Network Analysis Cheat Sheet

What is a Network?

Networks consist of components and interactions that together form a system.

- **Components**: nodes, vertices
- **Interactions**: links, edges
- **System**: network, graph

Origins

- **1735:** Euler was puzzled by solving the bridges of Königsberg (origins of graph theory)
- **1950:** Erdős was puzzled by social networks structure
- **1999:** Barabási was puzzled by the Internet
- **Now:** We are puzzled by all of them (brain, social networks, communication, and transportation networks)

Overview

The field covers various types of networks and theories:

- **Graph theory:**
  - Origins (Eulerian graphs)
- **Complex networks:**
  - Random graphs (Erdos-Renyi)
  - Small world graphs (Watts-Strogatz)
  - Scale free graphs (Barabasi-Albert)
  - The configuration model (Molloy-Reed)

The Königsberg Bridges

Königsberg Bridges (now Kaliningrad, Russia): The problem of finding a route through the city that would cross each bridge exactly once was an early and significant problem in graph theory.

- A route must start and finish at the same place.
- It must cross each bridge exactly once.
- In graph terminology, a **vertex** represents a region, and an **edge** represents a bridge between two regions.

The Formulation and Evolution of Graph Theory

- The formulation of graph theory is attributed to **Leonhard Euler** with the Seven Bridges of Königsberg problem.
- Networks and graph theory became more popular, thanks in great part to **Paul Erdős**. His interest in networks was driven by the puzzles presented in social structures.

Random Graphs

What is the structure of social networks?

- Paul Erdős introduced the concept of random graphs in the 1950s.
- In random graphs, the existence of an edge is determined by a probability p.

Erdős-Rényi Model

In the 1960s, Paul Erdős and Alfréd Rényi developed a foundational model for random graphs, which is still used as a starting point for understanding the structure of networks. Their model, known as the **Erdős-Rényi model** or $G(n, p)$, describes a simple random graph where:

- n represents the number of nodes in the graph.
- p is the probability that any given pair of nodes is connected by an edge.
- The average degree (expected number of connections per node) can be estimated by $p(n-1)$.

This model is significant for exploring the properties of networks, such as connectivity and the presence of subgraphs, as a function of the probability p.

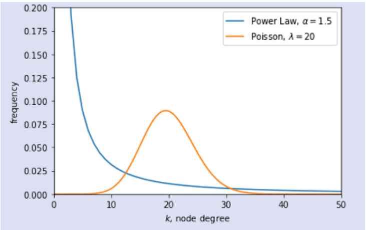


Figure 1: Illustration of the Erdős-Rényi model showing two random graphs with different probabilities p.

Degree Distribution

The degree distribution is a crucial property of networks, detailing how the degrees (number of connections) of nodes are distributed across the network.

- **Bell-Curved Distribution (Normal/Poisson/Binomial):** This is typical for random graphs like those in the Erdős-Rényi model, where most nodes have around the average number of connections.

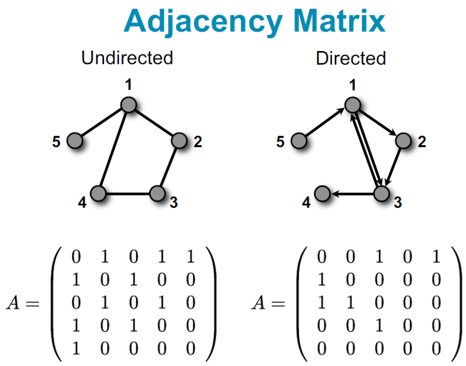
- **Scale-Free Distribution:** Characterized by a power-law distribution, indicating that while many nodes have few connections, a small number have many connections (hubs). This distribution is common in many real-world networks and is not typically observed in random graphs.

- **Interpretation:** The scale-free nature indicates that hubs are an inherent feature, contrasting with the more uniform distribution of connections in random graphs.

Adjacency Matrix

An adjacency matrix represents the connections between nodes in a graph. For undirected graphs, this matrix is symmetric because if node i is connected to node j, then node j is also connected to node i. The matrix for a directed graph may not be symmetric, reflecting the directionality of the connections.

- For an **undirected graph**:
  - The adjacency matrix is symmetric.
  - If A is the adjacency matrix, then $A_{ij} = A_{ji}$.
- For a **directed graph**:
  - The adjacency matrix is generally not symmetric.
  - The presence of an edge from i to j does not imply an edge from j to i.



$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \qquad A = \begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 2: Adjacency matrices for undirected and directed graphs.

Notation for a Graph

A graph G is defined as a set of nodes (or vertices) V and a set of links (or edges) E, expressed as $G = (V, E)$.

- V: nodes or vertices.
- E: links or edges.
- $|V|$: denotes the number N of vertices in the graph.
- $|E|$: denotes the number L of edges in the graph.

