

02-680 Module 4

Essentials of Mathematics and Statistics

Aidan Jan

September 16, 2025

Graphs

Graphs are powerful tools for modeling structured **relationships** between entities. A graph consists of:

- **Nodes** (or Vertices): The entities in the graph
- **Edges** (or Links): The relationships between pairs of nodes.

A graph is also called a **network**.

Undirected Graphs

- In **undirected graphs**, edges represent mutual relationships. If node A is connected to node B , then node B is also connected to node A .

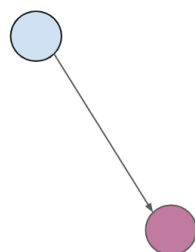
Many real-world relationships are mutual. This representation is suitable for systems where all connections are peer-to-peer.

Directed Graphical Models

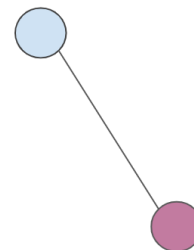
In a social network, each **node** represents a person, and each **directed edge** represents a relationship such as following, influencing, or messaging. These directed connections capture how information, behavior, or influence propagates across the network.

Directed Graph vs. Undirected Graph

In a directed graph, an edge is an **ordered pair** of vertices ("an edge from u to v ") and in an undirected graph, an edge is an unordered pair of vertices ("an edge between u and v ").



directed graph



undirected graph

A **graph** is a mathematical structure used to model pairwise relationships. We typically write a graph as a tuple $G = \langle V, E \rangle$.

- V is a set of nodes or vertices
- E is a set of edges, each connecting a pair of nodes.
 - $E \subseteq V \times V$ if G is **directed**
 - $E \subseteq \{\{u, v\} : u, v \in V\}$ if G is **undirected**
 - Note that one edge can loop around the same node. (e.g., $u = v$)

Nodes in a graph are typically drawn as circles and edges are represented as lines. In directed graphs, each edge is drawn with an arrow indicating its orientation ("which way it goes").

Parents and Children

In a directed graph, there exists the notion of parent nodes and child nodes. If an arrow connects two variables X and Y (in either direction) we say that X and Y are adjacent. If there is an arrow from X to Y , then X is a parent of Y , and Y is a child of X . A **directed path** between two variables is a set of arrows all pointing in the same direction linking one variable to the other such as:



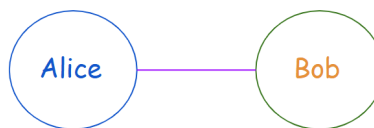
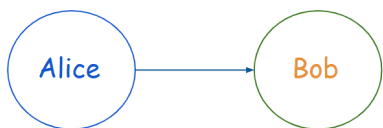
Some Examples of Graphs

To understand a **directed graph**, think about the difference between:

- Twitter follows: $A \rightarrow B$ (directed)
- Facebook friendships: $A - B$ (undirected)

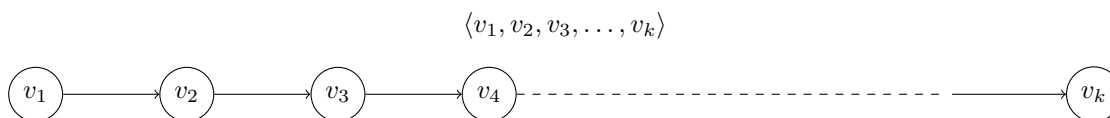
In Twitter, Alice can follow Bob without Bob following back — this is a **one-way (directed) edge**.

In Facebook, friendship is mutual — it exists only if both agree, representing an **undirected edge**.



Path in a Directed Graph

Some examples of things that can be represented as **graphs**: train/flight maps, cell interactions, social networks, etc. A **path** is a sequence of nodes:



- $\langle v_i, v_{i+1} \rangle \in E$ if G is **directed**
- $\{v_i, v_{i+1}\} \in E$ if G is **undirected**

We say that the **path** has length k , and that a path is from v_1 to v_k .

Route Graph

In the case of a route graph (e.g., for navigation or delivery), each edge might be labeled with a street name (a string), but in other types of graphs the labels could be numerical values (such as distance, time, or cost) - which we usually call **weights**.

- A **path label** could be the concatenation of street names (e.g. "Main St." → "Broadway" → "5th Ave.")
- A **path weight** could be the **sum** of distances or travel time along the path. Edge labels can carry qualitative (string) or quantitative (numeric) information, and we interpret paths differently depending on the type of label.

