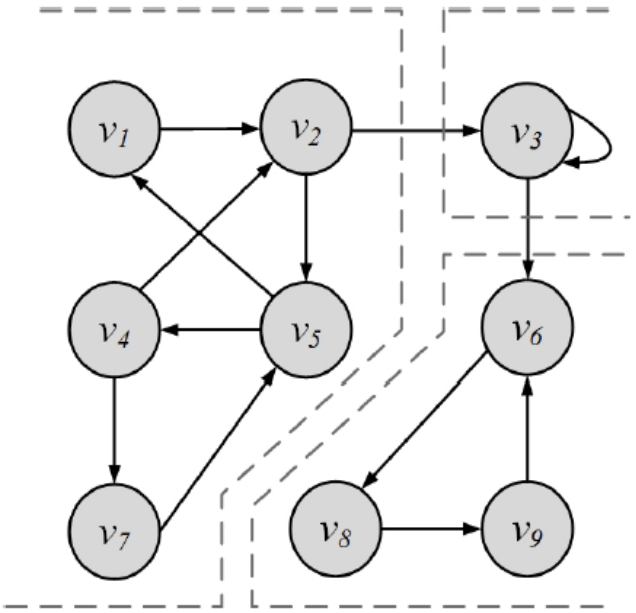
- A **component** in an undirected graph is a connected **subgraph**, i.e., there is a path between every pair of nodes inside the component

- In directed graphs, we have a **strongly connected** components when there is a path from *u* to *v* and one from *v* to *u* for every pair *(u,v)*.

- The component is **weakly connected** if replacing directed edges with undirected edges results in a connected component

Component Examples:





3 components

3 Strongly-connected components

Shortest Path

- **Shortest Path** is the path between two nodes that has the shortest length.
  - We denote the length of the shortest path between nodes $l_{i,j}$ nd v_j as
  - AKA, the *distance* between v_i and v_j

- The concept of the neighborhood of a node can be generalized using shortest paths. An **n-hop neighborhood** of a node is the set of nodes that are within n hops distance from the node.

Diameter

- The diameter of a graph is the length of the longest shortest path between any pair of nodes between any pairs of nodes in the graph

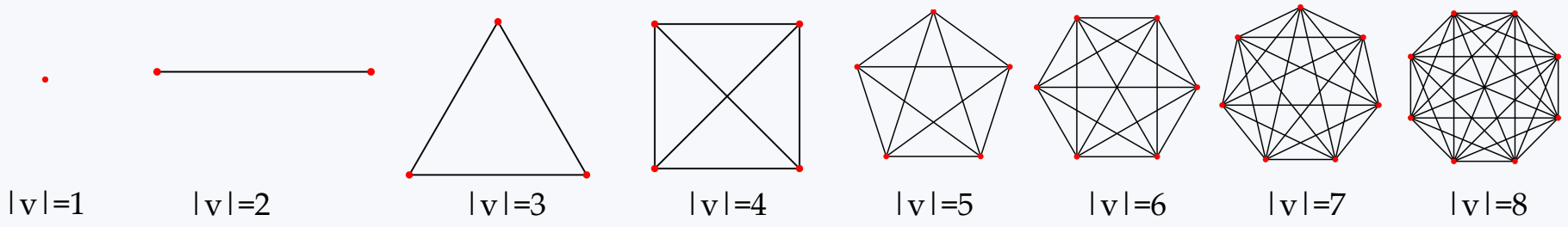
$$diameter_G = \max_{(v_i, v_j) \in V \times V} l_{i,j}.$$

Special Graphs

Complete Graphs

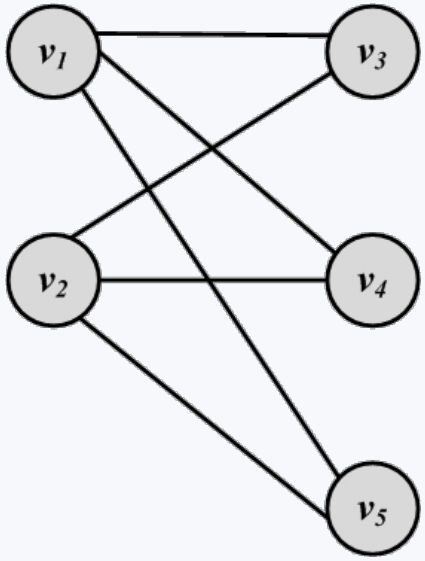
- A complete graph is a graph where for a set of nodes V, all possible edges exist in the graph
- In a complete graph, any pair of nodes are connected via an edge
- Also called a *clique*
- Most extreme definition for a community

