Graph and Network Theory

Let $G = (V, E, w)$, with $|V| = n$, be a weighted graph. Then the matrix $A(G) = [a_{ij}]_{n\times n}$ for $i = 1, ..., n$, $j = 1, ..., n$ define in the following way:

$$a_{ij} = \begin{cases} w_{ij} & \text{if } x_i \text{ is adjacent to } x_j, \\ 0 & \text{otherwise.} \end{cases}$$

where w_{ij} is the weight on edge between i and j, is called the **weighted adjacency matrix**.

A few notes in addition about the adjacency matrix

A weighted adjacency matrix can always be converted into a binary one, if desired, by using some threshold τ on the edge weights

$$a_{ij} = \begin{cases} 1 & w_{ij} > \tau, \\ 0 & \text{otherwise.} \end{cases}$$

Let $D = \{x_i\}_{i=1}^{n}$ and $x_i \in R^d$, be a dataset consisting of n observations in d-dimensional space. We can define a weighted graph $G = (V, E, w)$, where there exists a node for each observation in D, and there exists an edge between each pair of observations, with weight

$$w_{ij} = sim(x_i, x_j)$$

where $sim(x_i, x_j)$ denotes the similarity between points x_i and x_j.

For instance, similarity can be defined as being inversely related to the Euclidean distance between the observations via the transformation

$$w_{ij} = sim(x_i, x_j) = \exp\left\{-\frac{|x_i - x_j|^2}{2\sigma^2}\right\}$$

where σ is the spread parameter (equivalent to the standard deviation in the normal density function).

Dataset:

student :	*grade*1	*grade*2	*grade*3
*student*1	2	3	4
*student*2	2	3	3
*student*3	5	5	5
*student*4	3	3	3

Let's calculate the distances between observations $|x_i - x_j|^2$.

	1	2	3	4
1	0	1	196	4
2	1	0	289	1
3	196	289	0	144
4	4	1	144	0

	1	2	3	4
1	0	1	196	4
2	1	0	289	1
3	196	289	0	144
4	4	1	144	0

Let's calculate $sim(x_i, x_j)$ for $\sigma = 1.7$.

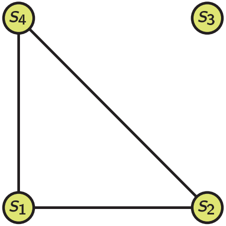
	1	2	3	4
1	1	$0,84348533254$	0	$0,50618601239$
2	$0,84348533254$	1	0	$0,84348533254$
3	0	0	1	0
4	$0,50618601239$	$0,84348533254$	0	1

student :	*grade*1	*grade*2	*grade*3
*student*1	2	3	4
*student*2	2	3	3
*student*3	5	5	5
*student*4	3	3	3

For the dataset let us convert the weighted adjacency matrix into a binary one A for $\tau = 0.5$.

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

Graphs from Data Matrix - example

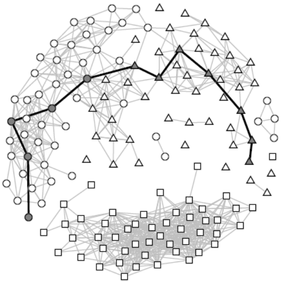


student :	*grade*1	*grade*2	*grade*3
*student*1	2	3	4
*student*2	2	3	3
*student*3	5	5	5
*student*4	3	3	3

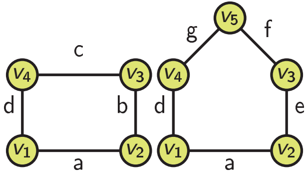
The similarity graph for the Iris dataset

The pairwise similarity between distinct pairs of points was computed with $\sigma = \frac{1}{\sqrt{2}}$. The mean similarity between points was 0.197, with a standard deviation of 0.290.

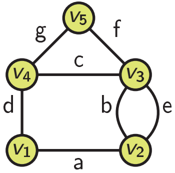
A binary adjacency matrix was obtained using a threshold of $\tau = 0.777$, which results in an edge between points having similarity higher than two standard deviations from the mean. The resulting Iris graph has 150 nodes and 753 edges.



