MSc in Artificial Intelligence and Robotics
MSc in Control Engineering
A.Y. 2019/20

Neuroengineering

Laura Astolfi, PhD
Department of Computer, Control and Management Engineering Antonio Ruberti
Sapienza University
E-mail: laura.astolfi@uniroma1.it

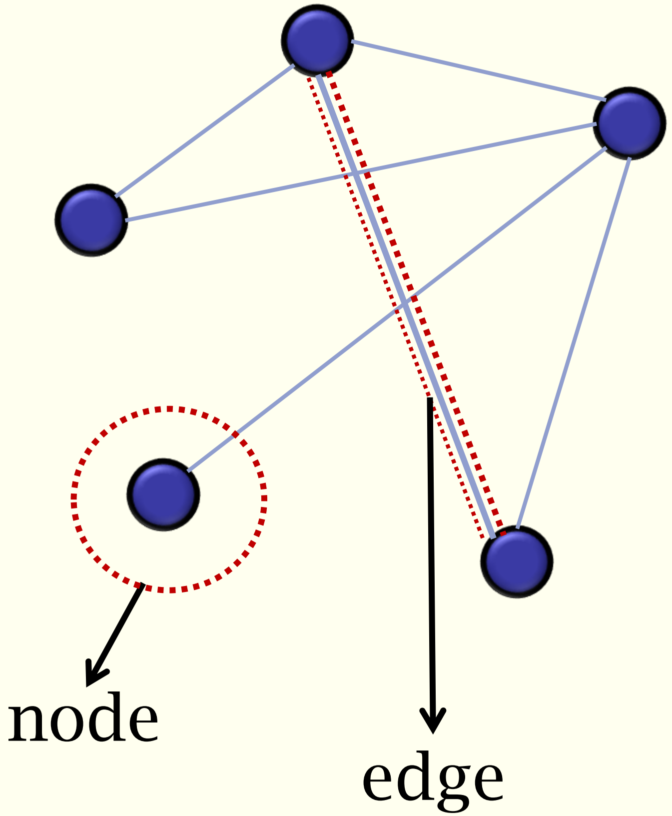
10- GRAPH THEORY AND NEUROSCIENCE

Learning objectives

1. Understand what is a graph and its possible representations
2. Remember the differences between binary, weighted, directed and undirected graphs
3. Describe the properties of adjacency matrices for different graph types
4. Remember the definition and meaning of the main graph indices at the local, global and meso-scales used in neuroscience

Graph Theory

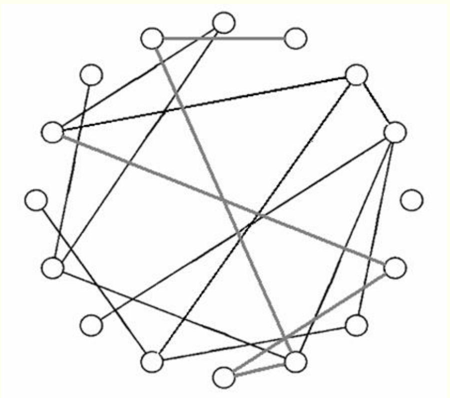
- Networks can be represented as graphs or as adjacency matrices
- Graphs have vertices (also called nodes), and edges that represent connections among vertices
- In network neuroscience, vertices correspond to brain regions or electrodes and edges correspond to some measure of connectivity for the EEG



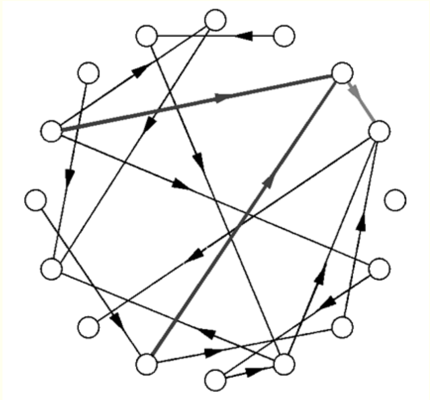
Graph properties

- **Directionality**
  - ➢ Undirected graphs
  - ➢ Directed graphs
- **Weight**
  - ➢ Binary graphs
  - ➢ Weighted graphs