Graph Function

We can also **label** or **weight edges** in some scenarios. What does this mean?

- In many types of graphs (especially weighted or labeled graphs) we want to assign a value or label to each edge.
- These labels can be:
 - Strings (e.g., street names, course codes, etc.)
 - Numbers (e.g., cost, distance, weight, time)

Formally, edges with string labels are typically represented by a function:

$$\ell : E \rightarrow \Sigma$$

(or maybe edges with weight labels function $w : E \rightarrow \mathbb{R}$)

Instead of assigning labels manually, we define a **function** that maps each edge to a value.

Function	Purpose	Output Type
ℓ	Edge label	A string (from Σ^*)
w	Edge weight	A real number (\mathbb{R})

Neighborhood of a Node: Undirected Graph

Let $G = \langle V, E \rangle$ be an **undirected graph**, and let $u \in V$ be a **node**. The neighborhood of u is the set $v \in V : \{u, v\} \in E$.

- That is, the set of all neighbors of u

$N_G(v)$ is the set of all nodes u

- If $\{u, v\} \in E$, it means there is an edge connecting u and v , so node u is considered a neighbor of node v .

Degree: Undirected Graph

The **degree** of a node u in an undirected graph G is the size of the neighborhood of u in G , that is, the number of nodes adjacent to u .

$$\text{degree}_G(v) := |N_G(v)| + \mathbb{1}(v \in N_G(v))$$

- Here, $\mathbb{1} : \{1, 0\}$ is an indicator function.
- Degree is defined as the size of the node's neighborhood, **plus 1** if v is in its own neighborhood (e.g., self-loop.)
- For an undirected graph, a loop at a vertex contributes **twice to the degree** of that vertex.

Aside: Indicator Function

The **indicator function** is a simple mathematical tool used to express whether a condition is **true** or **false**, in a numerical form. Given an arbitrary set X , the indicator function of a subset A of X is the function:

$$1_A : X \mapsto \{0, 1\}$$

defined by

$$1_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases}$$

Degree: Directed Graph

- For an edge $\langle u, v \rangle$ from node u to node v in a directed graph, we say that:
 - the node v is an **out-neighbor** of the node u
 - the node u is an **in-neighbor** of the node v
 - we have **in-degrees** and **out-degrees** for their respective neighborhoods.
 - All nodes going into or out of v are referred to as being **adjacent** to v .

Regular Graphs

In an *undirected* graph, it is **regular** if every node has the same degree.

- Let $d \geq 0$ be an integer. A d -regular graph is a graph G such that every node has degree precisely equal to d . If G is d -regular for any d , then we say that G is a regular graph.

In a *directed* graph, it is **regular** if every node has identical **in-degrees** and **out-degrees**.

- For example, in-degree = 2 and out-degree = 2 for every vertex.

Complete Graphs

A **complete graph**, or **clique** is an undirected graph $G = \langle V, E \rangle$ such that $\{u, v\} \in E$ for any two distinct nodes $u \in V$ and $v \in V$.

- Every two distinct nodes are directly connected by an edge
- There are no missing connections between any two nodes
- The graph is maximally connected.

If $V = \{A, B, C\}$, then the edge set is: $E = \{\{A, B\}, \{A, C\}, \{B, C\}\}$.

We also call complete graphs **cliques**, especially in the context of subgraphs. We do that so often we have a special notation for that: K_n , for a clique of size n .

- A clique is a set of nodes in a graph that form a complete subgraph.
- That means, every node in the clique is connected to every other node in that group.