The **union** of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is the graph $G = (V, E) = G_1 \cup G_2$ which is the union of their vertex and edge sets: $V = V_1 \cup V_2$ and $E = E_1 \cup E_2$.

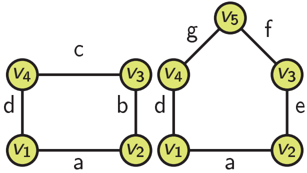


Union of graphs

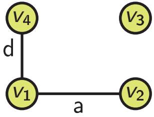


The product of graphs

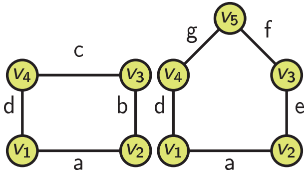
Let the graphs be given $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$.



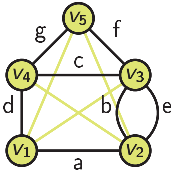
The graph $G = (V, E)$ is called **the product of graphs** G_1 and G_2,
$G = G_1 \cap G_2$, gdy $V = V_1 \cap V_2$ as far as $V \neq \emptyset$ and $E = E_1 \cap E_2$



A graph $G = (V, E)$ is called a **join** of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ (we denote a join by $G_1 + G_2$), if $V = V_1 \cup V_2$ and $E = E_1 \cup E_2 \cup \{\{v_1, v_2\} : v_1 \in V_1, v_2 \in V_2\}$



Join of graphs

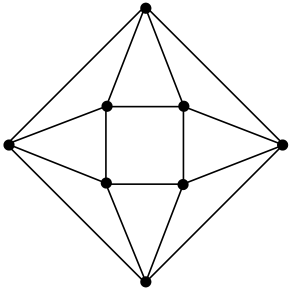


Theorem

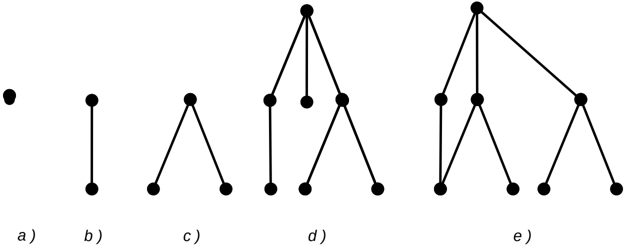
If $G = G_1 + G_2$, then $|V| = |V_1| + |V_2|$ and $|E| = |E_1| + |E_2| + |V_1| * |V_2|$.

A graph without cycles is called **acyclic**.

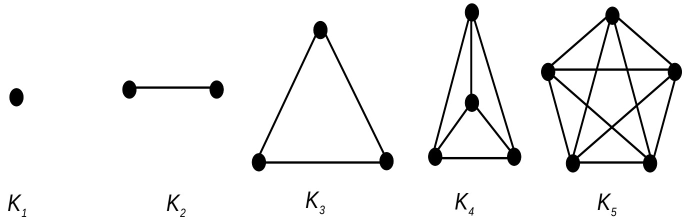
A graph is called **connected** if for any two vertices u and v there exists a path connecting u and v, i.e. with start vertex u and end vertex v.

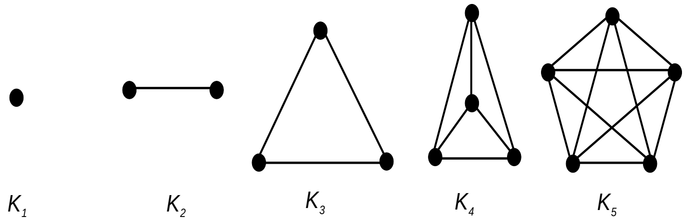


A connected and acyclic graph is called a **tree**.



The **complete graph** K_n is the graph with n vertices and all the pairs of vertices are adjacent to each other.