In an undirected graph, each edge is bidirectional and the adjacency matrix is symmetric. In a directed graph, edges are unidirectional, which may lead to an asymmetric adjacency matrix.

Subgraphs

Given a graph $G = (V, E)$, a subgraph induced by a subset of nodes $S \subseteq V$ is denoted as $G' = (S, F)$ and is defined by:

- The nodes in S.
- The edges in F such that every edge $(u, v) \in F$ has both u and v in S.

Degree of Nodes

The degree of a node in a graph is the number of connections it has to other nodes. In an undirected graph, the degree is simply the number of edges connected to the node. In a directed graph, each node has an in-degree (number of incoming edges) and an out-degree (number of outgoing edges). The degree distribution across the entire graph can reveal important properties about the network's structure and dynamics.

Degree in Graphs

In an undirected graph, the degree k_i of a node i is the number of links incident on that node. The total number of links L in the graph can be computed as:

$$L = \frac{1}{2} \sum_{i=1}^{N} k_i$$

This is because each edge is counted twice, once for each node it connects. The average degree $\langle k \rangle$ of the graph is then given by:

$$\langle k \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i = \frac{2L}{N}$$

This average is a basic characteristic of the network's connectivity.

In-Degree and Out-Degree in Directed Graphs

In directed graphs, we differentiate between in-degree k_i^{in} and out-degree k_i^{out} for a node i. The in-degree counts incoming links, while the out-degree counts outgoing links. The degree k_i for node i in a directed graph is the sum of both:

$$k_i = k_i^{in} + k_i^{out}$$

The total number of links L in a directed graph is counted by summing either the in-degrees or out-degrees of all nodes:

$$L = \sum_{i=1}^{N} k_i^{in} = \sum_{i=1}^{N} k_i^{out}$$

This total link count helps understand the overall density and structure of the network.

Degree Distribution

The degree distribution in a graph gives us insight into the connectivity of the nodes. It is defined as follows:

- If N_k is the number of nodes with degree k, then the degree distribution p_k is the probability that a randomly selected node has degree k. This is given by the formula:

  $$p_k = \frac{N_k}{N}$$

  where N is the total number of nodes in the graph.

- The average degree $\langle k \rangle$ of the graph is then calculated by summing over all possible degrees, weighted by their probabilities:

  $$\langle k \rangle = \sum_{k=0}^{\infty} k p_k$$

This measure provides an average level of connectivity of the nodes within the graph and is a fundamental characteristic of network structure.

Degree Distribution in Real Graphs

The degree distribution is a way to understand the connectivity of a network by looking at how many connections each node has. In real-world networks, this distribution often reveals a mix of nodes with few connections and a few nodes with many connections, known as 'hubs'.

We typically visualize degree distribution using two types of plots:

- A **linear scale plot** shows the proportion of nodes for each degree value, providing a straightforward view of the distribution.

- A **log-log scale plot** is used to highlight features of the distribution that follow a power law, often found in large networks. This type of plot can make it easier to see whether a network has a scale-free structure, where the distribution of nodes' connections creates a straight line when plotted on logarithmic scales.

The graphical representation of the network and its degree distribution help us to quickly identify key characteristics such as the presence of hubs and the network's overall connectivity pattern.

Understanding the Adjacency Matrix

An adjacency matrix A represents the connections between nodes in a graph $G = (V, E)$. It is a square matrix with dimensions equal to the number of nodes $|V|$. The elements of A indicate whether pairs of vertices are adjacent or not in the graph:

- $A_{ij} = 1$ if there is an edge (i, j) in E, meaning nodes i and j are connected.

- $A_{ij} = 0$ if there is no edge (i, j) in E, meaning nodes i and j are not connected.

In matrix notation, A_{ij} always refers to the element in row i and column j.

Properties of Adjacency Matrices

The structure of an adjacency matrix reveals certain characteristics of the graph it represents:

- If G is undirected, then A is symmetric because edges have no direction—$A_{ij} = A_{ji}$.

- If G has a self-loop, meaning an edge that connects a vertex to itself, there will be a non-zero element in the diagonal of A—$A_{ii} \neq 0$.

- G is a complete graph if there is an edge between every pair of distinct vertices. The adjacency matrix A of a complete graph has all non-zero elements except for the diagonal if self-loops are not allowed—$A_{ij} \neq 0$ for all $i \neq j$.

Understanding these properties is vital for interpreting the adjacency matrix and the corresponding graph structure.

Subnetworks and Cliques

- A subnetwork is a graph formed from a subset of nodes and all the links among these nodes in the original network.

- A clique is a special kind of subnetwork; it's a complete subnetwork, meaning all its nodes are connected to each other.

- An n-clique, or a complete graph with n nodes, is one where every node is connected to every other node.

- A bipartite clique is a complete bipartite graph, divided into two subsets of vertices, where each vertex from one subset is connected to every vertex in the other subset.

