upf. Universitat Pompeu Fabra Barcelona

Lecture 2 · Graph Theory I

Networks, Crowds and Markets

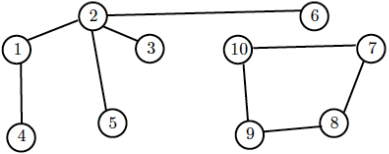
Today's Lecture

- Special graphs: complete, null, path, cycle.
- Degree. Handshaking Lemma. Degree sequence. Average degree. Degree distribution.
- Isomorphic graphs
- Subgraphs
- Adjacency matrix. Powers of adjacency matrix.
- Variations: Multigraphs, directed graphs, weighted graphs

Graph

Definition

A **graph** is a pair $G = (V, E)$ where $V = [N]$ is the set of N nodes (or vertices) and $E \subseteq \binom{V}{2}$ is a set of unordered pairs of distinct vertices, called edges (or links).



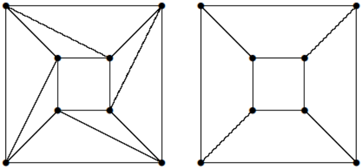
$$V = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$$
$$E = \{\overline{12}, \overline{14}, \overline{23}, \overline{25}, \overline{26}, \overline{78}, \overline{89}, \overline{910}\}$$

We often write $\{i, j\}$ or simply ij for an edge rather than $\overline{ij}$.

Subgraphs

Definition

$H = (V', E')$ is a **subgraph** of $G = (V, E)$ if all the nodes and edges of H belong to G; $V' \subseteq V$, $E' \subseteq E$. We write $H \subseteq G$.



G $\quad$ H

Here G and H have the same vertex sets but this is not the case generally for a subgraph.

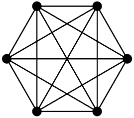
Special graphs

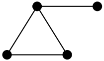
Special Graphs: The Complete Graph K_N

Definition

Graph is complete if every pair of vertices is joined by an edge.

A complete graph has $\binom{N}{2} = \frac{N(N-1)}{2}$ edges.





If $H \subseteq G$ and H is complete then H is called a clique of G.

In `NetworkX`:

`import networkx as nx` – load NetworkX
`import matplotlib.pyplot as plt`

`G = nx.complete_graph(N)` – the complete graph with N nodes.

```
nx.draw(G, with_labels=True, node_color="lightblue", node_size=600)
plt.show().
```

Special Graphs: The Null Graph (aka the empty graph)

Definition

Graph is empty if it does not contain any edge, therefore $E = \emptyset$.

The complement $\overline{G}$ of a graph $G = (V, E)$ has the same set of vertices and an edge is in $\overline{G}$ if an only if it is not present in G.

The empty graph is the complement of the complete graph.

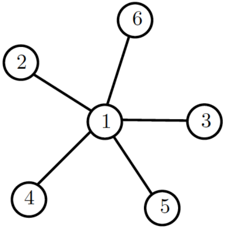
In `NetworkX`:

`G = nx.empty_graph(N)` – the empty graph with N nodes

Special Graphs: The Star Graph S_N

Definition

A **star graph** S_N has one central vertex connected to all other $N-1$ vertices, and no other edges. The central node has degree $N-1$, while all other nodes have degree 1.



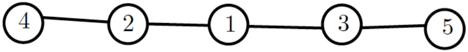
In `NetworkX`:

`G = nx.star_graph(N)` the star graph with $N+1$ nodes

The Path Graph P_N

Definition

A **Path Graph** is a connected graph $G = (V, E)$ whose nodes can be listed in order $(v_1, \dots, v_N)$ and the edges are $v_i v_{i+1}$. The central nodes have degree 2, the ones in the terminal vertices have degree 1.



Here $v_1 = 4$, $v_2 = 2$, $v_3 = 1$, $v_4 = 3$, $v_5 = 5$.

In `NetworkX`:

`G = nx.path_graph(N)` the path graph with N nodes

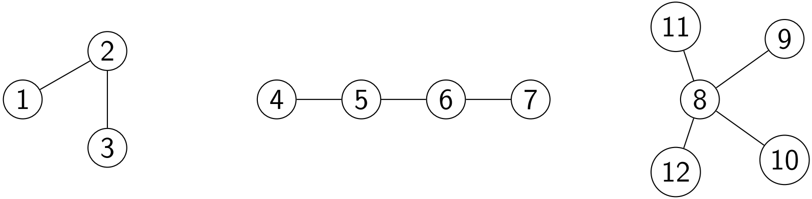
Connected components

Definition

$G = (V, E)$ is connected if any two vertices are connected by a path.

Definition

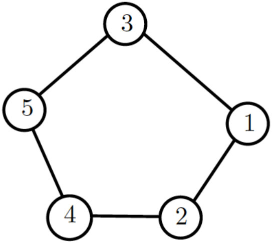
Every graph can be decomposed into its connected components, that is, maximal subgraph that are connected.