In any K_n,

- $\deg(v) = n - 1$;
- The number of edges of any complete graph K_n is

$$|E_{K_n}| = \frac{n(n-1)}{2}.$$

In any K_n,

- $\deg(v) = n - 1$;
- The number of edges of any complete graph K_n is

$$|E_{K_n}| = \frac{n(n-1)}{2}.$$

Proof. Because, in K_n we have n vertices and each is of degree $n - 1$, so

$$\sum_{i=1}^{n} \deg(v_i) = n(n-1)$$

By virtue of the **handshake lemma** we have

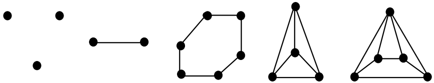
$$\sum_{i=1}^{n} \deg(v_i) = 2|E_{K_n}|$$

hence

$$|E_{K_n}| = \frac{n(n-1)}{2}$$

■

A simple graph for which every vertex has the same degree is called a **regular graph**. A regular graph with vertices of degree r is called a **r–regular graph** or regular graph of degree r. A 3-regular graph is called as a **cubic graph**.

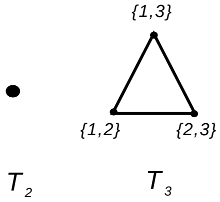


a) b) c) d) e)

Property

A $r-$regular graph with n vertices possesses $\frac{nr}{2}$ edges.

A triangular graph T_n is the simple graph which vertices are bijectivly mapped to two-element subsets of a $n-$element set X and any two vertices are connected with edge if and only if the corresponding sets are not disjoint.



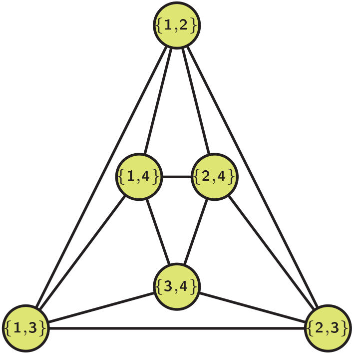
or in other words

In a triangular graph T_n vertices represent the 2-element subsets of an $n-$set, and two vertices are adjacent if they share a common element.

For the set $\{1, 2, 3, 4\}$ we get two−element subsets

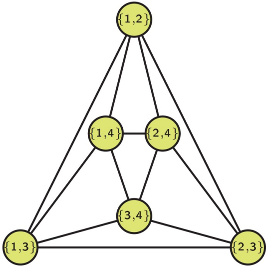
$$\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}$$

and we get



Properties of T_n

1. T_n possesses exactly $|V| = \binom{n}{2} = \frac{n(n-1)}{2}$ vertices.
2. The degree of any vertex is $2n - 4$.



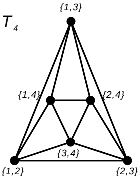
Corollary

Triangular graphs T_n are $(2n - 4)-$regular graphs.

A triangular graph has an application in sports tournament scheduling, where every pair of teams must play against each other exactly once - **Round-Robin Tournaments**.

In a round-robin tournament with $n-$teams

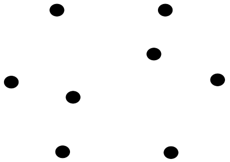
- Each match between two teams can be represented as a vertex in T_n
- An edge exists between two matches if they share a common team.



Why is this useful?

- Helps in designing an efficient schedule where teams do not play multiple games simultaneously.
- Ensures fairness by balancing rest periods between games.
- Used in chess, curling, and esports tournament planning.

If $E = \emptyset$ and $V \neq \emptyset$ then a graph $G = (V, E)$ is called the **empty graph** The empty graph with n vertices will be denoted by N_n.

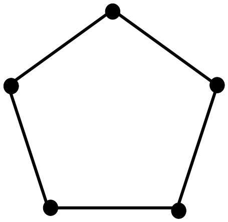


N_8

Property

1. Any empty graph N_n consists of n isolated vertices.
2. An empty N_n 0−regular graph.

A connected 2−regular graph is called a **cycle graph** or a **circular graph**. A circular graph with n vertices will be denoted by C_n.

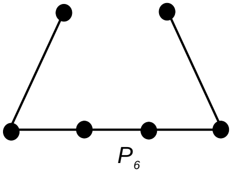


C_5

Property

A graph C_n has exactly n edges.

