Universitat
Pompeu Fabra
Barcelona

Lecture 3 · Graph Theory II

Networks, Crowds and Markets

Quick recap

- Networks have different aspects in different applications.
- Some of these aspects appear in many context (e.g. small world).
- Simple undericted graph: first formalization of a network.
- Eulerian walks: first example of a non-trivial question.
- A bunch of special graphs: full, star, path.
- Bipartite graphs, 2-colorings.
- Degree, degree sequence, handshaking lemma.

Colab1 gave an introduction to working with Python and NetworkX.

Today's Lecture

1. Degree distribution
2. Isomorphic graphs
3. Adjacency Matrix. Powers of adjacency matrix
4. Beyond simple undirected graphs
   - Multigraphs and loops
   - Directed and weighted graphs

Degree Distribution

Definition

The degree distribution, $p = (p_k)_{k \in \mathbb{N}}$, is a sequence that provides the relative ratio of the vertices with a given degree k.

(In particular, $p_k = 0$ for $k \geq N$.)

Degree Distribution

Definition

The degree distribution, $p = (p_k)_{k \in \mathbb{N}}$, is a sequence that provides the relative ratio of the vertices with a given degree k.
(In particular, $p_k = 0$ for $k \geq N$.)

- $p_k = \dfrac{N_k}{N}$, ($N_k =$ number of vertices of degree k)
- $\displaystyle\sum_{k=0}^{\infty} p_k = 1$ normalization
- $\overline{\deg}(G) = \frac{1}{N} \sum_{v \in V} \deg(v) = \frac{1}{N} \sum_{k=0}^{\infty} k \cdot N_k = \sum_{k=0}^{\infty} k \cdot p_k$

  By handshaking lemma: $\overline{\deg}(G) = \frac{2L}{N}$

This notation and terminology alludes to probability.

Consider a probability distribution $q = (q_k)$ on $\{0, 1, 2, \ldots\}$.

If $X \sim q$ then $\mathbb{E}X = \sum_{k=0}^{\infty} k \cdot q_k$.

This notation and terminology alludes to probability.

Consider a probability distribution $q = (q_k)$ on $\{0, 1, 2, \ldots\}$.

If $X \sim q$ then $\mathbb{E}X = \sum_{k=0}^{\infty} k \cdot q_k$.

Consider now a random sample $X_1, \ldots, X_N \sim q$.

The statistics N_k counts the number of times we observed $X_i = k$.

Sample distribution: $p = (p_k)$ with $p_k = \frac{1}{N} N_k$ for $k \geq 0$.

This notation and terminology alludes to probability.

Consider a probability distribution $q = (q_k)$ on $\{0, 1, 2, \ldots\}$.

If $X \sim q$ then $\mathbb{E}X = \sum_{k=0}^{\infty} k \cdot q_k$.

Consider now a random sample $X_1, \ldots, X_N \sim q$.

The statistics N_k counts the number of times we observed $X_i = k$.

Sample distribution: $p = (p_k)$ with $p_k = \frac{1}{N} N_k$ for $k \geq 0$.

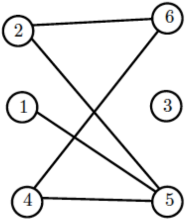
The sample average is

$$\overline{X} = \frac{1}{N} \sum_{i=1}^{N} X_i = \frac{1}{N} \sum_{k \geq 0} k \cdot N_k = \sum_{k \geq 0} k \cdot p_k.$$

If N is large $p \approx q$ by the Law of Large Numbers.

. . . one could imagine q that "generated" our degree sequence.

