Social and Graph Data Management: Introduction to Data Models and Measures

Pablo Mollá Chárlez

Contents

1 Introduction to Data Models and Measures

Social networks are **abstract representations of the relationships between individuals**, capturing the intricate web of human interactions. These networks manifest across various domains, such as within organizations (e.g., companies, classrooms), professional fields (e.g., physics researchers), and online platforms (e.g., Facebook friends, Twitter followers).

Figure 1: Web Social Networks by Michael Coghlan

In this course, we will explore the models and measures essential for graph analysis, focusing on identifying the unique properties that distinguish social networks from other types of graphs. Additionally, we will examine key applications of social graph data, including influence analysis and link prediction, to understand how these networks impact and predict real-world behaviors and connections.

2 Graphs

A graph is a mathematical structure used to model pairwise relationships between objects. Formally, a graph G is defined as a tuple $G = (V, E, w)$ where:

- V is a finite set of **vertices** or nodes.
- $E \subseteq \times V$ is a set of edges or **links**, representing binary relationships between vertices.
- $w : E \to \mathbb{R}$ is an optional weight function assigning a weight to each edge.

Let $N = |V|$ denote the number of vertices and $L = |E|$ the number of edges in the graph.

2.1 Types of Graphs

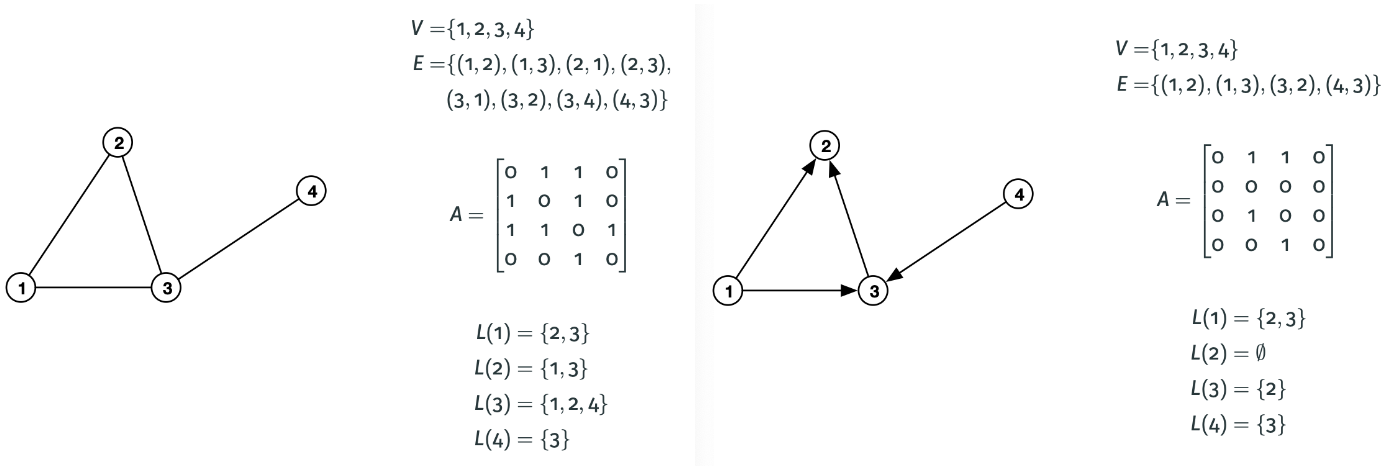
Graphs can be categorized based on the nature of their edges and weights:

2.1.1 Directed vs Undirected

Firstly, we can distinguish between **Directed** and **Undirected** graphs.

- **Undirected Graph**: For every edge $\{v_i, v_j\} \in E$, the edge $\{v_j, v_i\}$ also exists. There is no inherent direction between connected vertices.
- **Directed Graph (Digraph)**: Edges have a direction, represented as ordered pairs (v_i, v_j). The presence of (v_i, v_j) does not imply (v_j, v_i).