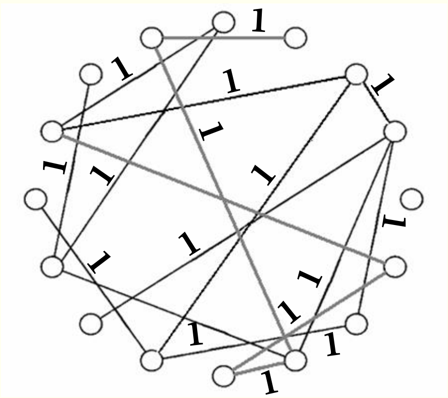
Undirected



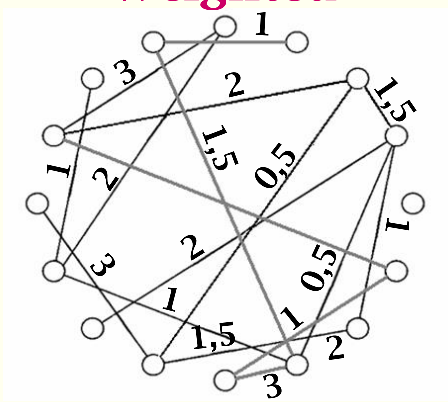
Directed



Binary

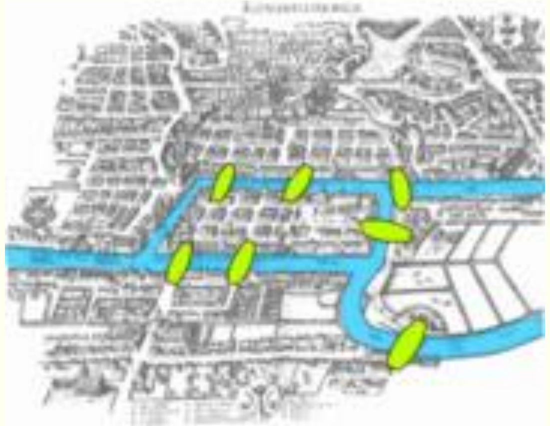


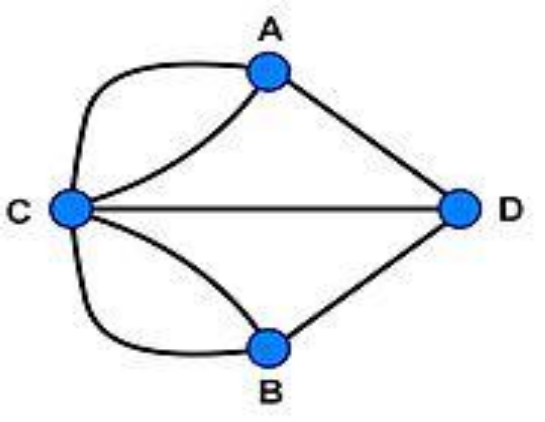
Weighted



Birth of graph theory

1786: Euler solved the problem of «the seven bridges of Königsberg”: *is there a walk through the city that would cross each of the bridges once and only once?*





Euler proved that the problem has no solution and laid the foundations of graph theory and topology.