Special Graphs: The Cycle Graph C_N

Definition

A **cycle graph** C_N is obtained by connecting N vertices in a closed chain: $E = \{v_1 v_2, v_2 v_3, \ldots, v_{N-1} v_N, v_N v_1\}$.



In `NetworkX`:

`G = nx.cycle_graph(N)` the path graph with N nodes

Special Graphs: Bipartite graphs

Definition

A graph $G = (V, E)$ is bipartite if the set of vertices can be partitioned into two parts $V = V_1 \cup V_2$ such that all edges have one end in V_1 and one end in V_2

- The star S_N
- The full bipartite graph $K_{N,N}$
- Cycle C_N for even N.

Theorem

G is bipartite if and only G has no odd cycles as subgraphs.

(equiv. no closed walks of odd size)

In `NetworkX`:

```
from networkx.algorithms import bipartite
nx.is_bipartite(G)
```

Appendix: Bipartite if and only if No Odd Cycle — Proof

$\Longrightarrow$ Suppose G is bipartite with parts V_1, V_2. Every walk alternates between V_1 and V_2. Any closed walk therefore has even length (measured in the number of edges). In particular, no odd cycle can exist.

$\Longleftarrow$ Suppose G has no odd cycle.
Assume without loss of generality that G is connected (see exercise below). Fix a vertex r and define a partition by parity of distance from r:

$$V_1 = \{v : \text{ every walk from } r \text{ to } v \text{ has odd length}\},$$

$$V_2 = \{v : \text{ every walk from } r \text{ to } v \text{ has even length}\}.$$

This is well defined: if there were both an even and an odd walk from r to the same vertex v, concatenating one with the reverse of the other would produce an odd closed walk (from r to r), which necessarily contains an odd cycle, contradicting the assumption (see exercise below).

This argument shows that $V_1 \cap V_2 = \emptyset$. It remains to show that every edge must go between V_1 and V_2 (see exercise below). This implies that G is bipartite.

Exercise 1 How do we adjust the proof if G is not connected?

Exercise 2 Show that any odd closed walk contains an odd cycle.

Exercise 3 Show that, if G has no odd cycles, there can be no edge within V_1 constructed above.

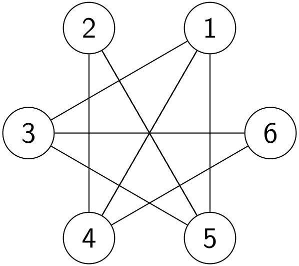
Pretty much the same proof is given in Bondy & Murty, Graph Theory with Applications, Theorem 1.2; see here.

Degree and degree distribution

Degree

Definition

The **degree** of a node $v \in V$ in an undirected graph, represented by $\deg(v)$, is the number of edges incident to it.



$\deg(1) = 3$

$\deg(2) = 2$

$\deg(3) = 3$

$\deg(4) = 3$

$\deg(5) = 3$

$\deg(6) = 2$

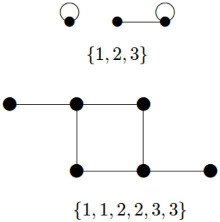
In `NetworkX`:

`print(G.degree(v))` – print the degree of node v

Degree sequence

Definition

The degree sequence of a graph is obtained by ordering, in an increasing way, the degrees of its nodes.



In `NetworkX`:

`deg_seq = sorted([d for n, d in G.degree()])`

`print(deg_seq_sorted)` – print the degree sequence of G

Degrees

$G = (V, E)$ a graph, $N = |V|$, $L = |E|$

- $\delta(G) = \min_{v \in V} \deg(v)$ minimum degree of G
- $\Delta(G) = \max_{v \in V} \deg(v)$ maximum degree of G
- $\overline{\deg}(G) = \frac{1}{N} \sum_{v \in V} \deg(v)$ average degree of G. ([B] uses $\langle k \rangle$)

Theorem (Handshaking Lemma)

The sum of the degrees of the vertices of a graph equals twice the number of edges:

$$\sum_{v \in V} \deg(v) = 2L.$$

In particular, the number of vertices of odd degree is always even.

Appendix: Handshaking Lemma — Proof

For each node, $\deg(v)$ gives all edges adjacent to v.

So, each edge ij is counted twice in the expression $\sum_{v \in V} \deg(v)$

(in $\deg(i)$ and in $\deg(j)$).

This gives the formula

$$\sum_{v \in V} \deg(v) \ = \ 2|E|.$$

Exercise

Let $G = (V, E)$ be a graph with $N = 10$ nodes with the following degree sequence $\{0, 0, 1, 1, 1, 2, 2, 2, 3, 6\}$.

a) Find the number of edges without drawing the graph.

b) Find the average degree of the nodes in G.

c) Draw a possible graph that satisfies the conditions.