The following examples are unweighted graphs:



$V = \{1, 2, 3, 4\}$
$E = \{(1,2), (1,3), (2,1), (2,3), (3,1), (3,2), (3,4), (4,3)\}$

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$L(1) = \{2, 3\}$
$L(2) = \{1, 3\}$
$L(3) = \{1, 2, 4\}$
$L(4) = \{3\}$

$V = \{1, 2, 3, 4\}$
$E = \{(1,2), (1,3), (3,2), (4,3)\}$

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

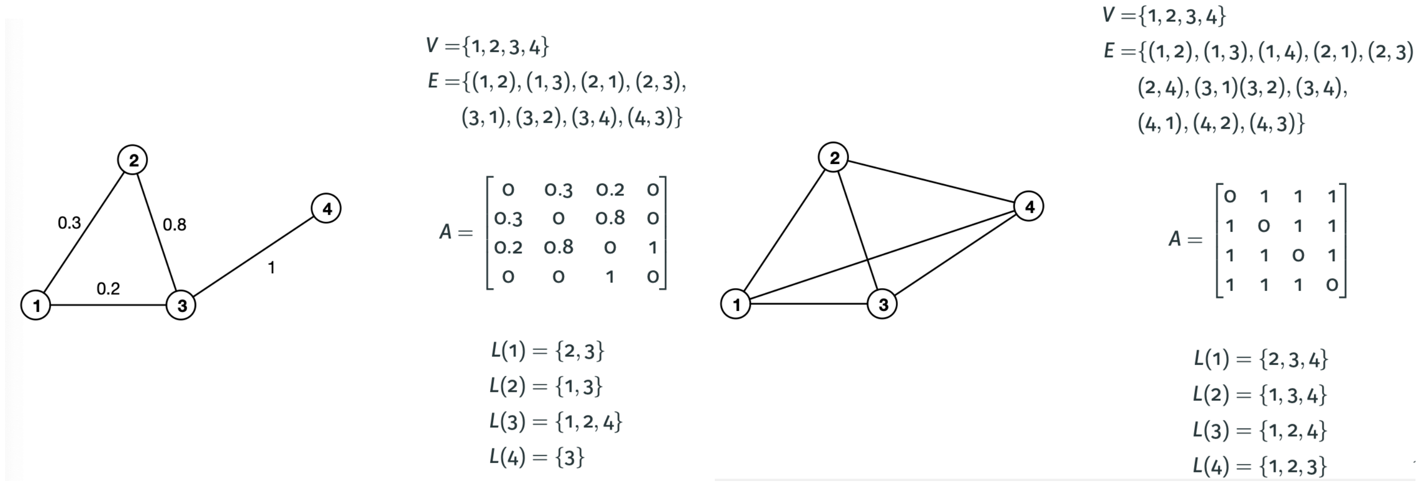
$L(1) = \{2, 3\}$
$L(2) = \emptyset$
$L(3) = \{2\}$
$L(4) = \{3\}$

Figure 2: Undirected Graph & Directed Graph

2.1.2 Weighted vs. Unweighted

Secondly, we differentiate between **Weighted** and **Unweighted** Graphs.

- **Weighted Graph**: A weight function w is defined, assigning a numerical value to each edge, often representing **cost**, **distance**, or **capacity**.
- **Unweighted Graph**: No weight function is defined; edges are either present or absent without any associated value.



$V = \{1, 2, 3, 4\}$
$E = \{(1,2), (1,3), (2,1), (2,3), (3,1), (3,2), (3,4), (4,3)\}$

$$A = \begin{bmatrix} 0 & 0.3 & 0.2 & 0 \\ 0.3 & 0 & 0.8 & 0 \\ 0.2 & 0.8 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$L(1) = \{2, 3\}$
$L(2) = \{1, 3\}$
$L(3) = \{1, 2, 4\}$
$L(4) = \{3\}$

$V = \{1, 2, 3, 4\}$
$E = \{(1,2), (1,3), (1,4), (2,1), (2,3)$
$(2,4), (3,1)(3,2), (3,4),$
$(4,1), (4,2), (4,3)\}$

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

$L(1) = \{2, 3, 4\}$
$L(2) = \{1, 3, 4\}$
$L(3) = \{1, 2, 4\}$
$L(4) = \{1, 2, 3\}$

Figure 3: Weighted Undirected Graph & Perfect Graph

Remark A complete graph is an undirected graph in which every pair of distinct vertices is connected by a unique edge.

2.2 Representing Edges

Edges in a graph can be represented using different data structures, each with its advantages and trade-offs:

- Adjacency Matrix An adjacency matrix A is an $N \times N$ matrix where:

$$A_{i,j} = \begin{cases} w(i,j) & \text{if } \{v_i, v_j\} \in E \text{ (for weighted graphs)} \\ 1 & \text{if } \{v_i, v_j\} \in E \text{ (for unweighted graphs)} \\ 0 & \text{otherwise} \end{cases}$$

The main characteristics are:

  - **Space Complexity**: Requires $O(N^2)$ space.
  - **Use Case**: Efficient for dense graphs where the number of edges is close to N^2.
  - **Access Time**: Allows constant-time $O(1)$ access to check the existence or weight of an edge.