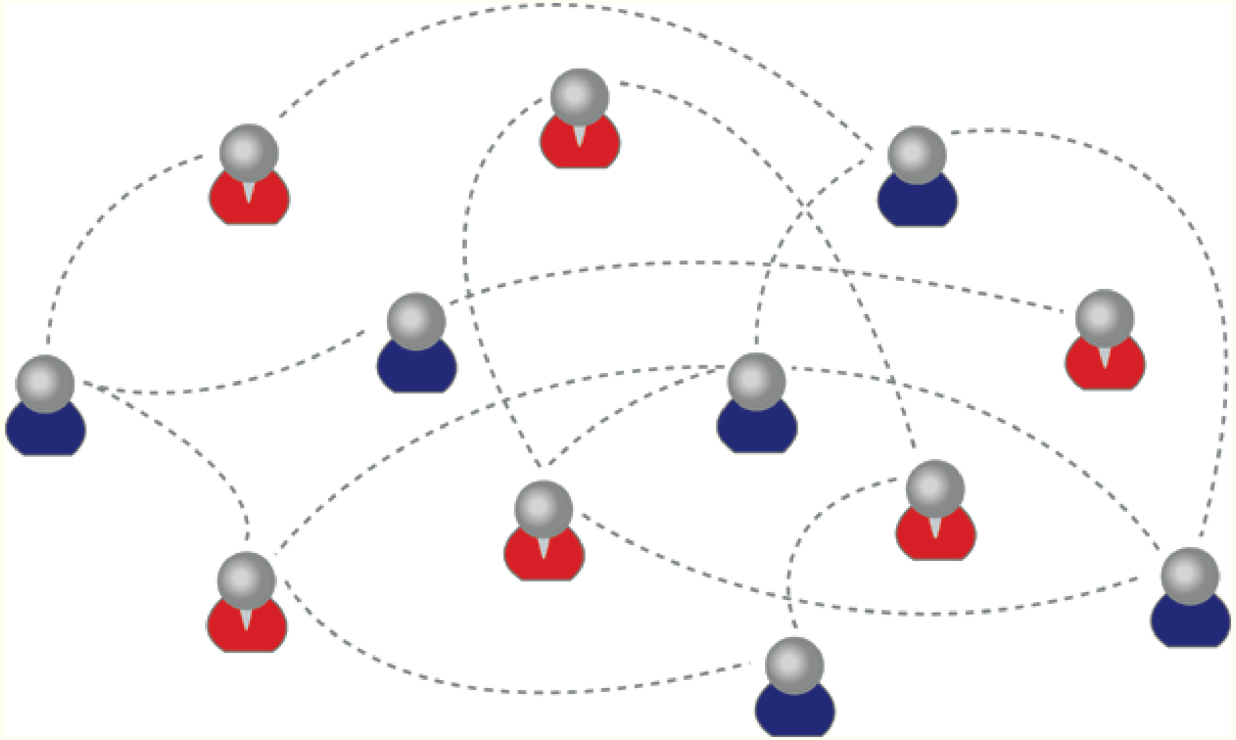
Examples of graphs

Social networks

nodes: people

edges: relations

Binary, undirected graph



Examples of graphs

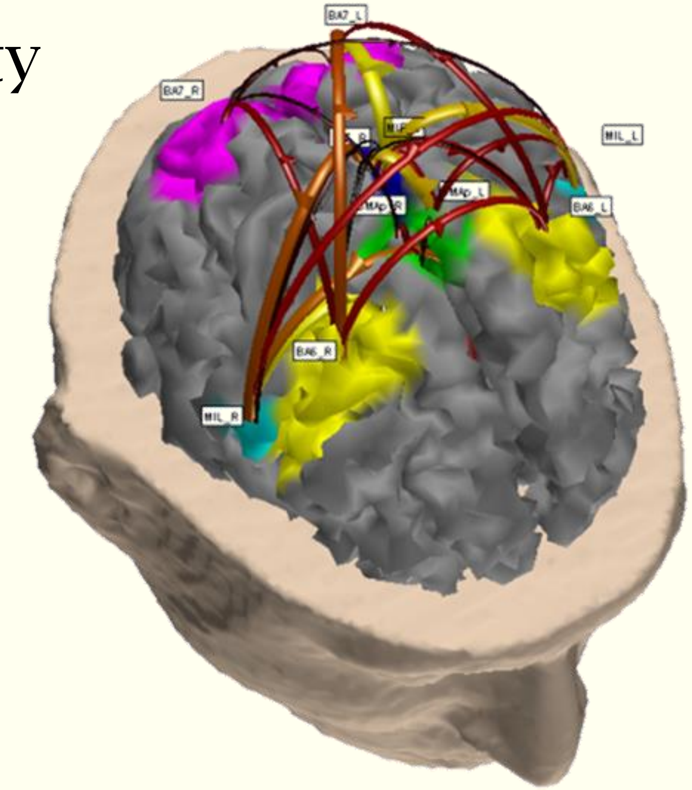
Highway networks
nodes: towns
edges: distances