Example



$$p_k = \begin{cases} \frac{1}{6} & \text{if } k = 0 \\ \frac{1}{6} & \text{if } k = 1 \\ \frac{1}{2} & \text{if } k = 2 \\ \frac{1}{6} & \text{if } k = 3 \end{cases}$$

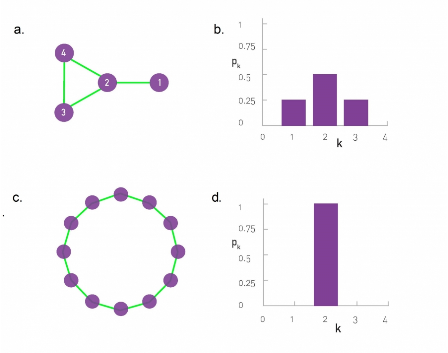
$$\overline{\deg}(G) = \sum_{k=0}^{\infty} k \cdot p_k = \frac{1}{6} \cdot 0 + \frac{1}{6} \cdot 1 + \frac{1}{2} \cdot 2 + \frac{1}{6} \cdot 3 = \frac{5}{3}$$

In `NetworkX`:

```
import numpy as np

deg_seq = sorted([d for n, d in G.degree()])
values, counts = np.unique(deg_seq, return_counts=True)
distribution = counts / counts.sum()
```

Plotting the degree distribution



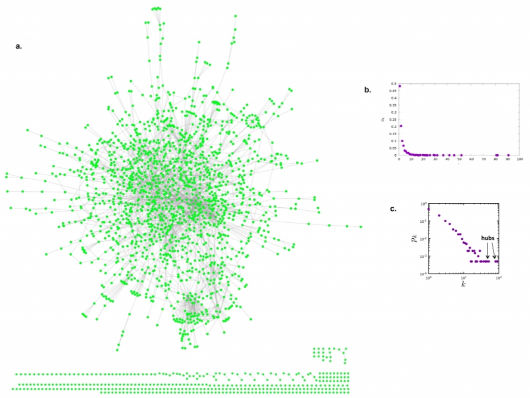
In NetworkX:

```
plt.hist(deg_seq, bins=range(max(deg_seq)+2), align='left', rwidth=0.8)
plt.xlabel("Degree")
plt.show()
```

Check this colab.

Power law

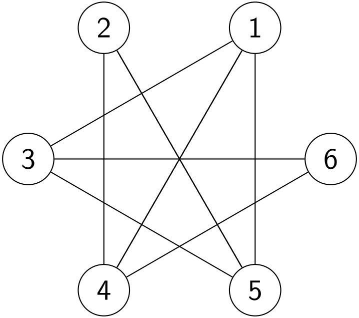
Many real networks have degree distributions that follow a **power law**, meaning the probability of high-degree nodes decays as a power of their degree, creating rare but influential hubs.



If $p_k \approx \alpha k^{\beta}$ then $\log p_k \approx \log \alpha + \beta \log k$ ($\log p_k$ is linear in $\log k$)

Exercise

Given the following graph:

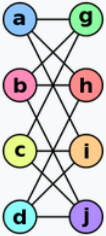
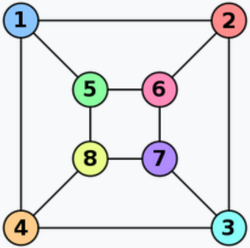


a) Determine the average degree of the graph.

b) Determine the degree distribution.

Isomorphic graphs

Isomorphic Graphs

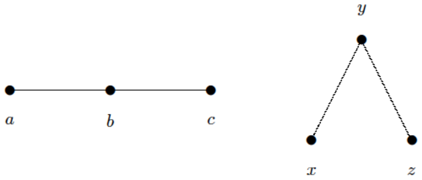
Graph G	Graph H	An isomorphism between G and H
		$f(a) = 1$ $f(b) = 6$ $f(c) = 8$ $f(d) = 3$ $f(g) = 5$ $f(h) = 2$ $f(i) = 4$ $f(j) = 7$

Isomorphic Graphs

Definition

Two graphs G and H are **isomorphic** if there exists a bijection φ between their set of nodes that preserves the edges. Alternatively:

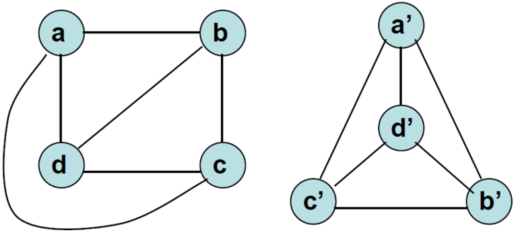
$$\exists\varphi : V_G \to V_H \ : \ \overline{uv} \in E_G \iff \overline{\varphi(u)\varphi(v)} \in E_H$$



Isomorphic graphs must have the same degree distribution.