- Adjacency List

  An adjacency list L consists of N lists, where each list $L(i)$ contains all vertices v_j such that $\{v_i, v_j\} \in E$. The main characteristics are:

  - **Space Complexity**: Requires $O(N + E)$ space.
  - **Use Case**: Efficient for sparse graphs where the number of edges E is much less than N^2.
  - **Access Time**: Checking for the existence of a specific edge may require $O(N)$ time in the worst case, as it may involve searching through a list.

3 Measures & Properties

3.1 Degree of a node

The degree $k(i)$ of a node i is the number of nodes it is connected to via edges:

$$k(i) = |\{\,(i,j) \mid j \in V,\ (i,j) \in E\}|$$

For **directed graphs**, the degree is divided into:

- **Incoming degree** $k_{in}(i)$: $k_{\text{in}}(i) = |\{\,(j,i) \mid j \in V,\ (j,i) \in E\}|$
- **Outgoing degree** $k_{out}(i)$: $k_{\text{out}}(i) = |\{\,(i,j) \mid j \in V,\ (i,j) \in E\}|$

3.2 Degree Distribution

The degree distribution p_i is the probability that a randomly selected node has degree i:

$$p_i = \frac{N_i}{N}$$

where N_i is the number of nodes with degree i and N is the total number of nodes. This distribution satisfies:

$$\sum_{i=0}^{\infty} p_i = 1$$

3.3 Average Degree

The average degree $\langle k \rangle$ is the expected degree of a node in the graph:

$$\langle k \rangle = \sum_{i=0}^{\infty} i \cdot p_i = \frac{L}{N}$$

where L is the total number of edges and N is the total number of nodes

For instance, following the first example:

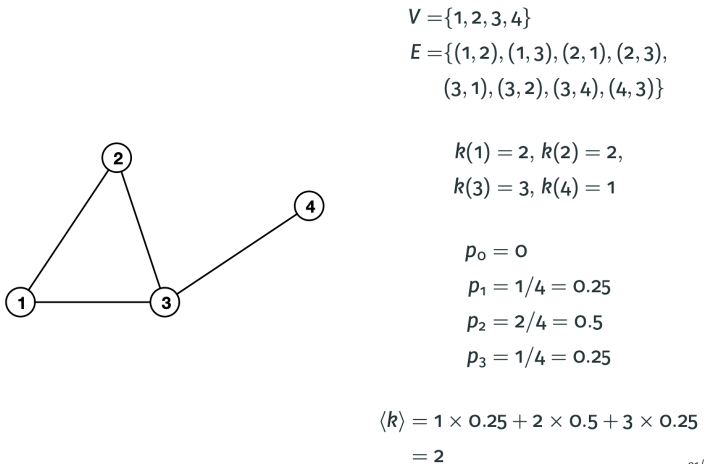


Figure 4: Node Degree, Degree Distribution & Average Degree

3.4 Paths in Graphs

A path in a graph is a sequence of nodes $v_1, v_2, \ldots, v_k$ in V where each consecutive pair (v_i, v_{i+1}) is an edge in E. For example, a path representation could be:

$$V = \{v_1, v_2, v_3, v_4\}, \; P = \{(v_1, v_2), (v_2, v_3), (v_3, v_4)\}$$

However, in **directed graphs**, paths must follow the direction of the edges.

3.4.1 Number of Paths of Length l

The **number of paths of length** l between two nodes i and j, denoted $N_{ij}^{(l)}$, can be computed using the adjacency matrix A:

$$\boxed{\textbf{For l = 1}} \quad N_{ij}^{(1)} = A_{ij}$$
$$\boxed{\textbf{For l > 1}} \quad N_{ij}^{(l)} = [A^l]_{ij}$$

Here, A^l represents the adjacency matrix raised to the $l - th$ power, and $[A^l]_{ij}$ is the entry in the $i - th$ row and $j - th$ column of A^l.

3.5 Distances in Graphs

The **distance** d_{ij} between two nodes i and j is defined as:

1. **Undirected Graph**:

$$d_{ij} = \text{number of edges in the shortest path between } i \text{ and } j$$

2. **Directed Graph**:

$$d_{ij} = \text{weight of the shortest path from } i \text{ to } j$$

In weighted directed graphs, the distance accounts for the cumulative weights along the shortest path.