Weighted, undirected graph

Examples of graphs

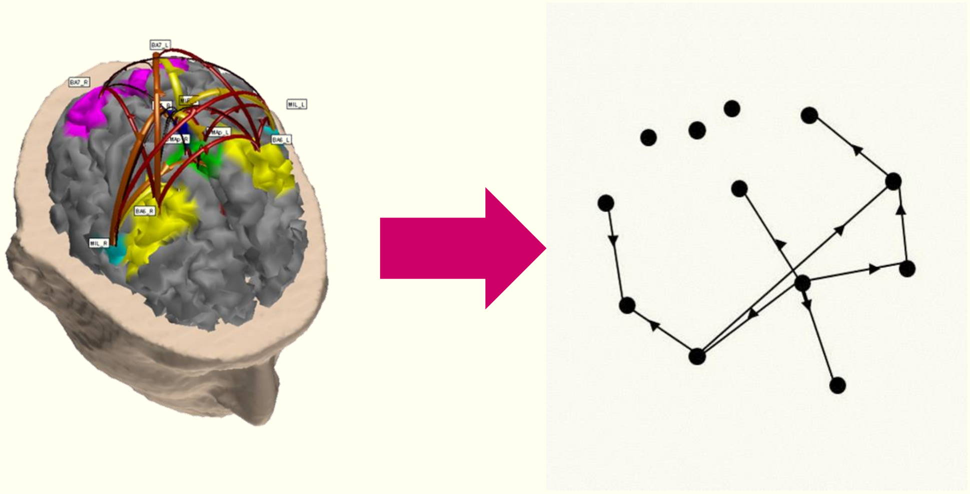
Brain networks
nodes: brain regions
edges: correlation or causality

Weighted, (un)directed graph



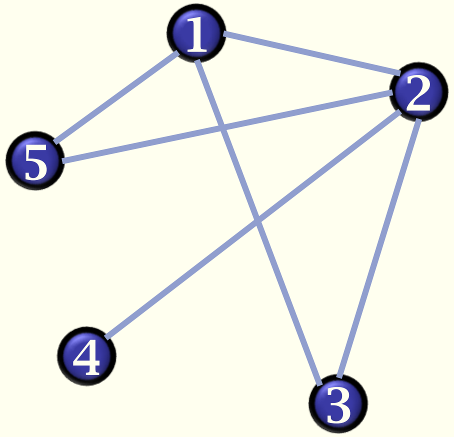
Need for graph analysis of brain networks

- Understand the network structure and properties at the local, global and meso- scales
- Extract quantifiable, objective, measurable and concise indices to be used:
  - For a statistical comparison between conditions, time points, groups of subjects…
  - As markers of specific pathological conditions (to support diagnosis, prognosis and evaluation of the effects of a clinical intervention)
  - As features for a classification aimed to decode the brain activity



the brain network can be represented by a graph which is graphical and mathematical object that we can use to compute and extract indices that describe some properties of the network (SLIDE 14)

Adjacency matrix

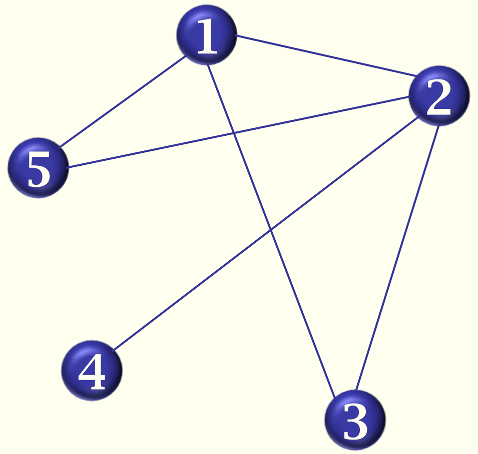


$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}$$

- $a_{ij} = 0$ if there is no link between i and j
- $a_{ij} \neq 0$ if there is a link between i and j
  - Binary graphs → $a_{ij} = 1$
  - Weighted graphs → $a_{ij} = w$ (w= weight of the edge)

