Graph Representation

- **Graph**
  - An ordered pair $G(V,E)$ with a set of vertices $V$ and a set of edges $E$

- **Extended Graph Representation**
  - Directed vs. undirected graph
    - Whether each edge has a direction
  - Weighted vs. unweighted graph
    - Whether each edge has a weight
  - Labeled vs. unlabeled graph
    - Whether each vertex has a label
  - 2-D vs. 3-D graph representation
    - Each vertex has angles between two linked edges
Why Graph Mining is Important?

- **Data are often represented as a graph**
  - Biological networks
  - Chemical compounds
  - Internet
  - WWW
  - Electric circuits
  - Workflows
  - Social networks

- **Graph is a general model for data mining !!**

Graph Data Mining Topics (1)

- **Single Graph Mining**
  - Frequent sub-graph pattern mining
    - Finding sub-graphs that frequently occur in a graph
  - Graph clustering (Vertex clustering)
    - Partitioning a graph into sub-graphs
  - Vertex classification
    - Classifying a vertex in a graph
Graph Data Mining Topics (2)

- **Graph Dataset Mining**
  - Frequent sub-graph pattern mining
    - Finding sub-graphs that frequently occur among graphs
  - Graph data clustering
    - Grouping similar graphs
  - Graph data classification
    - Classifying a new graph

<table>
<thead>
<tr>
<th>id</th>
<th>graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image1" alt="Graph 1" /></td>
</tr>
<tr>
<td>2</td>
<td><img src="image2" alt="Graph 2" /></td>
</tr>
<tr>
<td>3</td>
<td><img src="image3" alt="Graph 3" /></td>
</tr>
<tr>
<td>4</td>
<td><img src="image4" alt="Graph 4" /></td>
</tr>
</tbody>
</table>

Applications

- **Application of Single Graph Mining**
  - Biological network analysis
  - Social network analysis
  - Web community analysis

- **Application of Graph Dataset Mining**
  - Biochemical structure analysis
  - Program control flow analysis
  - XML structure analysis

- **Challenges**
  - Finding the complete set satisfying the minimum support threshold
  - Developing efficient and scalable algorithms
  - Incorporating various kinds of user-specific constraints
Chapters 11 and 13, Graph Data Mining

- **General Definitions**
  - **Graph Clustering**
  - **Subgraph Pattern Mining**

**Connectivity**

- **Degree, deg(vᵢ)**
  - The number of links from vᵢ to other vertices
  - Incoming degree and outgoing degree for directed graphs
  - Weighted degree (sum of the weights of the edges directly connected) for weighted graphs

- **A Set of Neighbors, N(vᵢ)**
  - A set of vertices directly linked to the vertex vᵢ
  - Also called adjacent neighbors or direct neighbors

- **Degree Distribution, P(k)**
  - Probability that a vertex has exactly k links
  - The number of vertices whose degree is k over the total number of vertices
Length & Size

- Walk
  - A sequence of vertices such that each is linked to its succeeding one
- Path
  - A walk such that each vertex in the walk is distinct
- Path Length
  - The number of edges in path $p$
- Shortest Path between $v_i$ and $v_j$
  - A path with the smallest length out of all paths from $v_i$ to $v_j$
- Characteristic Path Length of $G$
  - Average length of the shortest paths between each pair of vertices
- Diameter of $G$
  - Largest length of the shortest paths between each pair of vertices

Density

- Density of $G(V,E)$
  - The number of actual edges in $G$ over the number of all possible edges
  \[ D(G) = \frac{2|E|}{|V|(|V|-1)} \]
- Clique
  - A fully connected graph (also called, complete graph)
  - $D(G) = 1$
- Quasi-Clique
  - Close to clique
  - A densely connected sub-graph
  - $D(G) > \theta$ where $\theta$ is a user-specified threshold
Modularity

- **Clustering Coefficient of** $v_i$
  - The density of a sub-graph $G'(V',E')$ where $V'$ is the set of neighbors of $v_i$
  - $C(v_i) = \frac{\bigcup_{j, z \in N(v_i)} \{v_j, v_z\}}{|N(v_i)|(|N(v_i)|-1)}$
  - Measuring the effectiveness of $v_i$ on denseness

- **Average Clustering Coefficient of** $G(V,E)$
  - Average of the clustering coefficients of all vertices in $V$
  - Maximum is 1
  - Measuring the modularity of $G$

---

Centrality

- **Closeness**, $C_c(v_i)$
  - Detects the vertices located in the center of a graph
  - $C_c(v_i) = \frac{1}{\sum_{j \neq i} |p_i(v_j, v_i)|}$
  - where $|p_i(v_j, v_i)|$ is the shortest path length between $v_j$ and $v_i$

- **Betweenness**, $C_b(v_i)$
  - Detects the vertices located between two clusters
  - $C_b(v_i) = \sum_{s \neq i \neq t} \frac{\sigma_s(v_i)}{\sigma_s}$
  - where $\sigma_{st}$ is the number of shortest paths between $s$ and $t$, and
  $\sigma_s(v_i)$ is the number of shortest paths between $s$ and $t$, which pass through the vertex $v_i$
Chapters 11 and 13, Graph Data Mining

- General Definitions
- Graph Clustering
- Subgraph Pattern Mining

Graph Clustering

- Problem Definition
  - Finding densely connected sub-graphs $G'(V',E')$ from a graph $G(V,E)$
  - Finding sub-graphs with dense intra-connections and sparse interconnections (modularity)

- Methods
  - Density-based methods
  - Hierarchical methods
  - Partition-based methods
Chapters 11 and 13, Graph Data Mining

- General Definitions
- Graph Clustering
  - Density-based Methods
  - Partition-based Methods
  - Hierarchical Methods
- Subgraph Pattern Mining