3.6 Diameter of a Graph

The diameter $d_{\max}$ of a graph is the maximum distance between any pair of nodes in the graph:

$$d_{\max} = \max_{i,j \in V} d_{ij}$$

This metric represents the longest shortest path in the graph.

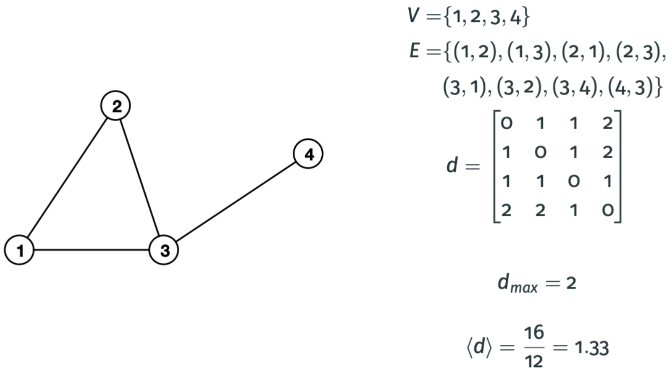

1
2
3
4
$V = \{1, 2, 3, 4\}$
$E = \{(1, 2), (1, 3), (2, 1), (2, 3), (3, 1), (3, 2), (3, 4), (4, 3)\}$
$d = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 \\ 2 & 2 & 1 & 0 \end{bmatrix}$
$d_{max} = 2$
$\langle d \rangle = \frac{16}{12} = 1.33$

Figure 5: Distance Matrix, Diameter and Average Distance

3.7 Average Distance

The average distance $\langle d \rangle$ in a graph is the mean of all pairwise distances between nodes:

$$\langle d \rangle = \frac{1}{N(N-1)} \sum_{\substack{i,j \in V \\ i \neq j}} d_{ij}$$

where $N = |V|$ is the total number of nodes and the summation considers all unique pairs of distinct nodes. For instance, in the previous 2 undirected graph version, we would have:

Notice that, for the number of paths of length 1, as mentioned in the definition, it coincides directly with the **adjacency matrix A**, however for larger lengths, we need to compute the power of the adjacency matrix A to the given length l. For instance,

- **Paths of Length** 1 **(Adjacency Matrix** A**)**:

$$N_{ij}^1 = A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

  Each (i, j) entry represents a direct connection between nodes i and j.

- **Paths of Length** 2 (A^2):

$$N_{ij}^2 = A^2 = \begin{bmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 3 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

  Each (i, j) entry represents the number of paths of length 2 between nodes i and j.

  For example:

  * From node 1 to node 1, there are 2 paths of length 2, $P_1 = \{(1, 2), (2, 1)\}$ and $P_2 = \{(1, 3), (3, 1)\}$.
  * From node 1 to node 4, there is 1 path of length 2 which is $P_3 = \{(1, 3), (3, 4)\}$.
  * From node 3 to node 3, there are 3 paths of length 2, $P_4 = \{(3, 1), (1, 3)\}, P_5 = \{(3, 2), (2, 3)\}, P_6 = \{(3, 4), (4, 3)\}$.

The example of Livejournal, which is a social network to exchange information and opinions, with a diameter of only 38 despite having millions of users (**vertices**) and connections (**edges**), illustrates a phenomenon known as the six degrees of separation. This principle suggests that, in large social networks, there are surprisingly few connections needed to link any two individuals. Even in vast networks, paths between users tend to be short, which is why most people are connected through only a small number of intermediaries. This is a common property of many real-world networks, where the **diameter** (the longest shortest path between any two nodes) remains small relative to the network's size.

3.8 Graph Connectivity

In graph theory, it can be distinguished two main types of graphs in terms of how they are connected:

1. **Undirected Graphs**:

   - Connected Graph: Any two vertices in the graph can be linked by a path, meaning there is a way to travel from any vertex to any other vertex.
   - Disconnected Graph: The graph is split into two or more connected components, where for each component is a subgraph in which any two vertices are connected by paths, but no path exists between vertices in different components.

2. **Directed Graphs**:

   - **Strongly Connected**: For any pair of vertices i and j, there exists a directed path from i to j and from j to i.
   - **Weakly Connected**: There is a path between any two vertices i and j if we ignore the direction of edges, treating the graph as undirected.