Cliques and Bi-partite Cliques

A clique in a graph represents a subset of vertices that are all directly connected to each other, forming a complete subgraph. Specifically:

- A clique is a complete graph such that every two distinct vertices are connected by a unique edge. This can be written as $E = V \times V$, meaning each vertex in set V is connected to every other vertex.

- An n-clique is a complete graph consisting of n nodes, where every node is connected to every other node within the clique.
- A bi-partite clique, also known as a complete bi-partite graph, is a special type of graph where vertices can be divided into two disjoint sets V_1 and V_2 such that there are no edges between vertices of the same set, and every vertex of set V_1 is connected to every vertex of set V_2. The notation $E = (V_1 \times V_2)$ reflects this complete interconnection between the two sets.
- A (n_1, n_2)-clique is a bi-partite clique where the sizes of the two vertex sets are n_1 and n_2 respectively. Here, $|V_1| = n_1$ and $|V_2| = n_2$, indicating the number of vertices in each part of the bi-partite graph.

Such structures are important in graph theory for understanding the maximal subgraph of fully connected nodes and have applications in network analysis, such as finding communities or dense connections within a network.

Sparsity in Real Networks

Real-world networks tend to be sparse. This means that the number of actual links L is much less than the maximum number of possible links L_{max}.

- Theoretically, the maximum number of links L_{max} in an undirected graph without self-loops is given by the binomial coefficient:

  $$L_{max} = N2 = \frac{N(N-1)}{2}$$

  where N is the total number of nodes in the graph.
- Sparsity is characterized by $L \ll L_{max}$, indicating that nodes have fewer connections than theoretically possible.

The concept of sparsity is important in understanding the structure and complexity of a network. Sparse networks often exhibit interesting patterns such as community structure, and the presence of hubs can significantly affect their dynamics and function.

Degree of Separation and Body Size

This graph illustrates an intriguing relationship between social connectivity and health outcomes. It suggests that an individual's likelihood of being obese can be related to their position within a social network. Here, the 'degrees of separation' refers to the social distance between individuals.

- A direct connection (1 degree of separation) to an obese person increases one's chances of obesity significantly.
- As the social distance increases (2, 3, or more degrees of separation), the influence on one's probability of being obese diminishes.

This concept underlines the impact of social factors on health behaviors and outcomes. It also highlights the importance of considering social network structures in public health interventions.

Network Distributions: Bell Curve vs. Power Law

The two graphs represent different types of network distributions. The Bell Curve, or Gaussian distribution, is typically found in random networks. Here, most nodes have approximately the same number of links, and there are hardly any nodes that are significantly more connected than average.

In contrast, the Power Law Distribution is characteristic of scale-free networks. These networks have many nodes with only a few links and a small number of highly connected nodes known as hubs. This distribution indicates that in such networks, hubs play a critical role in the network's connectivity.

- The **Bell Curve** represents homogeneity in node connectivity.
- The **Power Law Distribution** reflects heterogeneity, pointing to an inequality in the distribution of connections.

Understanding these distributions is vital for analyzing the robustness and vulnerability of networks, as well as their growth mechanisms and dynamic processes.

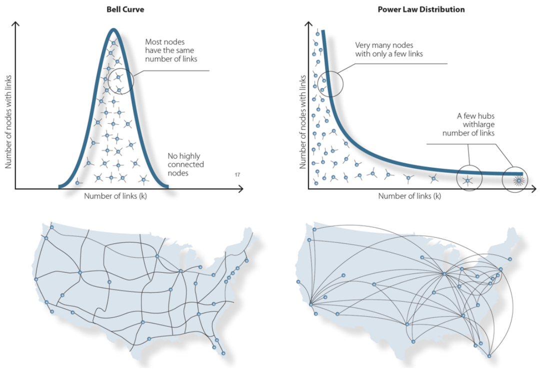

Bell Curve
Most nodes have the same number of links
Number of nodes with links
No highly connected nodes
17
Number of links (k)
Power Law Distribution
Very many nodes with only a few links
Number of nodes with links
A few hubs withlarge number of links
Number of links (k)

Figure 3: Bell Curve and Power Law

Confounding of Homophily and Contagion in Social Networks

Homophily refers to the tendency of individuals to associate and bond with similar others, while contagion implies a direct influence between individuals in a network. In observational social network studies, distinguishing between homophily and contagion effects can be challenging for several reasons:

- **Selection Bias:** Individuals may choose relationships based on shared characteristics or preferences, leading to homophily. However, this can be mistaken for contagion as those individuals may also influence each other over time.
- **Simultaneity:** The co-occurrence of homophily and contagion can happen over the same period, making it difficult to disentangle the direction of the relationship between two correlated behaviors.
- **Unobserved Confounding:** There may be unmeasured variables that influence both the formation of ties (homophily) and the behaviors or attributes that are seemingly 'contagious'.

- **Dynamic Network Structures:** The evolving nature of social networks adds complexity to causal inference since the network structure and individual behaviors are constantly changing.

