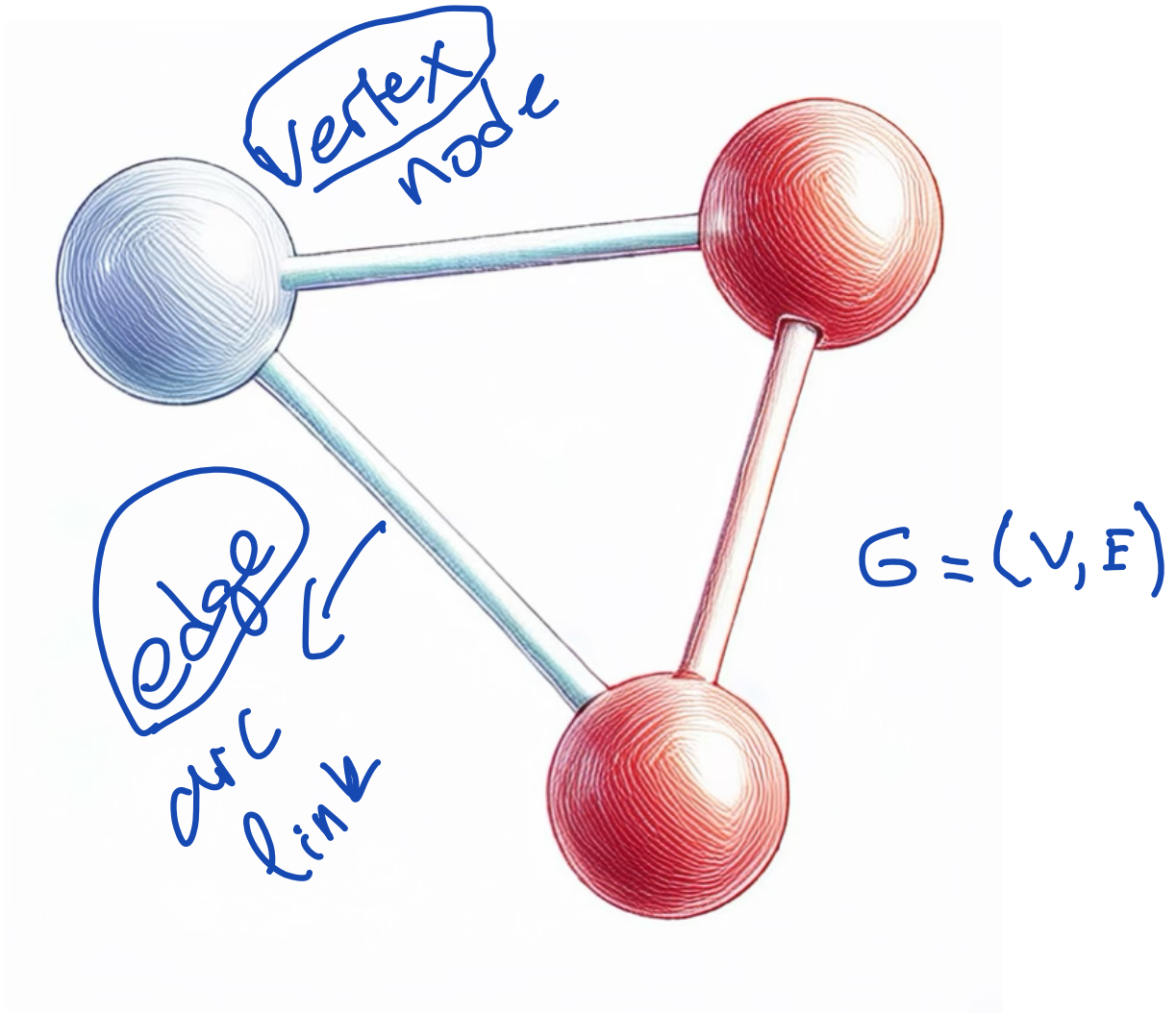


## Introduction to Network Science 8.26.25



- Introduction to Network Science 8.26.25
  - Overview
  - Learning Objectives
  - What is Network Science?
    - Definition
    - Key Concepts
    - Historical Context

- Types of Networks
  - Undirected Networks
  - Directed Networks
  - Weighted Networks
  - Bipartite Networks
  - Temporal Networks
- Simple Graph Examples
  - Example 1: A Basic Undirected Graph
  - Example 2: A Directed Graph (Digraph)
- More Complex Graph Structures
  - Example 3: A Social Network
  - Example 4: A Transportation Network
- Graph Properties and Terminology
  - Degree of a Node
  - Paths and Cycles
  - Connected vs. Disconnected Graphs
- Special Types of Graphs
  - Tree Structure
  - Complete Graph
  - Weighted Graphs
- Basic Network Properties
  - Size Metrics
  - Connectivity
  - Degree Distribution
  - Path Properties
- Network Models
  - Random Networks (Erdős-Rényi)
  - Small-World Networks (Watts-Strogatz)
  - Scale-Free Networks (Barabási-Albert)
- Real-World Network Examples
  - Social Networks
  - Biological Networks
  - Technological Networks