Adjacency matrices

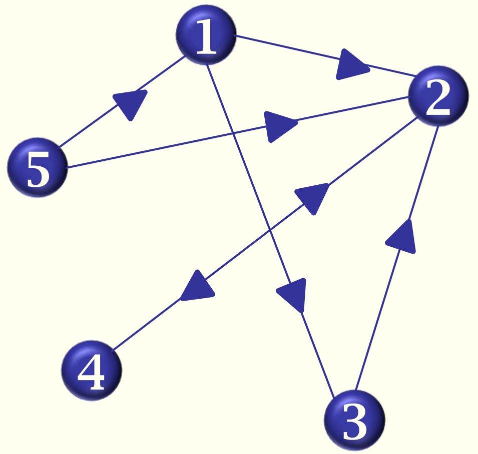
1) Binary undirected graph



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

Binary, symmetrical matrix

2) Binary directed graph



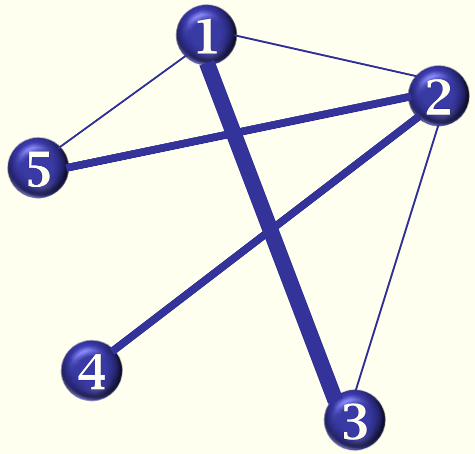
$$A=\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Binary, asymmetrical matrix

Adjacency matrices

1) Weighted undirected graph

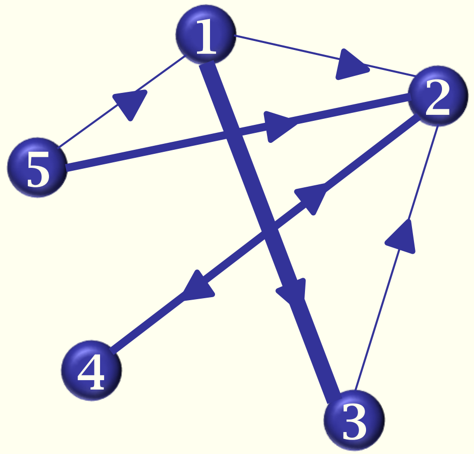
the meaning of these weights is strongly realted to the nature of the network and to the meaning of the network



$$A=\begin{bmatrix} 0 & 0.1 & 0.9 & 0 & 0.1 \\ 0.1 & 0 & 0.1 & 0.4 & 0.5 \\ 0.9 & 0.1 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 & 0 \\ 0.1 & 0.5 & 0 & 0 & 0 \end{bmatrix}$$

Weighted, symmetrical matrix

2) Weighted directed graph



$$A=\begin{bmatrix} 0 & 0.1 & 0.9 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0 \\ 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 & 0 \\ 0.1 & 0.5 & 0 & 0 & 0 \end{bmatrix}$$

Weighted, asymmetrical matrix

Graph indices commonly used in neuroscience

- Density
- Degree
- Distance
- Global Efficiency
- Local Efficiency
- Modularity
- Divisibility