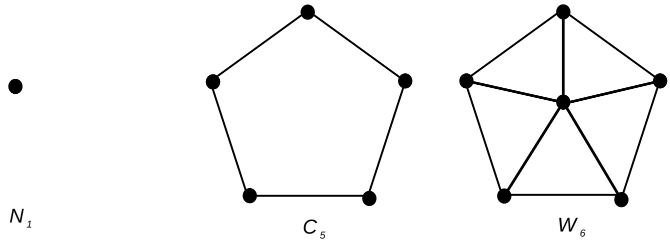
If we delete from a graph C_n exactly one edge we get a graph which is called a **path graph**. A path graph with n vertices will be denoted by P_n.



Property

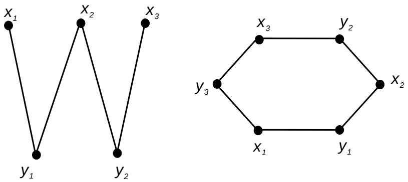
1. Any P_n is not regular.
2. A graph P_n possesses exactly 2 vertices of degree one and the remaining $n-2$ vertices are of degree two.

A ***disk*** W_n is the graph $W_n = C_{n-1} + N_1$.



$$W_6 = C_5 + N_1$$

Let $G = (V, E)$ be a simple graph.
G is called **bipartite** if its set of vertices V is the union of two nonempty disjoint sets V_1 i V_2 such that every edge of G connects a vertex from V_1 with a vertex from V_2.



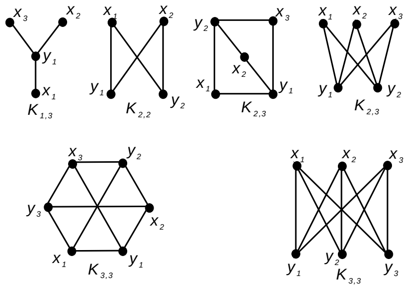
Theorem

If G is bipartite, then every cycle in G is of even length.

Proof. Let $V = V_1 \cup V_2$ and let$v_1 v_2 v_3 \ldots v_m v_1$ be cycle in G. Let's assume, without loss of generality, that $v_1 \in V_1$, then $v_2 \in V_2$, $v_3 \in V_1$, etc. (The vertices with even indices belong to the set V_2, and those with odd indices to the set V_1). Since the cycle is length m and $v_m \in V_2$, so the cycle has an even length. ■

Complete bipartite graphs

A bipartite graph is called **complete bipartite graph**, if every vertex from V_1 is connected with every vertex from V_2 with exactly one edge. A complete bipartite graph for which $|V_1| = r$ and $|V_2| = s$ will be denoted by $K_{r,s}$.



Properties

1. $K_{r,s} = N_r + N_s$
2. Every $K_{r,s}$ has exactly $r + s$ vertices.
3. Every $K_{r,s}$ has exactly $r \cdot s$ edges.

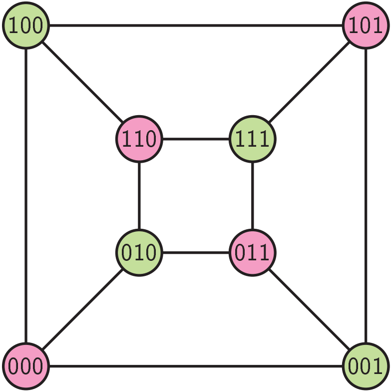
Algorithm for checking bipartite graph

```
BIPART(G, s; S, T, bip)
Q ← 0, bip ← true, S ← ∅
for every v ∈ V
      do
          d(v) ← ∞
d(s) ← 0, s ⇒ Q
while Q ≠ ∅ and bip = true
      do
          u ← head[Q]
          for every v ∈ Adj[u]
               do
                   if d(v) = ∞ then d(v) ← d(u) + 1; v ⇒ Q
                   else if d(u) = d(v) then bip ← false
if bip = true
    then
          for v ∈ V
               do
                   if d(v) = 0(mod2) then S ← S ∪ {v}
          T ← V \ S
```

Algorithm

Input: a connected graph G , an arbitrarily chosen vertex s of G
Output: if G is bipartite, then *bip* is *true* and we get S i T, such that $S \cup T = V$ i $S \cap T = \emptyset$

A graph which vertices can be bijectively mapped to all binary sequences $(a_1, a_2, \ldots, a_n)$ and its edges are connected if and only if corresponding sequences differ at exactly one position (one bit) is called $n-$**hypercube** and will be denoted by Q_n.