Isomorphic Graphs

Example: $V_H = \{a, b, c, d\}, \quad V_G = \{a', b', c', d'\}.$



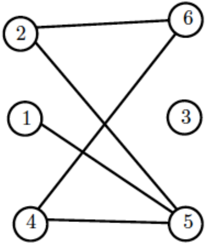
$\varphi(a) = a', \quad \varphi(b) = b', \quad \varphi(c) = c', \quad \varphi(d) = d'.$

Adjacency matrix

Adjacency Matrix (of a simple unridected graph)

Definition

The **Adjacency Matrix** of an undirected graph is a square $N \times N$ matrix A in which the position (i, j) equals 1 if there is an edge between nodes i and j, otherwise, we type a 0.

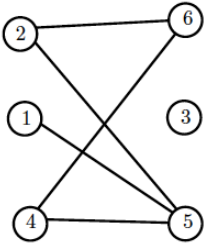


$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Adjacency Matrix (of a simple unridected graph)

Definition

The **Adjacency Matrix** of an undirected graph is a square $N \times N$ matrix A in which the position (i, j) equals 1 if there is an edge between nodes i and j, otherwise, we type a 0.



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

In `NetworkX`:

`nx.to_numpy_array(G, nodelist=sorted(G.nodes()))` – G to A

`G = nx.from_numpy_array(A)` – A to G

NumPy provides fast, memory-efficient arrays and mathematical tools for scientific computing.

Adjacency Matrix

Some elementary properties

- For an undirected graph, the adjacency matrix A is symmetric.
- The sum of entries in row i is $\deg(i)$.
- Since there are no self-loops, the diagonal entries are zero.
- The entry $(A^2)_{ij}$ counts the number of walks of length 2 from i to j. In particular, $\mathrm{tr}(A^2) = 2L$.
- More generally, $(A^m)_{ij}$ counts the walks of length m from i to j. In particular, $\mathrm{tr}(A^3) = 6\times$ (number of triangles).

Note

Matrix powers count walks — this will reappear later in centrality measures and diffusion.

More complicated types of graphs

Multigraphs and loops

Note

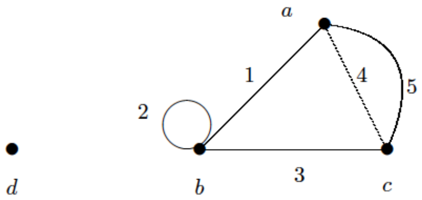
The graphs we have seen so far are called **simple** undirected graphs

- If two or more edges join the same pair of nodes, the graph is called a **multigraph**.
- If one edge joins a node with itself, we call it a **loop**.

In applications more complicated types of graphs may appear.

Undirected multigraphs

Example of a **multigraph with a loop**:



$$V = \{a, b, c, d\} \qquad E = \{1, 2, 3, 4, 5\}$$

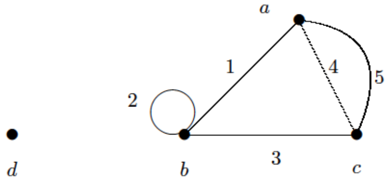
Remark: In this case, the edges have their own labels.

In `NetworkX`:

```
G = nx.MultiGraph()
G.add_nodes_from([1, 2, 3])
G.add_edge(1, 2)
G.add_edge(1, 2)   # parallel edge between 1 and 2
G.add_edge(2, 3)
```

Adjacency matrix of a multigraph

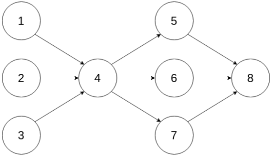
The entry ij is the number of edges joining i and j



$$A = \begin{pmatrix} 0 & 1 & 2 & 0 \\ 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- **Remark:** Adjacency matrices are symmetric ($a_{ij} = a_{ji}$) as they are undirected graphs.