- Information Networks
- Network Analysis Workflow
  - Step 1: Data Collection
  - Step 2: Network Construction
  - Step 3: Basic Analysis
  - Step 4: Advanced Analysis
  - Step 5: Visualization and Interpretation
- Key Network Metrics
  - Centrality Measures
  - Clustering
  - Assortativity
- Applications and Impact
  - Social Sciences
  - Biology and Medicine
  - Technology and Engineering
  - Economics and Finance
- Tools and Software
  - Python Libraries
  - Specialized Software
- Challenges and Future Directions
  - Data Challenges
  - Methodological Challenges
  - Emerging Areas
- Summary
- Key Takeaways
- Key Insights for Learning
- Practice Exercise
- Next Steps

## Overview

field

Network Science is the study of complex systems represented as networks (graphs) where entities are nodes and relationships are edges. This field combines elements from mathematics, computer science, physics, and social sciences to understand how interconnected systems behave.

math

## Learning Objectives

By the end of this lecture, you will be able to:

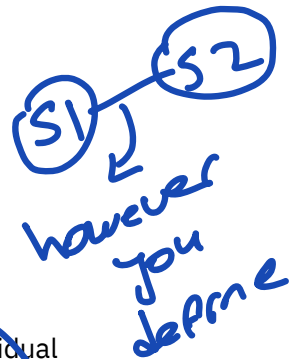
- Understand basic network science concepts and terminology
- Identify different types of networks and their properties
- Analyze fundamental network metrics
- Recognize real-world applications of network science

## 1. What is Network Science?

### Definition

Network Science is the study of complex systems as networks, where:

- **Nodes** (vertices) represent entities (people, computers, proteins, etc.)
- **Edges** (links) represent relationships or interactions between entities



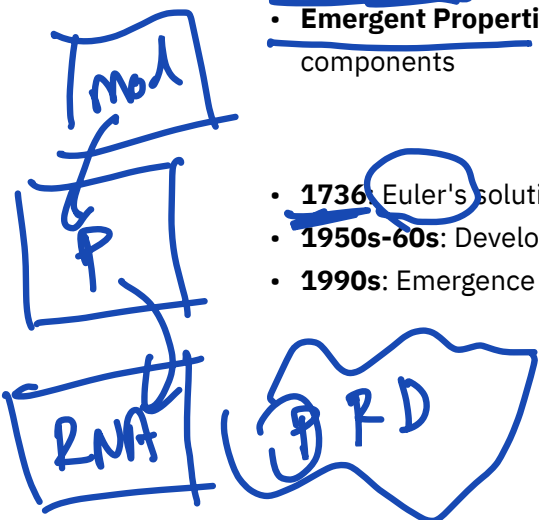
### Key Concepts

- **Graph Theory**: Mathematical foundation for network analysis
- **Complex Systems**: Systems with many interacting components
- **Emergent Properties**: Behaviors that arise from interactions, not individual components

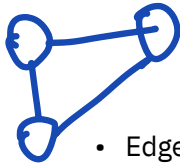
### Historical Context

- **1736**: Euler's solution to the Königsberg bridge problem
- **1950s-60s**: Development of graph theory in mathematics
- **1990s**: Emergence of "small-world" and "scale-free" networks

network



- **2000s:** Network Science as a distinct interdisciplinary field



## 2. Types of Networks

### Undirected Networks

- Edges have no direction (e.g., friendship networks)
- Symmetric relationships
- Example: Facebook friendships

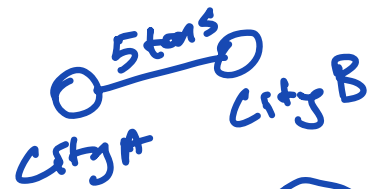
### Directed Networks

- Edges have direction (e.g., following on social media)
- Asymmetric relationships
- Example: Twitter followers



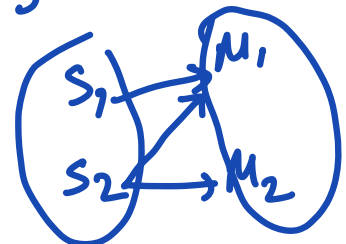
### Weighted Networks

- Edges have weights (e.g., communication frequency)
- Quantitative relationships
- Example: Email communication volume