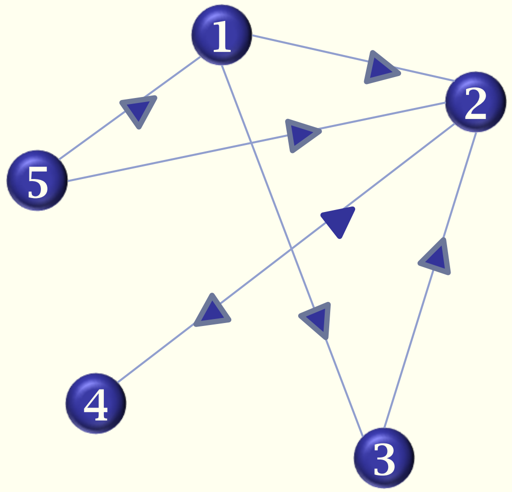
Density

Density ***k*** of a binary graph is the ratio between the number L of edges in the graph and the maximum possible number of edges L_{tot}

$$k = {}^{L}/_{L_{tot}}\ , \qquad k \in [0,1] \begin{cases} 0 & \text{no edges in the network} \\ 1 & \text{fully connected network} \end{cases}$$

Given an N-nodes graph:

- $L_{tot} = N * (N-1)/2$ for undirected graphs
- $L_{tot} = N * (N-1)$ for directed graphs



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

Example:

nodes

$N = 5$

$L = 7$

$L_{tot} = 5 * (5-1) = 20$

$k = {}^{L}/_{L_{tot}} = {}^{7}/_{20} = 0{,}35$

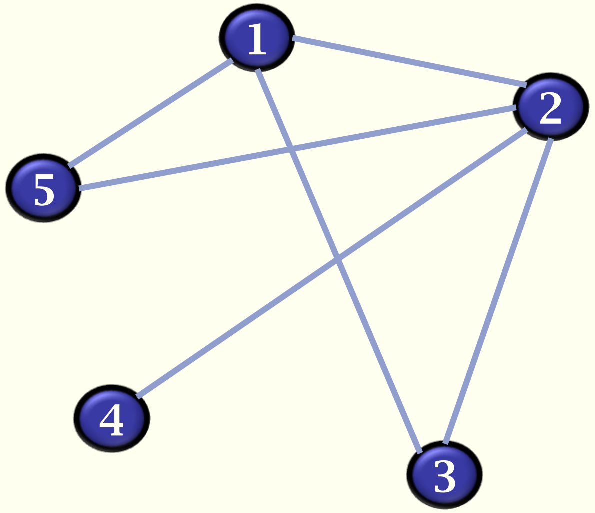
Degree

Undirected graphs:

The degree $\boldsymbol{g}$ of a node i ($i \in [1, N]$, N= number of nodes of the graph) is the number of edges connected to that node:

$$g_i = \sum_{j=1, i \neq j}^{N} a_{ij} \qquad g_i \in [0, N-1]$$

A node degree quantifies the role of a node in a network.
The higher the degree, the more involved the node.



Example

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

$g_1 = 3$
$g_2 = 4$
$g_3 = 2$
$g_4 = 1$
$g_5 = 2$

Degree

Directed graphs:

If the graph is directed, we can define:

In-degree of a node i ($i \in [1, N]$) as the number of edges directed to node i

$$g_i^{in} = \sum_{j=1, i\neq j}^{N} a_{ji} \qquad g_i^{in} \in [0, N-1]$$

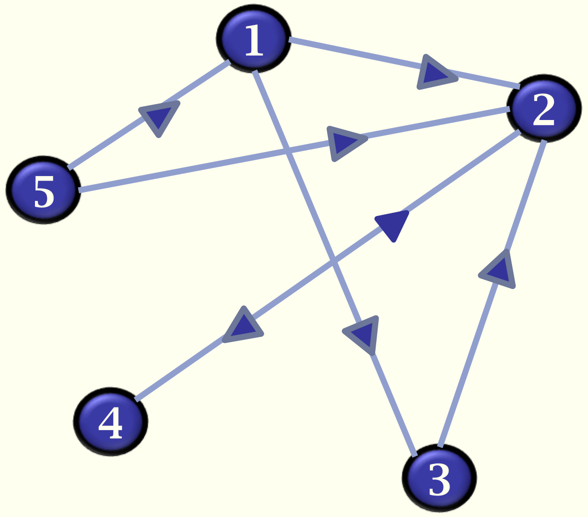
Out-degree of a node i ($i \in [1, N]$) as the number of edges originated from node i