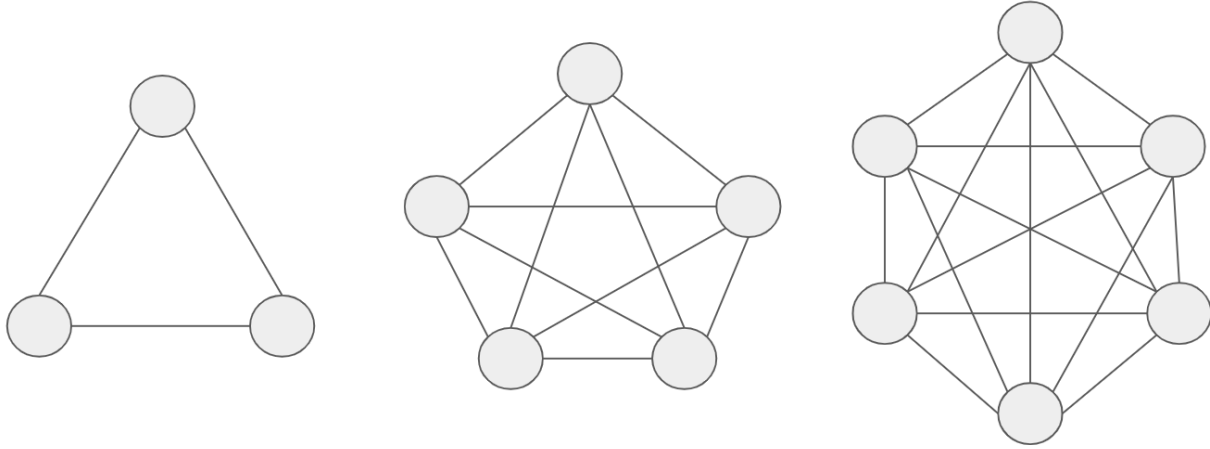
So in graph theory, the word "clique" often refers to:

- A complete graph (if it's the entire graph)
- Or a complete subgraph (a smaller part of a bigger graph)

We also know in that case that it is $|V| - 1$ regular. In a complete graph, every node is connected to all the other nodes.

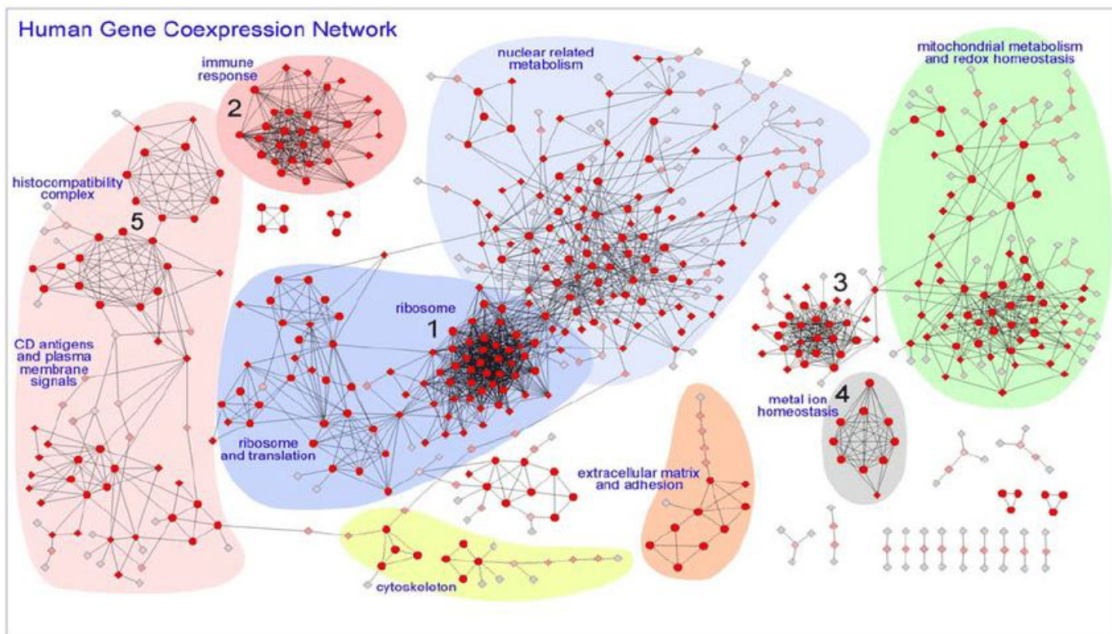
- If there are $|V| = n$ nodes total, then each node is connected to the other $n - 1$ nodes. Therefore the degree of every node is $n - 1$.
- Therefore, a complete graph with n nodes is $(n - 1)$ regular.

Examples of complete graphs of varying sizes:



Subgraphs

- Many times we only want to look at a particular part of a graph, think about say a regulatory network.
- (A regulatory network is a directed graph with nodes that are genes, and edges going from one gene to another if the source somehow impacts the expression of the sink.)
- If we run an experiment and get a list of genes that are differentially expressed in some condition
- (i.e., the expression level is different from the null/healthy case)
- We may want to try and intuit something looking only at those changes.