$$E = \binom{|V|}{2}$$



Bipartite Graphs

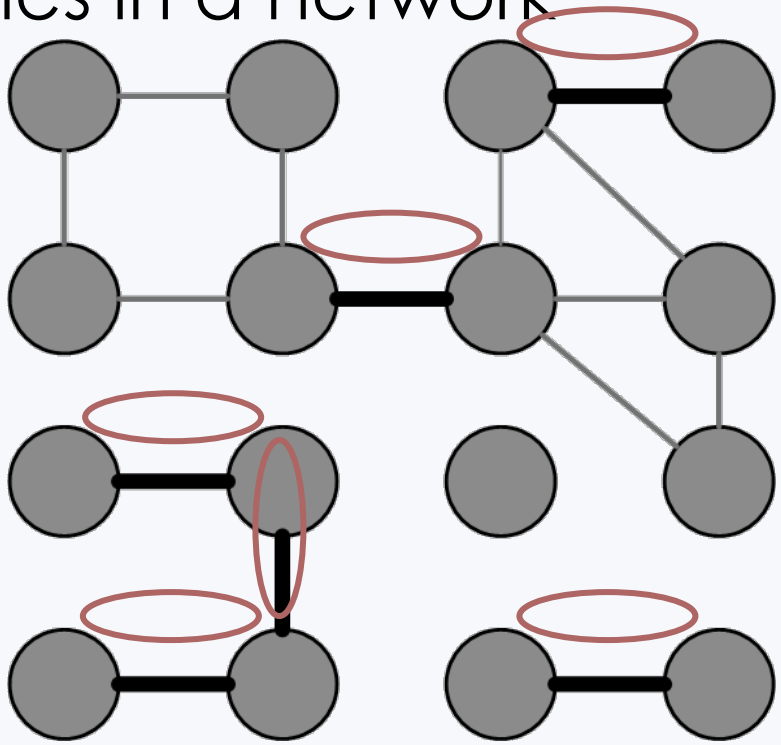
- A bipartite graph G(V; E) is a graph where the node set can be partitioned into two sets such that, for all edges, one end-point is in one set and the other end-point is in the other set.
- Can represent connections between different types of actors
  - E.g., people who join clubs, authors who write papers, students who take classes, etc.



$$\begin{cases} V = V_L \cup V_R, \\ V_L \cap V_R = \emptyset, \\ E \subset V_L \times V_R \end{cases}$$

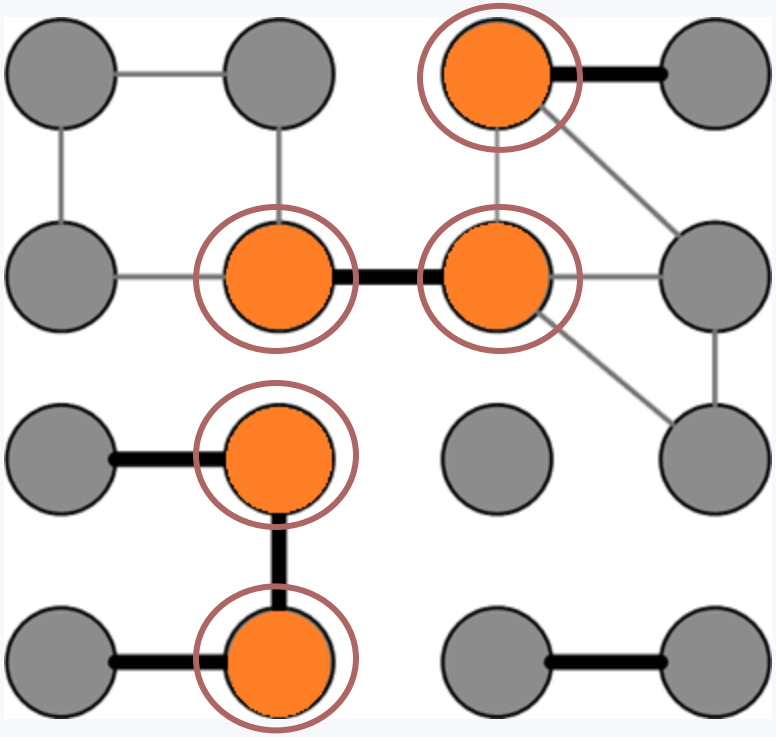
Cut edges (bridges)

- Cut edges are edges whose removal will increase the number of connected components
- Serve as critical connections or vulnerabilities in a network



Cut vertices (articulation points)

- Cut vertices are nodes whose removal will increase the number of connected components
  - Remove all incident edges as well
- E.g., actors that connect two different groups



Next week

• • •

Topic: Link prediction, who-follows-who problem in social networks