3.9 Clustering Coefficient

For a node i, the clustering coefficient C_i measures how interconnected the node's neighbors are. It's calculated as:

$$C_i = \frac{2e_i}{k_i(k_i - 1)}$$

where e_i is the number of edges between neighbors of i and k_i is the number of neighbors (degree) of i. The average clustering coefficient $\langle C \rangle$ provides a global measure of clustering in the graph, calculated as the average C_i across all nodes, defined as:

$$\langle C \rangle = \frac{1}{N} \sum_i C_i$$

The average clustering coefficient ranges from 0 to 1. The closer it is to 1, the more clustered the nodes are, meaning more of each node's neighbors are connected to each other. A value near 1 indicates a highly clustered (or "**cliquey**") network, while a value near 0 indicates little to no clustering. For instance, considering the undirected graph 2, it is **connected** and the clustering coefficient is computed as follows:

Explanation For the nodes in this graph:

1. **Node 1**: The degree of node 1 is $k_1 = 2$, it has neighbors 2 and 3, which are connected, so $e_1 = 1$. Therefore $C_1 = \frac{2 \cdot 1}{2 \cdot 1} = 1$.

2. **Node 2**: The degree of node 2 is $k_2 = 2$, it has neighbors 1 and 3, which are connected, so $e_2 = 1$. Therefore $C_2 = \frac{2 \cdot 1}{2 \cdot 1} = 1$.

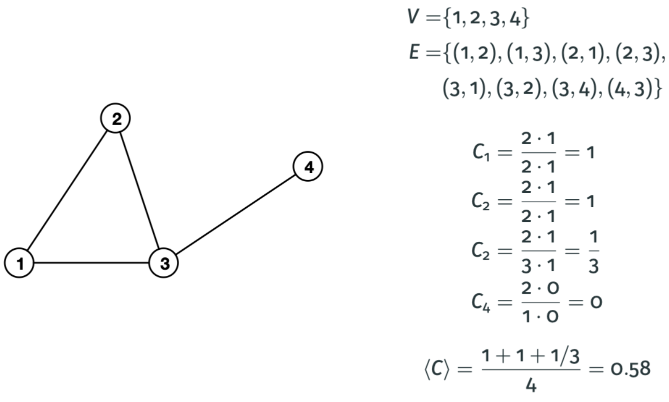
3. **Node 3**: The degree of node 3 is $k_3 = 3$, it has neighbors $1, 2$ and 4, in which only nodes 1 and 2 are connected, so $e_3 = 1$. Therefore $C_3 = \frac{2 \cdot 1}{3 \cdot 2} = \frac{1}{3} \approx 0.33$.

4. **Node 4**: The degree of node 4 is $k_4 = 1$, it has just node 3 as neighbor, then it has no connections between its neighbors (since it has only one neighbor, node 3), so $e_4 = 0$. Therefore $C_4 = \frac{2 \cdot 0}{1 \cdot 0} = 0$.

The average clustering coefficient $\langle C \rangle$ is the mean of the clustering coefficients for all nodes:

$$\langle C \rangle = \frac{1 + 1 + \frac{1}{3} + 0}{4} = 0.58$$

This average clustering coefficient indicates the overall tendency of the graph's nodes to form tightly connected groups. In this case, the value is moderate, showing some clustering but not a very dense one.



2
4
1
3
$V = \{1, 2, 3, 4\}$
$E = \{(1,2), (1,3), (2,1), (2,3),$
$(3,1), (3,2), (3,4), (4,3)\}$
$C_1 = \frac{2 \cdot 1}{2 \cdot 1} = 1$
$C_2 = \frac{2 \cdot 1}{2 \cdot 1} = 1$
$C_2 = \frac{2 \cdot 1}{3 \cdot 1} = \frac{1}{3}$
$C_4 = \frac{2 \cdot 0}{1 \cdot 0} = 0$
$\langle C \rangle = \frac{1 + 1 + 1/3}{4} = 0.58$

Figure 6: Average Clustering Coefficient

In the context of the **CondMat** collaboration network (a network representing collaborations between condensed matter physicists), the network has a clustering coefficient of 0.63 which implies that there is a 63% probability that two collaborators of a researcher also collaborate with each other. This high clustering coefficient indicates that researchers tend to work in tightly knit groups where many of their collaborators are also collaborating among themselves.