So we define a **subgraph** $G' = \langle V', E' \rangle$ where $V' \subseteq V$ and $E' \subseteq \{\langle u, v \rangle \mid u, v \in V' \wedge \langle u, v \rangle \in E\}$

- That is, we choose a subset of nodes and the edges associated only with the chosen nodes
- In the example above, we almost always would want to look at the induced subgraph, which is basically the same thing but E' is equal to the set of edges above rather than a subset.

Bipartite Graphs

A **bipartite graph** is $G = \langle (A \cup B), E \rangle$ where $E \subseteq \{\{u, v\} | u \in A, v \in B\}$.

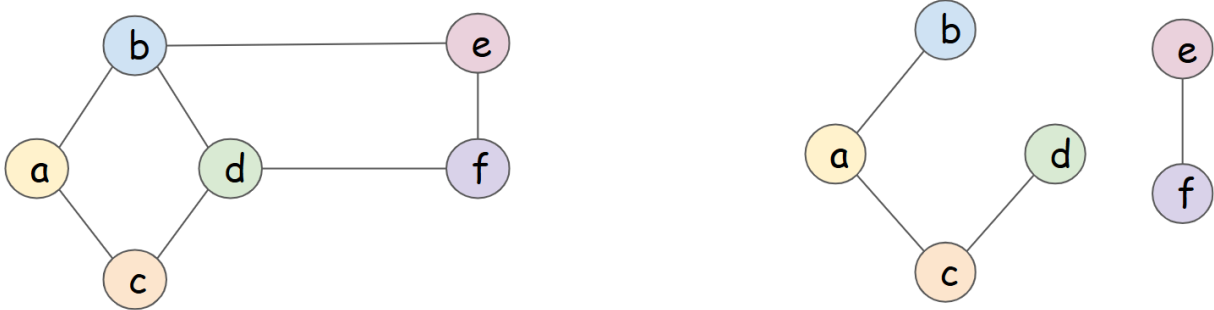
- In directed graphs edges can go from A to B or B to A , and $A \cap B = \emptyset$.
- That is, it is a graph where the nodes can be separated into two groups, and no edge exists within the group.
- A **isolated node** is one that has no connections. It still belongs to the graph, but plays no role in connectivity.

Connected Graphs

A graph G is **connected** if the following holds:

$$\forall u, v \in V : \exists \langle u, z_1, z_2, \dots, z_k, v \rangle \in V^* : \langle u, z_1 \rangle, \langle z_1, z_2 \rangle, \dots, \langle z_k, v \rangle \in E$$

That is, the graph is connected if for every pair of nodes, there is a path between them. We say that a connected component is a **subgraph** of G that is connected. (note a graph that is already connected will only have one connected component.)



The graph on the left is connected. The one on the right is not.

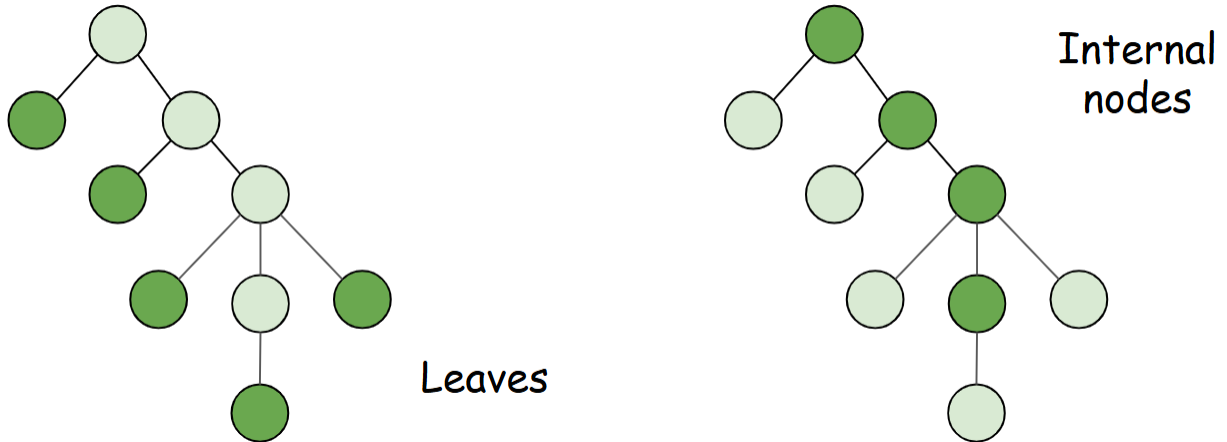
Trees

A **tree** is a special type of graph that is fully connected and has exactly $|V| - 1$ edges.