### Bipartite Networks

- Two types of nodes
- Edges only between different types
- Example: Users and movies in recommendation systems



*Dynamic*

### Temporal Networks

- Networks that change over time
- Dynamic relationships
- Example: Communication networks over time

*edges belong to unique sets*

## Simple Graph Examples

### Example 1: A Basic Undirected Graph

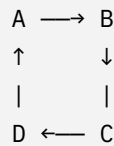




**Nodes:** A, B, C, D **Arcs:** A-B, B-C, C-D, D-A

This creates a square where each vertex connects to its neighbors. Notice that in an undirected graph, we can travel from any connected node to another in either direction.

### Example 2: A Directed Graph (Digraph)

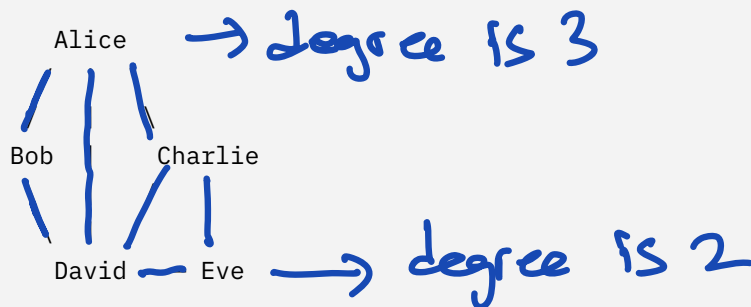


**Nodes:** A, B, C, D **Directed Arcs:** A→B, B→C, C→D, D→A

Here, the arrows show direction. You can go from A to B, but not necessarily from B to A without following the complete cycle.

## More Complex Graph Structures

### Example 3: A Social Network



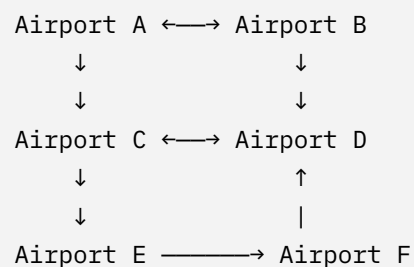
**Nodes:** Alice, Bob, Charlie, David, Eve **Relationships:**

- Alice knows: Bob, Charlie, David

- Bob knows: Alice, David
- Charlie knows: Alice, David, Eve
- David knows: Alice, Bob, Charlie, Eve
- Eve knows: Charlie, David

This represents a friendship network where Alice is highly connected (a hub), while Eve is more peripheral.

#### Example 4: A Transportation Network



#### Flight Routes:

- $A \leftrightarrow B$  (bidirectional flights)
- $A \rightarrow C$  (one-way route)
- $B \rightarrow D$  (one-way route)
- $C \leftrightarrow D$  (bidirectional flights)
- $C \rightarrow E$  (one-way route)
- $E \rightarrow F$  (one-way route)
- $F \rightarrow D$  (one-way route)

### Graph Properties and Terminology

#### Degree of a Node

The **degree** of a node is the number of arcs connected to it.

In our social network example:

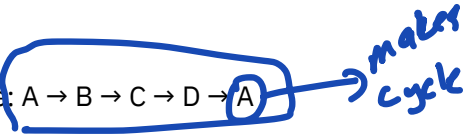
- Alice has degree 3 (connected to Bob, Charlie, David)

- David has degree 4 (connected to Alice, Bob, Charlie, Eve)
- Eve has degree 2 (connected to Charlie, David)

## Paths and Cycles

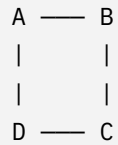
A **path** is a sequence of nodes connected by arcs. A **cycle** is a path that starts and ends at the same node.

Path example: Alice → David → Eve

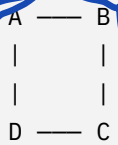


## Connected vs. Disconnected Graphs

Connected Graph:



Disconnected Graph:

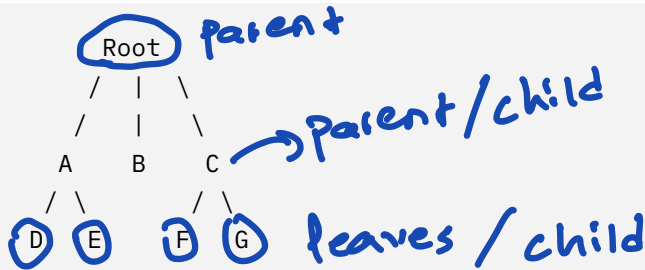


E — F

## Special Types of Graphs

### Tree Structure

A tree is a connected graph with no cycles - like a family tree or file system.