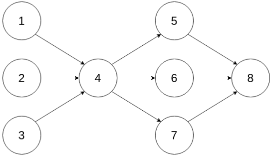
Directed Graph



Definition

A **directed graph** (digraph) is a pair $G = (V, E)$ where $V = [N]$ is the set of vertices and $E \subseteq V \times V$ is a set of ordered pairs of distinct vertices, called *directed edges* (or arcs).

Directed Graph



Definition

A **directed graph** (digraph) is a pair $G = (V, E)$ where $V = [N]$ is the set of vertices and $E \subseteq V \times V$ is a set of ordered pairs of distinct vertices, called *directed edges* (or arcs).

```
G = nx.DiGraph()
G.add_nodes_from(range(1, 9))
edges = [(1,4),(2,4),(3,4),(4,5),(4,6),(4,7),(5,8),(6,8),(2,8)]
G.add_edges_from(edges)
```

Examples of Directed Graphs

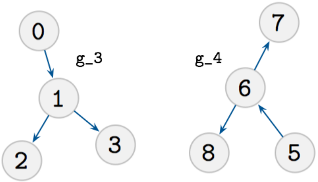
- **World Wide Web (WWW)** Vertices are webpages, edges are hyperlinks. Directed edges capture the one-way nature of links.
- **X/Twitter** Vertices are accounts, edges are "follow" relationships. Edges are directed (A follows B does not imply B follows A).
- **Academic Citation Network** Vertices are papers, edges are citations. Naturally directed in time: newer papers cite older ones.
- **SWIFT Network** Vertices are banks or institutions, edges represent international money transfers. Direction indicates sender $\rightarrow$ receiver of funds.
- **Credit Networks** Vertices are individuals or institutions, edges indicate lending/borrowing. Direction shows the obligation flow (debtor $\rightarrow$ creditor).

Isomorphism of directed graphs

Definition

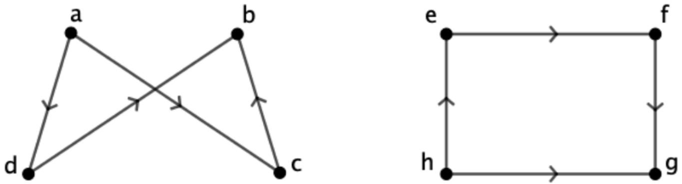
Two directed graphs G and H are **isomorphic** if there exists a bijection φ between their set of nodes that preserves the edges with its direction.

$$\exists\varphi : V_G \to V_H \text{ such that } (x, y) \in E_G \iff (\varphi(x), \varphi(y)) \in E_H$$

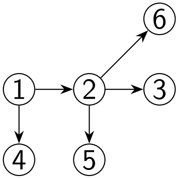


Quick question

Are they isomorphic?



In-degree and out-degree



Definition

For a directed graph $G = (V, E)$ and $v \in V$:

- The **in-degree** $d^-(v)$ is the number of edges arriving at v.
- The **out-degree** $d^+(v)$ is the number of edges leaving v.

Theorem (Handshaking Lemma for directed graphs)

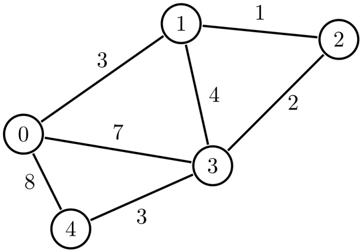
$$\sum_{v \in V} d^+(v) = \sum_{v \in V} d^-(v) = L.$$

In directed graphs the average in- and out-degrees are equal $(= \frac{L}{N})$

Weighted Graphs

Definition

A **weighted graph** is a pair (G, w) where $G = (V, E)$ is a simple graph and $w : E \to \mathbb{R}$ assigns a weight to each edge. Weights may represent distance, travel time, cost, or interaction strength (often nonnegative).