Generally, networks like social or collaboration networks exhibit higher clustering coefficients than random networks (networks formed by chance). This means they have a higher degree of local interconnectedness, reflecting the natural formation of communities or groups where individuals are more likely to be connected. Such a high clustering coefficient suggests that the network has a significant amount of clustering beyond what would be expected in a random network, highlighting the presence of cohesive subgroups within the larger network.

3.10 Node Centrality Measures

In graph theory, **node centrality measures** are used to identify the importance or influence of nodes within a network. Here's a quick overview of each measure, with calculations for the nodes in the previous graph:

1. **Degree Centrality (k_i)**: This is simply the degree of a node, representing the number of connections (edges) it has.

2. **Closeness Centrality (Cl_i)**: This measures how close a node is to all other nodes. It's the inverse of the sum of shortest path distances from the node to all others. **For each node, calculate the total distance to all others and take the inverse.** Closeness Centrality (Cl_i) is defined as:

$$Cl_{\mathrm{i}} = \frac{1}{\sum_j d_{ji}}$$

Let's use the undirected graph 2 as example. We remind that the **adjacency matrix** A for the graph is:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

We'll first compute the **shortest path distance matrix**, d, for all node pairs. The distances between each node are: $d_{12} = d_{21} = 1$, $d_{13} = d_{31} = 1$, $d_{14} = d_{41} = 2$, $d_{23} = d_{32} = 1$, $d_{24} = d_{42} = 2$, $d_{34} = d_{43} = 1$. The resulting distance matrix d is:

$$d = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 1 & 2 \\ 1 & 1 & 0 & 1 \\ 2 & 2 & 1 & 0 \end{bmatrix}$$

Then we, **calculate** $\sum_j d_{ji}$ and then Cl_i for each node:

$$\textbf{Node 1}: \sum_j d_{1j} = 0 + 1 + 1 + 2 = 4 \rightarrow Cl_1 = \frac{1}{4} = 0.25.$$

$$\textbf{Node 2}: \sum_j d_{2j} = 1 + 0 + 1 + 2 = 4 \rightarrow Cl_2 = \frac{1}{4} = 0.25.$$

$$\textbf{Node 3}: \sum_j d_{3j} = 1 + 1 + 0 + 1 = 3 \rightarrow Cl_3 = \frac{1}{3} \approx 0.333.$$

$$\textbf{Node 4}: \sum_j d_{4j} = 2 + 2 + 1 + 0 = 5 \rightarrow Cl_4 = \frac{1}{5} = 0.2.$$

3. **Betweenness Centrality**: The betweenness centrality $C_B(v)$ of a node v is defined as:

$$C_B(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths from node s to node t, and $\sigma_{st}(v)$ is the number of those shortest paths that pass through node v. Betweenness centrality measures how often a node serves as a bridge along the shortest path between other nodes. **For each node, count the shortest paths that pass through it**. Retaking the previous example, let's count for each node:

(a) **Node** 1: Does not lie on any shortest path between other pairs, because nodes $2, 3$ and 4 are directly connected, thus: $C_B(1) = 0$.

(b) **Node** 2: Does not lie on any shortest path between other pairs, because nodes $1, 3$ and 4 are directly connected, thus: $C_B(2) = 0$.

(c) **Node** 3: Lies on the shortest path between nodes 1 and 4, and between nodes 2 and 4.

$$C_B(3) = \sum_{s \neq 3 \neq t} \frac{\sigma_{st}(3)}{\sigma_{st}} = \frac{\sigma_{14}(3)}{\sigma_{14}} + \frac{\sigma_{24}(3)}{\sigma_{24}} = \frac{1}{1} + \frac{1}{1} = 2$$

Where $\sigma_{14}(3)$ accounts for the number of shortest paths that connect the node 1 and node 4 via node 3, and σ_{14} the number of shortest paths between nodes 1 and 4. In this scenario, they share the same value as there are no more possible paths to reach node 4 without passing through node 3.

(d) **Node** 4: Does not lie on any shortest path between other pairs, because nodes $1, 2$ and 3 are directly connected, thus: $C_B(4) = 0$.

4. **Eigenvector Centrality (e.g., PageRank)**: This considers not only the number of connections but also the influence of those connections. Nodes connected to high-ranking nodes will have higher centrality themselves. **Calculating eigenvector centrality requires finding the principal eigenvector of the adjacency matrix**.

It can be found by solving the eigenvector equation $A \cdot \vec{x} = \lambda \vec{x}$, where λ is the largest eigenvalue of A, and $\vec{x}$ is the eigenvector representing the centralities.