- One consequence of this is that there are no cycles; a cycle is a special type of path $\langle v_1, v_2, \dots, v_{k-1}, v_1 \rangle$ ($k \geq 2$) that ends at the same node it started at.

In a tree, nodes with only one neighbor are known as leaves. Nodes that are not leaves are called internal nodes.

$$\text{Leaves}(G) := \{v \in V | \text{degree}(v) = 1\}$$



Rooted Trees

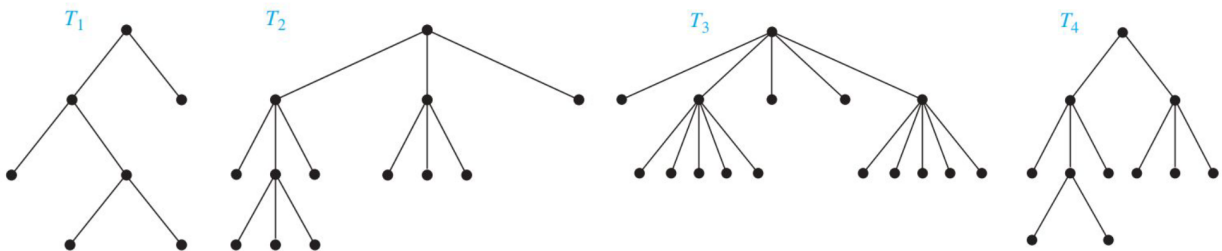
Many times we will deal with **rooted trees**, though not always. A **rooted tree** is one that has a specific **node** $r \in V$ designated as the root.

- The neighbors of the root are called its **children**.
- The children's neighbors (excluding their parent) are their children, and so on.
- This defines a parent-child hierarchy throughout the tree.
- In rooted trees, we also sometimes need to talk about **levels** (distance a given node is from the root).

Regular Trees

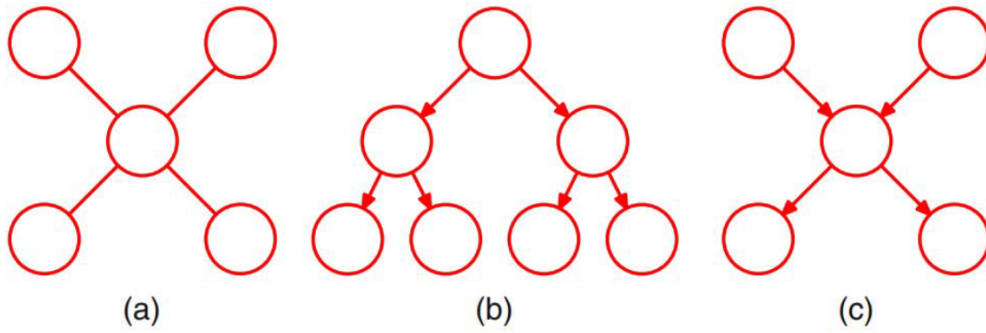
Up to now there was **no limit** on the number of neighbors (children) a node could have, and in general this is true, but many times we want to know some general property of a tree.

- We use the term k -ary trees (more commonly we will talk about binary tree, where k is 2.)
- A k -ary tree is one where every **internal node** has degree $\forall v \in V, v \text{ is internal} : |N(v)| \leq k + 1$.
- It makes more sense in **rooted trees**, it means each internal node has at most k children.
- In a rooted **binary tree**, each node has at most 2 children.
-



- In the above image, the T_1 , T_2 , and T_3 are 2-ary, 3-ary, and 5-ary trees, respectively.
- T_4 is not a k -ary tree, because not all the internal nodes have the same number of children.

Other Tree Structured Graphs



- (a) **Undirected Tree**: A tree where edges do not have direction
- (b) **Directed Tree**: A tree where edges do have direction
- (c) **Directed Polytree**: A directed tree where nodes are allowed to have multiple parents, but there still must be only one path between any pair of nodes. There can also be multiple nodes with no parents.