- **Attribution Errors:** Researchers may incorrectly attribute the similarity in behaviors among connected individuals to social influence when it could be due to their pre-existing similarities (homophily).

Understanding these challenges is crucial in designing robust social network studies and accurately interpreting the mechanisms underlying observed patterns of behaviors and attributes.

Assortativity in Networks

Assortativity is the preference for a network's nodes to attach to others that are similar in some way. This phenomenon can arise due to two key mechanisms:

1. **Selection or Homophily:** Nodes may choose to create links with other nodes that are similar, leading to a network where connections exist predominantly among similar nodes.

2. **Social Influence:** Nodes within a network may become more similar to one another over time due to the influence of existing connections, further increasing similarity across links.

While assortativity can foster strong communities and facilitate communication among similar nodes, it can also lead to negative outcomes such as the formation of ”echo chambers.” This happens when nodes are only exposed to ideas and opinions that reinforce their existing beliefs, potentially reducing diversity and the opportunity for exposure to different viewpoints.

Degree Assortativity in Networks

Degree assortativity, also known as degree correlation, refers to a tendency of nodes in a network to connect to others with a similar degree. In an assortative network, nodes with a high degree (hubs) are likely to be connected to other nodes with high degrees, creating a core-periphery structure. Social networks often exhibit this pattern.

On the other hand, disassortative networks, such as the Web, Internet, food webs, and biological networks, tend to have a hub-and-spoke structure where high-degree nodes are connected to many low-degree nodes. This kind of network layout ensures that there are central points of control or communication, which can be critical for the network's function.

Assortativity in NetworkX

In NetworkX, a Python library for studying graphs and networks, assortativity can be measured using different methods depending on the attribute of interest. The assortativity coefficient can be calculated for categorical attributes, such as gender, or for numerical attributes, like age. Additionally, degree assortativity, which is based on the Pearson correlation between degrees of adjacent nodes, can also be evaluated. Below are examples of how to compute each type of assortativity in NetworkX:

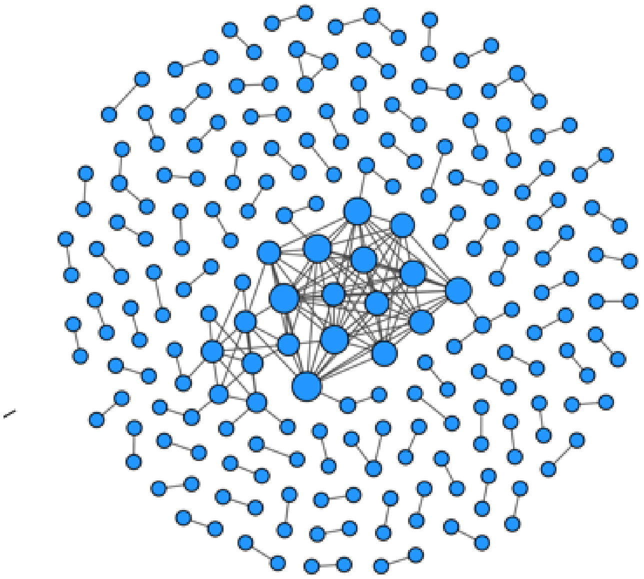


Figure 4: An example of an assortative network with a core-periphery structure and a disassortative network with a hub-and-spoke structure.

```
# based on a categorical attribute, such as gender
assort_a = nx.attribute_assortativity_coefficient(
    G, category)

# based on a numerical attribute, such as age
assort_n = nx.numeric_assortativity_coefficient(
    G, quantity)

# based on degree (Pearson correlation between degree of
# adjacent nodes)
r = nx.degree_assortativity_coefficient(G)
```

The resulting values give an indication of the tendency of nodes to connect with other nodes that are similar (assortative mixing) or different (disassortative mixing) in terms of the given attribute.

Assortativity in NetworkX

Degree assortativity can also be assessed by looking at the correlation between a node's degree and the average degree of its neighbors. This measure provides an insight into how nodes in a network preferentially attach to others with similar or dissimilar degree values. The average nearest-neighbor degree for nodes of degree k can be calculated using NetworkX functions as follows:

$$k_{nn}(i) = \frac{1}{k_i} \sum_j a_{ij} k_j \tag{1}$$

$$\langle k_{nn}(k) \rangle = \langle k_{nn}(i) \rangle_{k(i)=k} \tag{2}$$

Where a_{ij} represents the adjacency matrix of the network, k_i is the degree of node i, and k_j is the degree of node j. The code snippet to compute this in Python using NetworkX and SciPy is:

```
import scipy.stats
knn_dict = nx.k_nearest_neighbors(G)
k, knn = list(knn_dict.keys()), list(knn_dict.values())
r, p_value = scipy.stats.pearsonr(k, knn)
```

This code calculates the nearest neighbor average degree for each node and then computes the Pearson correlation coefficient to quantify the degree of assortativity.