In `NetworkX` we can create a weighted graph from the (weighted) adjacency matrix – replace each 1 with the corresponding weight.

Examples of Weighted Graphs

Road Map V = intersections or cities, E = roads. *Weights:* distance, travel time, traffic congestion, or cost of tolls. Used in shortest–path algorithms (GPS navigation).

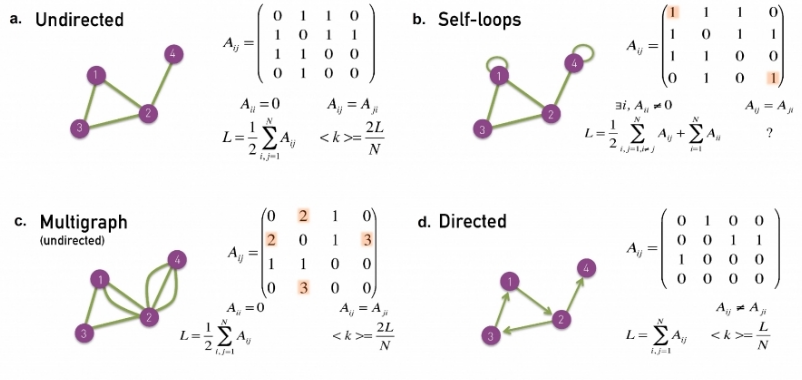
Social Networks V = individuals, E = relationships. *Weights:* frequency of interaction, strength of friendship, number of shared posts/messages. Captures tie strength rather than just "yes/no" connection.

E-mail Network V = people, E = e-mail communication. *Weights:* number of messages exchanged, total size of correspondence, or recency-weighted activity. Useful to detect communities or hubs of communication.

Food Chain Network V = species, E = "who eats whom". *Weights:* proportion of diet, biomass transfer, or energy flow between species. Central in ecology to understand stability of ecosystems.

Average Degree

In a simple graph, the vector $A\mathbf{1}$ gives the degrees. Thus $\frac{1}{N}\mathbf{1}^\top A\mathbf{1} = \frac{1}{N}\sum_{i,j=1}^{N} A_{ij}$ gives the average degree. How about other cases?



Barabási denotes the average degree by $\langle k \rangle$. We do not follow this convention.

Exercise

Suppose $G = (V, E)$ s.t. $L = 800$ edges and $N = 1000$ nodes. The average degree is

$$\overline{\deg}(G) \;=\; \frac{2L}{N} \;=\; 1.6.$$

For large graphs it is often reasonable to approximate the degree of a randomly chosen node by a Poisson distribution with mean $\lambda := \overline{\deg}(G)$:

$$p_k \;\approx\; q_k \;:=\; \mathbb{P}(\deg(v) = k) \;=\; \frac{\lambda^k}{k!}\, e^{-\lambda}, \qquad k = 0, 1, 2, \ldots$$

a) Determine the average degree of the graph.

b) What is the probability that a node has no edges attached?

c) How many nodes of degree 3 do we expect to find?

(later we discuss more in detail when such Poisson approximations make sense)

Exercise

Let B be the adjacency matrix of a directed graph G with no loops and A be the adjacency matrix of the undirected version of G. Which of the following properties are true? Justify your answer.

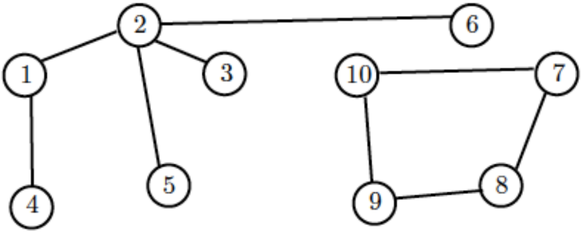
a) $a_{ij} + a_{ji} \geq 1$

b) $b_{ij} \leq a_{ij}$

c) $b_{ij} + b_{ji} \leq 1$

Exercise

Given the following graph:



a) List the set of vertices and edges
b) Find the degree for the 3 first nodes
c) Find the average degree of all the nodes in the graph