Graph Theory

Random Graphs

Sergio Peignier sergio.peignier@insa-lyon.fr

Associate Professor
INSA Lyon
Biosciences department

Table of contents

Erdös-Rényi Model

Erdös-Rényi Model

Simple procedure to generate **random graphs**.

Used as a **reference system** for many graph models.

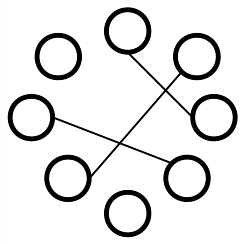
Two alternative definitions:

- Based on nb. nodes and nb. edges: $G(n, m)$.
- Based on nb. nodes and edge creation prob.: $G(n, p)$

G(n, m) definition

- Take n **disconnected nodes**: V
- Add m **edges uniformly** at random: $E \subseteq V \times V$

Equivalent def.: Pick **uniformly** at random a **graph** from the **set of all graphs** with n **nodes** and m **edges**.

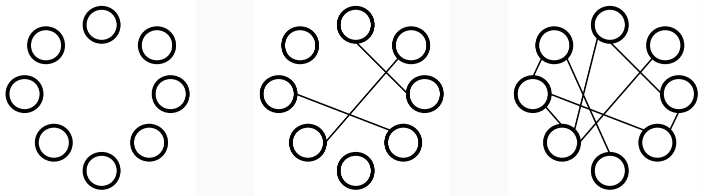


G(8,3)

G(n, p) definition

- Take n **disconnected nodes**: V
- Add an **edge between** a **random pair of nodes** with **probability** p.

Equivalent def.: Pick with **probability** $p^m(1-p)^{\binom{n}{2}-m}$ n a **graph** from the **set of all graphs with size** n.



G(8,0) G(8,0.1) G(8,0.3)

Characteristics

Degree distribution (exponential decay):

$$p_k = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

Clustering coefficient (small):

$$C_i = p$$

Avg. distance between nodes (small):

$$l = \frac{logN}{log(2p(N-1))}$$

Exercise

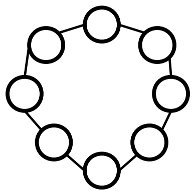
1. **Initialize** a graph with N **disconnected nodes**
2. Add one **random edge** and compute the **size** of the **largest connected component** (nx.connected_component_subgraphs).
3. Compute the **avg. node degree**
4. **Repeat** step 2 and 3 until the **graph is fully connected**.
5. Plot the **relative size** of the **largest connected component** with respect to the **avg. node degree**.
6. Observe the **structural percolation phase transition**

Watts-Strogatz model

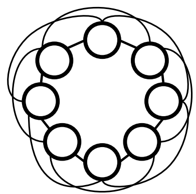
Regular Lattice

Definition: Graphs s.t. each node has the same degree k.

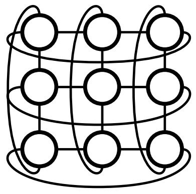
Translational symmetry



k = 2



k = 4 (1D)

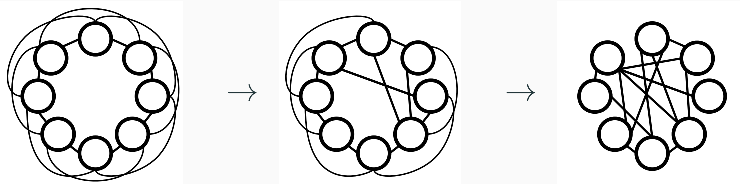


k = 4 (2D)

Watts-Strogatz model

Capture **large clustering coefficient** and **short distances**

Interpolates between an **ordered finite lattice** and a **random graph**.



Algorithm

Parameters:

- N: Nb of **nodes**
- K: Initial **coordination number**.
- p: **Rewiring probability**

Algorithm:

- Build a **ring lattice** with N nodes with **degree** K.
- **Randomly rewire** each edge with **probability** p
  Self connections and **duplicate edges** are **excluded**.

Characteristics

Degree distribution

Exponential decay

Avg clustering coefficient:

$C(p, K) = C(p = 0, K)(1 - p)^3 \quad \rightarrow$ function of p and K.

Avg. shortest distance between nodes

From $l(p = 0) = N/2K \quad \rightarrow$ large

to $1(p = 1) = \frac{\ln N}{\ln K} \quad \rightarrow$ small world

Exercise

1. Generate a **regular lattice** with N **nodes** of **degree** K
2. Generate **new graphs** applying the previous algorithm for **different values** of p **regularly spaced** in a **log scale from 0 to 1**.
3. In each case compute the **clustering coefficient** and the **average distance**.
4. Plot **both metrics** for **different values** of p.

Barabási-Albert Model

Barabási-Albert Model

Model **scale-free networks**

Based on **two principles**:

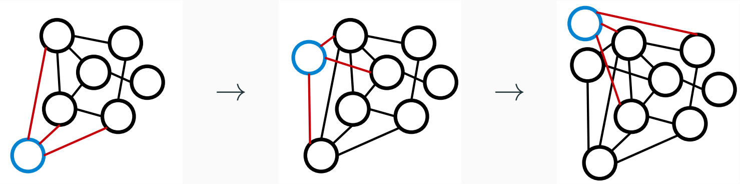
- **Evolving graph**:
  Graph **grow** as **new nodes enter the system**
- **Preferential attachment**:
  Nodes are **connected** with **higher probability** to **well connected nodes**.

Barabási-Albert Model

Algorithm

1. Start with m_0 **connected nodes**
2. Connect a **new** node with $m \leq m_0$ **links** to **existing nodes** from the graph
3. **Probability of connecting** to a **node of degree** k_i:

$$\pi(k_i) = \frac{k_i}{\sum_i k_i}$$



Characteristics

Avg. distance: $l = \frac{lnN}{lnlnN} \quad \rightarrow$ Small world

Clustering coefficient: $C = \frac{m}{8}\frac{(lnN)^2}{N} \quad \rightarrow$ N dependent

Degree distribution: Power law $\sim \gamma^{-k} \quad \rightarrow$ Scale free