Echo Chamber

Finally, this subsection could focus on the echo chamber effect in social networks, describing how it is related to the concepts of homophily and assortativity. It could discuss the implications of echo chambers in the context of information spread and opinion formation within networks.

Phenomena of Social Connectivity

Social networks exhibit fascinating phenomena regarding the connections and influence among individuals. Here we explore a few key concepts:

- **Degree of separation:** This is the measure of social distance between two people within a network. It is quantified by the number of steps required to connect one person (the ego) to another through acquaintances (alters). For example, a friend of a friend would have a degree of separation of two.
- **Six degrees of separation:** This concept postulates that any two people on Earth can be connected through, at most, six layers of social connections. It highlights the surprisingly small world we live in, suggesting that extensive global connectivity is possible through a limited number of social links.
- **Three degrees of influence:** This principle suggests that our actions or behaviors can influence not only our direct friends but also our friends' friends, and their friends in turn. The ripple effect of our social influence is thus thought to extend to three degrees of separation.

These concepts illustrate the intricate web of interconnectivity in social structures and the potential for widespread influence through seemingly distant social ties.

The Concept of Small Worlds in Social Networks

Social networks have revealed a surprising phenomenon about human connections: they are quite close-knit. This observation is crystallized in the concept known as the **six degrees of separation**:

- The theory suggests that any two individuals on Earth are separated by no more than six acquaintances.
- The idea has literary roots, first appearing in the 1929 short story ”Chains” by the Hungarian writer Frigyes Karinthy.
- It gained scientific backing through the work of psychologist Stanley Milgram, who, in 1967, conducted an experiment that provided the first empirical evidence supporting the small world theory. Milgram's study sought to measure the social distance between people in the United States, finding that, on average, chains of acquaintance were surprisingly short.
- The term ”six degrees of separation” was later popularized by John Guare in his 1991 play, embedding the concept into popular culture and emphasizing the compactness of social networks.

The recognition of such small degrees of separation has significant implications for understanding the spread of information and the potential for influence across large networks.

0.1 Paths: Definitions

- **Path**: A sequence of links traversed to go from a source to a target node.
  - In a directed network, links must be traversed according to their direction.
  - There may not be a path.
- **Cycle**: A path where source and target node are the same.
- **Simple path**: A path with no traversing the same link more than once.
  - Only simple paths will be dealt with.
- **Path length**: The number of links in the path.

0.2 Euler circa 1736: Königsberg Bridges

Question: Can you cross all 7 bridges just once each?

Answer: No. Euler showed that if more than two nodes have an odd degree, there is no Eulerian path that crosses every bridge exactly once. In the case of Königsberg, each of the four land masses was connected to an odd number of bridges, making it impossible to achieve the task.

0.3 Shortest Paths

The concept of the shortest path is integral in the study of network topologies. A shortest path between two nodes is defined by the path that has the minimal total length, with the length often representing the number of edges in unweighted networks or the sum of edge weights in weighted networks.

- In unweighted networks, the shortest path is the one with the least number of hops.
- In weighted networks, each edge has a numerical value (weight), and the shortest path minimizes the total sum of these weights.
- The shortest path length or distance is the metric used to quantify the efficiency of the path.

- If no path exists between two nodes, the shortest path is said to be undefined or infinitely long.

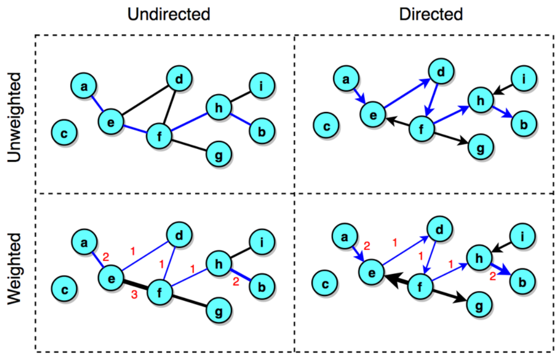


Figure 5: Illustrative examples of shortest paths in directed and undirected networks, both unweighted and weighted.

The figure shows examples of shortest paths in different types of networks. Note that the complexity of finding the shortest path can vary significantly depending on the network's structure and whether it is weighted or unweighted.

0.4 Average Path Length and Diameter

The structural characteristics of a network can be described using the shortest paths between nodes. Specifically, two metrics are of interest:

- **Diameter:** This is defined as the longest shortest-path length found within the network. Mathematically, it is represented as:

$$diameter = \max_{i,j} \ell_{ij}$$

  where ℓ_{ij} denotes the shortest-path length between nodes i and j.