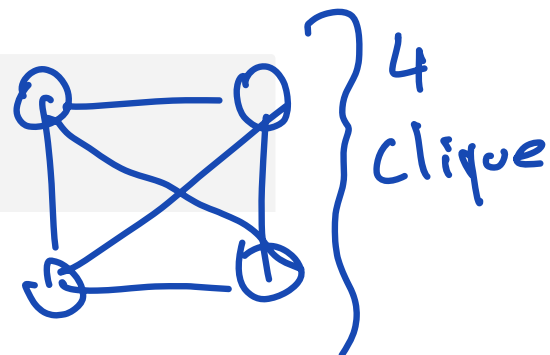


### Complete Graph

/ clique

A complete graph connects every node to every other node.

Complete Graph with 4 nodes:







### Weighted Graphs

Sometimes arcs have values (weights) representing cost, distance, or strength of connection.



The numbers represent distances, costs, or other measurable relationships.

### 3. Basic Network Properties

#### Size Metrics

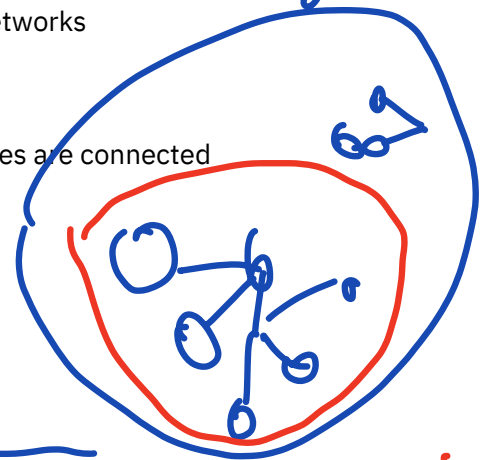
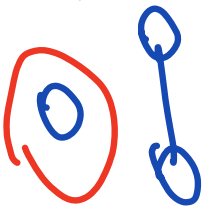
- **Number of nodes (N):** Total entities in the network
- **Number of edges (E):** Total relationships
- **Density:** Ratio of actual edges to possible edges
- Formula:  $\text{density} = 2E / (N \times (N-1))$  for undirected networks

#### Connectivity

- **Connected component:** Subset of nodes where any two nodes are connected by a path
- **Giant component:** Largest connected component
- **Isolated nodes:** Nodes with no connections

$$\frac{|E|}{\binom{N}{2}} = \frac{2E}{N \times (N-1)}$$

isolated node



Giant component

Order of a graph  
Size

### Degree Distribution

- **Degree:** Number of connections a node has
- **Average degree:** Mean number of connections per node
- **Degree distribution:** Distribution of degrees across all nodes

### Path Properties

- **Path:** Sequence of edges connecting two nodes
- **Shortest path:** Minimum number of edges between two nodes
- **Diameter:** Maximum shortest path length in the network
- **Average path length:** Mean shortest path length between all pairs

## 4. Network Models

### Random Networks (Erdős-Rényi)

- Each pair of nodes connected with probability  $p$
- Poisson degree distribution
- Low clustering coefficient

### Small-World Networks (Watts-Strogatz)

- High clustering like regular networks
- Short path lengths like random networks
- Created by rewiring regular networks

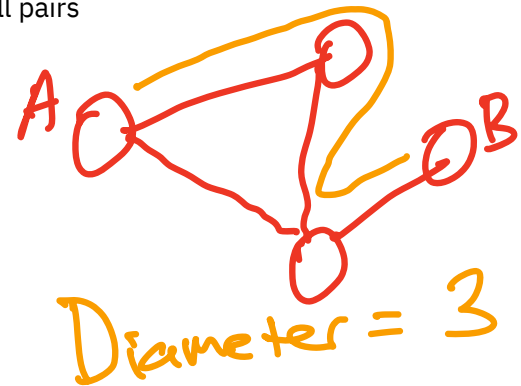
### Scale-Free Networks (Barabási-Albert)

- Power-law degree distribution
- Preferential attachment mechanism
- Few highly connected hubs, many poorly connected nodes

## 5. Real-World Network Examples

### Social Networks

- **Facebook:** Friendship networks
- **Twitter:** Follower networks
- **LinkedIn:** Professional networks



- **Academic collaborations:** Co-authorship networks

### **Biological Networks**

- **Protein-protein interactions:** How proteins interact
- **Gene regulatory networks:** How genes control each other
- **Metabolic networks:** Biochemical reactions
- **Neural networks:** Brain connectivity

### **Technological Networks**

- **Internet:** Router connections
- **Power grids:** Electrical transmission
- **Transportation:** Roads, flights, public transit
- **Communication:** Phone calls, emails