A bipartite graph is widely used in recommendation systems, such as **Netflix (movies)**, **Amazon (products)**, or **Spotify (music recommendations)**. The idea is to connect users with items (movies, products, or songs) they have interacted with.

Movie recommendation system (Netflix Model)

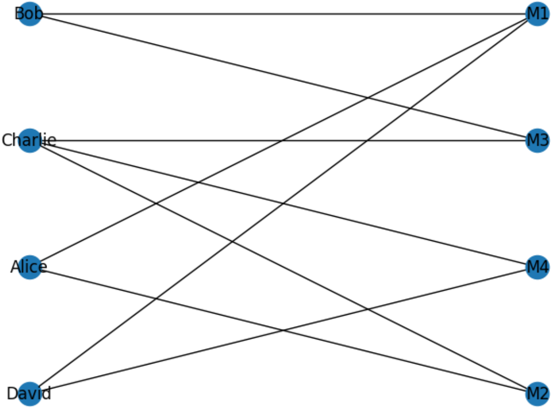
- Set U (Users): Alice (A), Bob (B), Charlie (C), David (D)
- Set V (Movies): Inception (M1), Interstellar (M2), Avengers (M3), Titanic (M4)

graph:

- Alice (A): M1, M2
- Bob (B): M1, M3
- Charlie (C): M2, M3, M4
- David (D): M1, M4

Application of bipartite graphs - Netflix

Users	M1 (Inception)	M2 (Interstellar)	M3 (Avengers)	M4 (Titanic)
A	✅ Watched	✅ Watched	❌ Not Watched	❌ Not Watched
B	✅ Watched	❌ Not Watched	✅ Watched	❌ Not Watched
C	❌ Not Watched	✅ Watched	✅ Watched	✅ Watched
D	✅ Watched	❌ Not Watched	❌ Not Watched	✅ Watched



Users	M1 (Inception)	M2 (Interstellar)	M3 (Avengers)	M4 (Titanic)
A	✅ Watched	✅ Watched	❌ Not Watched	❌ Not Watched
B	✅ Watched	❌ Not Watched	✅ Watched	❌ Not Watched
C	❌ Not Watched	✅ Watched	✅ Watched	✅ Watched
D	✅ Watched	❌ Not Watched	❌ Not Watched	✅ Watched

We measure similarity between users based on common movies watched.

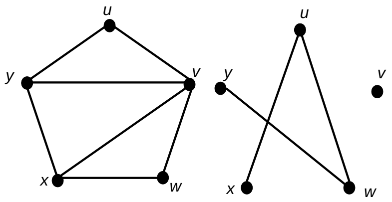
- Alice (A) and Bob (B) both watched M1 $\Rightarrow$ they have similar interests.
- Charlie (C) and Alice (A) both watched M2 $\Rightarrow$ they have similar interests.
- Charlie (C) and David (D) both watched M4 $\Rightarrow$ they have similar interests.

Users	M1 (Inception)	M2 (Interstellar)	M3 (Avengers)	M4 (Titanic)
A	✅ Watched	✅ Watched	❌ Not Watched	❌ Not Watched
B	✅ Watched	❌ Not Watched	✅ Watched	❌ Not Watched
C	❌ Not Watched	✅ Watched	✅ Watched	✅ Watched
D	✅ Watched	❌ Not Watched	❌ Not Watched	✅ Watched

Generate Recommendations

- Since Bob watched M3 and Alice (A) has not $\Rightarrow$ **recommend M3 to Alice**;
- Since David watched M1 and Charlie (C) has not $\Rightarrow$ **recommend M1 to Charlie**;
- Since Charlie watched M4 and Alice has not $\Rightarrow$ **recommend M4 to Alice**.

A **complement** or an **inverse** of a graph $G = (V, E)$ is the graph $\overline{G}$ which set of vertices is identical with the set of vertices of G and every vertices are adjacent if and only if they are not adjacent in G.



Property

1. A complement of K_n is the empty graph N_n.
2. A complement of bipartite graph $K_{r,s}$ is the union of K_r i K_s.

Thank you for your attention!!!