- **Average Path Length (APL):** The APL represents the average number of steps along the shortest paths for all possible pairs of network nodes. It is calculated differently for undirected and directed networks:

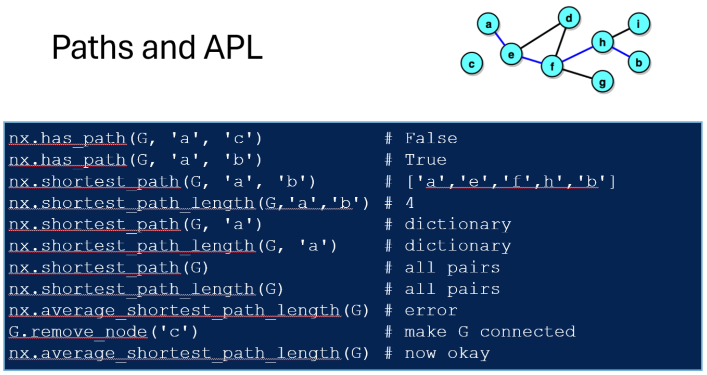
  - For undirected networks:

$$\langle \ell \rangle = \frac{\sum_{i,j} \ell_{ij}}{N2} = \frac{2\sum_{i,j} \ell_{ij}}{N(N-1)}$$

  - For directed networks:

$$\langle \ell \rangle = \frac{\sum_{i,j} \ell_{ij}}{N(N-1)}$$

The diameter gives us an idea of the "size" of the network, while the APL offers insight into the "tightness" of the network connectivity.



Paths and APL

```
nx.has_path(G, 'a', 'c')              # False
nx.has_path(G, 'a', 'b')              # True
nx.shortest_path(G, 'a', 'b')         # ['a','e','f',h','b']
nx.shortest_path_length(G,'a','b')    # 4
nx.shortest_path(G, 'a')              # dictionary
nx.shortest_path_length(G, 'a')       # dictionary
nx.shortest_path(G)                   # all pairs
nx.shortest_path_length(G)            # all pairs
nx.average_shortest_path_length(G)    # error
G.remove_node('c')                    # make G connected
nx.average_shortest_path_length(G)    # now okay
```

0.5 Assortativity in NetworkX

In NetworkX, a Python library for studying graphs and networks, assortativity can be measured using different methods depending on the attribute of interest. The assortativity coefficient can be calculated for categorical attributes, such as gender, or for numerical attributes, like age. Additionally, degree assortativity, which is based on the Pearson correlation between degrees of adjacent nodes, can also be evaluated. Below are examples of how to compute each type of assortativity in NetworkX:

```
# based on a categorical attribute, such as gender
assort_a = nx.attribute_assortativity_coefficient
    (G, category)

# based on a numerical attribute, such as age
assort_n = nx.numeric_assortativity_coefficient
    (G, quantity)

# based on degree (Pearson correlation between
    degree of adjacent nodes)
r = nx.degree_assortativity_coefficient(G)
```

The resulting values give an indication of the tendency of nodes to connect with other nodes that are similar (assortative mixing) or different (disassortative mixing) in terms of the given attribute.

0.6 Connectedness and Components

A network is considered connected if there is a path between any two nodes within it. This connectivity is fundamental to the network's functionality, as it allows for communication or flow between different parts of the network.

- If a network lacks paths between certain nodes, it is deemed to be **disconnected** and is said to contain multiple connected components.

- A **connected component** is a standalone subnetwork where there is a path between any two nodes within the component.

  - The largest connected component in a network is often referred to as the **giant component**. This component generally comprises a significant portion of the entire network.

  - A **singleton** represents the smallest possible connected component, typically consisting of a single isolated node.

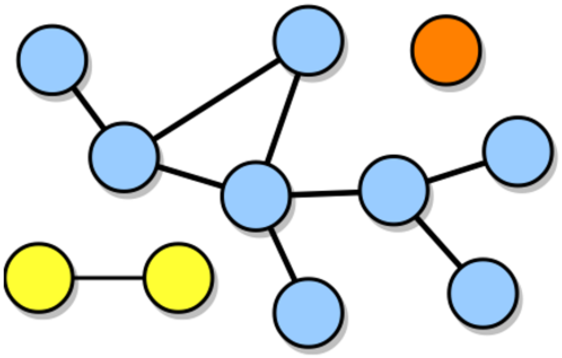


Figure 6: A visual representation of network connectedness and components.

0.7 Connectedness and Components

- A **network is connected** if there is a path between any two nodes.
- If a network is **not connected**, it is **disconnected** and consists of multiple connected components.
- A **connected component** is a connected subnetwork.
  - The **largest one** is termed the **giant component** and usually comprises a significant portion of the network.
  - A **singleton** is the smallest possible connected component.

0.8 Connectedness and Components

- We use NetworkX functions to evaluate the connectivity of a network.
- Different functions allow us to assess various aspects of connectedness.