### **Information Networks**

- **World Wide Web:** Web pages and links
- **Citation networks:** Academic paper references
- **Wikipedia:** Articles and internal links
- **Software dependencies:** Package relationships

## **6. Network Analysis Workflow**

### **Step 1: Data Collection**

- Identify entities (nodes) and relationships (edges)
- Collect relevant attributes
- Handle missing data and errors
- Validate data quality

### **Step 2: Network Construction**

- Choose appropriate network type
- Define node and edge attributes
- Handle multiple relationships
- Validate network structure

### **Step 3: Basic Analysis**

- Calculate network properties
- Identify key nodes and edges
- Analyze network structure
- Detect anomalies

### **Step 4: Advanced Analysis**

- Community detection
- Centrality analysis
- Network dynamics
- Predictive modeling

### **Step 5: Visualization and Interpretation**

- Create effective visualizations
- Interpret results in context
- Draw conclusions
- Communicate findings

## **7. Key Network Metrics**

### **Centrality Measures**

- **Degree centrality:** Number of connections
- **Closeness centrality:** Average distance to other nodes
- **Betweenness centrality:** Role as bridge between others
- **Eigenvector centrality:** Importance based on neighbors' importance

### **Clustering**

- **Clustering coefficient:** How connected neighbors are
- **Local clustering:** Clustering around specific nodes
- **Global clustering:** Overall network clustering

### **Assortativity**

- **Degree assortativity:** Do high-degree nodes connect to other high-degree nodes?
- **Attribute assortativity:** Do similar nodes connect to each other?

## 8. Applications and Impact

### Social Sciences

- Understanding social influence and diffusion
- Analyzing organizational structures
- Studying political networks

### Biology and Medicine

- Drug discovery and protein interactions
- Disease spread modeling
- Brain connectivity analysis

### Technology and Engineering

- Internet and communication networks
- Power grid resilience
- Transportation optimization

### Economics and Finance

- Supply chain analysis
- Financial contagion modeling
- Market structure analysis

## 9. Tools and Software

### Python Libraries

- **NetworkX**: Comprehensive network analysis
- **igraph**: Fast network algorithms
- **PyTorch Geometric**: Graph neural networks

### Specialized Software

- **Gephi**: Interactive network visualization
- **Cytoscape**: Biological network analysis
- **Pajek**: Large network analysis
- **UCINET**: Social network analysis

## 10. Challenges and Future Directions

### Data Challenges

- **Scale:** Handling massive networks
- **Quality:** Dealing with noisy data
- **Privacy:** Protecting sensitive information
- **Dynamic:** Analyzing evolving networks

### Methodological Challenges

- **Causality:** Distinguishing correlation from causation
- **Temporal dynamics:** Understanding network evolution
- **Multilayer networks:** Analyzing multiple relationship types
- **Non-stationarity:** Handling changing network properties

### Emerging Areas

- **Graph Neural Networks:** Machine learning on networks
- **Temporal networks:** Time-evolving networks
- **Multilayer networks:** Multiple relationship types
- **Network medicine:** Medical applications

## Summary

Network Science provides powerful tools for understanding complex systems across many domains. By representing systems as networks, we can:

1. **Identify patterns** in complex interactions
2. **Predict behavior** based on network structure
3. **Optimize systems** by understanding connectivity
4. **Understand emergence** of collective behavior

The field continues to grow with new applications in machine learning, medicine, and technology. Understanding network science fundamentals provides a foundation for advanced analysis and research.

## Key Takeaways

- Networks represent complex systems as nodes and edges
- Different network types capture different relationship types
- Network properties reveal system behavior and structure
- Network analysis follows a systematic workflow
- Applications span social, biological, and technological domains
- Tools and methods continue to evolve rapidly

## Key Insights for Learning

When you encounter any system with relationships, try to identify:

1. What are the entities? (These become your nodes)
2. How do they connect? (These become your arcs)
3. Do the connections have direction? (Directed vs. undirected)
4. Do the connections have different strengths or costs? (Weighted vs. unweighted)

Graph theory provides a powerful language for describing and analyzing these relationships, making complex systems easier to understand and optimize.

## Practice Exercise

Try modeling your own social network or the websites you visit regularly as a graph. Identify the nodes and arcs, and consider what insights you might gain from analyzing the structure. What patterns do you notice? Who or what serves as important connection points?

## Next Steps

- Learn about network connectivity and paths
- Explore centrality measures in detail

- Study community detection methods
- Understand network dynamics and evolution
- Practice with real-world datasets
- Explore advanced topics like graph machine learning