The following Python code snippet demonstrates these checks using NetworkX:

```
# Check if the graph G is connected
nx.is_connected(G)

# Get the connected components sorted by size
comps = sorted(nx.connected_components(G),
               key=len, reverse=True)

# Find the nodes in the largest connected component
nodes_in_giant_comp = comps[0]

# Create a subgraph for the giant component
GC = nx.subgraph(G, nodes_in_giant_comp)

# Check if the giant component is connected
nx.is_connected(GC)

# Check for strong and weak connectivity
nx.is_strongly_connected(D)
nx.is_weakly_connected(D)

# Get lists of weakly and strongly connected components
list(nx.weakly_connected_components(D))
list(nx.strongly_connected_components(D))
```

The code outputs various boolean values and lists indicating the connectivity status of the network and its components.

Figure 7: Visualization of connected components in a network.

0.9 Trees in Networks

A tree is a special type of graph that has important properties and applications in network theory:

- It is a **connected network without cycles**, meaning there is exactly one path between any two nodes.
- A tree with N nodes always contains $N - 1$ links.
- Trees are inherently **hierarchical**. A root node can be designated, with each subsequent node having exactly one parent node and potentially multiple children nodes.
- The **root node** is the only node without a parent.
- **Leaf nodes**, or leaves, are nodes without children.

The structure of a tree lends itself to efficient organization and retrieval of data, as seen in binary search trees, file systems, and phylogenetic trees.

Figure 8: A tree structure with nodes and hierarchical relationships.

0.10 Trees and Graph Types in NetworkX

In the context of NetworkX, a library for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks, different types of graphs can be analyzed to determine whether they constitute a tree. A tree is a special form of a graph that is connected and acyclic. Here we evaluate different graphs:

- A **complete graph** K_4 is not a tree because it contains cycles.
- A **bipartite graph** B and a **cycle graph** C are also not trees due to the presence of cycles.
- On the other hand, a **star graph** S and a **path graph** P are both trees because they are connected and do not contain any cycles.

The following code snippet checks the tree property for various graphs:

```
K4 = nx.complete_graph(4)
nx.is_tree(K4)  # False

B = nx.complete_bipartite_graph(4,5)
nx.is_tree(B)  # False

C = nx.cycle_graph(4)
nx.is_tree(C)  # False

S = nx.star_graph(6)
nx.is_tree(S)  # True

P = nx.path_graph(5)
nx.is_tree(P)  # True
```

0.11 Finding Shortest Paths

The algorithm employed to ascertain the shortest paths within a network is known as the **breadth-first search** (BFS). This process initiates from a designated source node, often referred to as the root, and proceeds as follows:

- Commence at the source node and explore the entirety of the network's breadth, up to a certain distance from the source node.
- Prioritize visiting nodes that are closer (i.e., within a smaller breadth) to the source before moving on to nodes that are further away (i.e., at a greater depth).

An alternative approach is to begin from each individual node and determine all possible shortest paths between pairs of nodes. This method, typically known as the all-pairs shortest path algorithm, can be computationally intensive for large networks, with a time complexity on the order of $O(N^2)$, where N represents the number of nodes in the network.

0.12 Characteristics of Small-world Networks

Small-world networks are a type of mathematical model that describe a type of network architecture. They are characterized by:

- **High Clustering:** Nodes tend to create tightly knit groups characterized by a relatively high density of edges; this is more evident than in random networks.
- **Short Path Lengths:** Despite the high clustering, the path lengths between any two nodes in the network tend to be short, similar to those in random networks.

These properties are a result of an architecture that includes both local clustering like regular lattices and the short path lengths typical of random graphs. This balance is often described by the parameter p in network models, which represents the probability of rewiring each edge in a regular lattice network. When $p = 0$, the network is a regular lattice, and when $p = 1$, the network is completely random. The small-world property emerges for intermediate values of p, giving the network both local and global efficiency for information transfer.

0.13 Clustering Coefficient

The clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together. For a given node, the clustering coefficient is defined as the proportion of links between the nodes within its neighborhood divided by the number of links that could possibly exist between them. Formally, the clustering coefficient $C(i)$ for a node i is given by:

$$C(i) = \frac{2\tau(i)}{k_i(k_i - 1)} \tag{3}$$

where $\tau(i)$ is the number of triangles involving the node i, and k_i is the degree of i. The term $k_i(k_i - 1)/2$ represents the maximum number of triangles that could involve i, assuming every neighbor of i is connected to every other neighbor. The clustering coefficient is undefined for nodes with degrees of less than 2, as such nodes cannot form triangles. In the context of the NetworkX library, a node with no triangles is assigned a clustering coefficient of 0.

0.14 Network Clustering Coefficient

The clustering coefficient of the network is the average of the clustering coefficients of the nodes:

$$C = \frac{\sum_{i;k_i>1} C(i)}{N_{k>1}}$$

Again, we should exclude singletons and nodes with $k = 1$, but NetworkX assumes those have $C = 0$.

```
# dict node -> no. triangles
nx.triangles(G)
# clustering coefficient of node
nx.clustering(G, node)
# dict node -> clustering coefficient
nx.clustering(G)
# network's clustering coefficient
nx.average_clustering(G)
```

0.15 Network Clustering Coefficient

Some networks, for instance, social networks, tend to have high clustering coefficients due to triadic closure, meaning we often meet new people through common friends. Conversely, other types of networks such as bipartite or tree-like structures exhibit low clustering coefficients, indicating a different pattern of connectivity.

Triadic Closure

Triadic closure is a fundamental concept in social networks which postulates that if a person A is strongly connected to both persons B and C, there is a high likelihood that persons B and C will also be connected. In network terminology, if a node A is connected to both nodes B and C, then there is a high probability that B and C are also connected, forming a triangle within the network.

Clustering in Different Network Types

The table below provides examples of various networks and their clustering coefficients, demonstrating the prevalence of triadic closure in social networks in comparison to other types of networks:

Note: In the table, $N_k > 1$ denotes the number of nodes with more than one connection, which is used in the calculation of the network's clustering coefficient. It's important to exclude singletons (nodes with $k = 1$) as they do not contribute to the formation of triangles.

0.16 Local Clustering Coefficient

The local clustering coefficient (LCC) is a measure used to evaluate how close a node's neighbors are to being a complete graph or clique. The LCC for a node is defined as the number of links between the nodes within its neighbourhood divided by the number of links that could possibly exist between them.

Example Calculation

Given a node with k neighbours, the number of possible links between the neighbours is $k(k-1)/2$. The LCC is the ratio of the actual number of links to the possible number.

Note: The data included here is illustrative. In the actual LaTeX document, the data needs to be filled in as per the specific details of the network being analyzed.

0.17 Centrality Measures

Centrality measures are important to understand various aspects of a node's position within a network. They answer questions related to a node's importance or influence.

- **Indegree Centrality**: This represents how many people know or are connected to a particular node. It can be a measure of popularity or recognition within the network.
- **Outdegree Centrality**: This reflects how many people a node knows or to how many others it is connected. It can indicate a node's level of activity or engagement within the network.

Centrality can be understood as a local measure, indicating the immediate influence or reach of a node, often visualized by the number of direct connections (edges) it has.

0.18 Degree Centrality

Degree centrality measures the importance of a node within the network based on the number of connections it has. The degree centrality $C^D(i)$ for a node i is given by the formula:

$$C^D(i) = \frac{k_i}{N-1}$$

where k_i is the degree of node i, and N is the total number of nodes in the network. Nodes with higher degree centrality are considered more central in the network as they have a larger number of connections to other nodes. This measure is often used to identify influential individuals in social networks or key infrastructure components in transportation and communication networks.

0.19 Betweenness Centrality

Betweenness centrality is a measure of centrality in a network that indicates the importance of a node as a bridge along the shortest path between two other nodes. It is defined as:

$$\tilde{C}_B(i) = \sum_{j<k} \frac{d_{jk}(i)}{d_{jk}} \tag{4}$$

where d_{jk} is the number of shortest paths between nodes j and k, and $d_{jk}(i)$ is the number of those paths that pass through node i. This measure helps to identify the nodes that most frequently act as bridges in the network communication.

0.20 Closeness Centrality

Closeness centrality is a way of identifying nodes that can quickly interact with all others in the network due to their location. It is inversely related to the average distance from a node to all other nodes in the network. The closeness centrality of node i is defined as the reciprocal of the sum of the shortest path lengths from i to all N nodes in the network:

$$\tilde{C}_C(i) = \left[\sum_{j=1}^{N} d(i,j) \right]^{-1} \tag{5}$$

where $d(i,j)$ is the shortest path length between nodes i and j. The normalized closeness centrality for a node takes the total number of nodes into account and is given by:

$$C_C(i) = \frac{\tilde{C}_C(i)}{N-1} \tag{6}$$

This measure emphasizes the notion that a node is more central if it is closer to all other nodes, thereby reducing the path lengths for communication or transfer across the network.

0.21 Eigenvector Centrality

Eigenvector centrality is a measure of the influence of a node within a network. It assigns relative scores to all nodes in the network based on the principle that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes.

The eigenvector centrality x_i for node i is given by the formula:

$$x_i = \frac{1}{\lambda} \sum_{j \in \Lambda(i)} x_j \tag{7}$$

In matrix terms, this can be rewritten as:

$$x_i = \frac{1}{\lambda} \sum_{j \in G} a_{ij} x_j \tag{8}$$

where λ is a constant, $\Lambda(i)$ is the set of neighbors of node i, a_{ij} are the entries of the adjacency matrix of the network, and G represents the entire set of nodes. This can be simplified to the matrix equation:

$$AX = \lambda X \tag{9}$$

where A is the adjacency matrix of the network, X is the eigenvector of centrality scores, and λ is the largest eigenvalue of A. This method ensures that nodes are considered influential not just by the number of their connections, but also by the significance of the nodes they are connected to.