$$g_i^{out} = \sum_{j=1, i\neq j}^{N} a_{ij} \qquad g_i^{out} \in [0, N-1]$$

Degree of a node i ($i \in [1, N]$) as the total number of edges originated from and directed to node i

$$g_i = g_i^{in} + g_i^{out} \qquad g_i \in [0, 2 * (N-1)]$$

Example



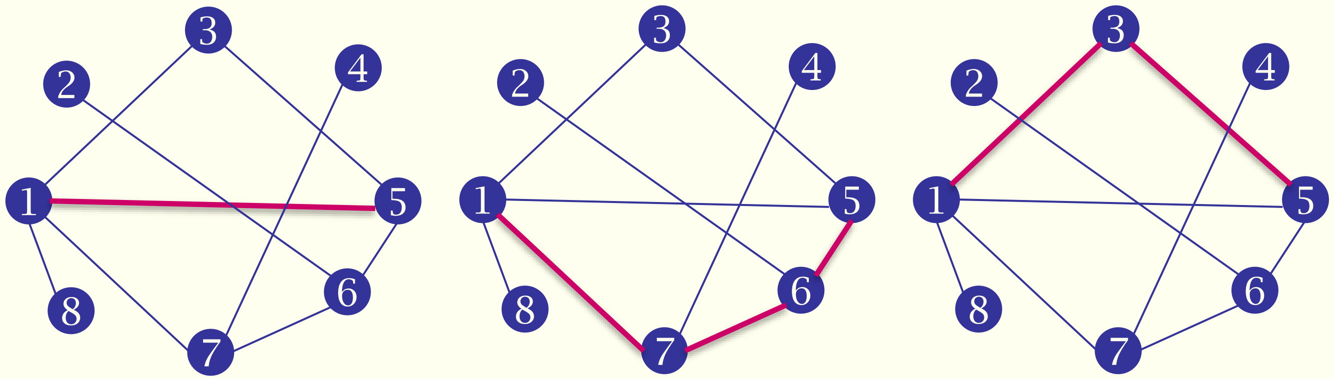
$$A=\begin{array}{c c}
\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} & \\
\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}
\end{array}$$

out-degree	in-degree	degree
$g_1{}^{out} = 2$	$g_1{}^{in} = 1$	$g_1 = 3$
$g_2{}^{out} = 1$	$g_2{}^{in} = 4$	$g_2 = 5$
$g_3{}^{out} = 1$	$g_3{}^{in} = 1$	$g_3 = 2$
$g_4{}^{out} = 1$	$g_4{}^{in} = 1$	$g_4 = 2$
$g_5{}^{out} = 2$	$g_5{}^{in} = 0$	$g_5 = 2$

to move to the next indeces we need to define first the distance between 2 nodes. This is not a physical distance but it is topological and logical distance. The distance between 2 nodes is the minimum path (any possible sequence of edges linking to nodes) length linking 2 nodes

Distance

Path: any possible sequence of edges linking two nodes, e.g.:

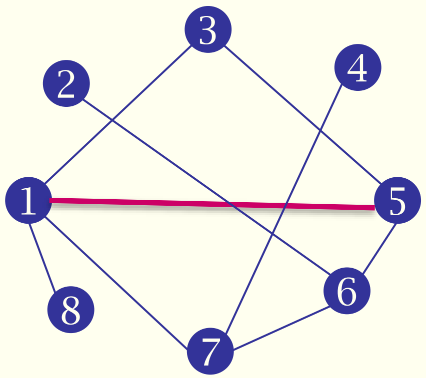


3 paths linking 1 and 5:

- **1** → **5** (1 step)
- **1** → 7 → 6 → **5** (3 steps)
- **1** → 3 → **5** (2 steps)

The path length is given by the number of edges (not by the physical distance)

Distance *d(i,j)* between nodes *i* and *j* is the shortest path between them:

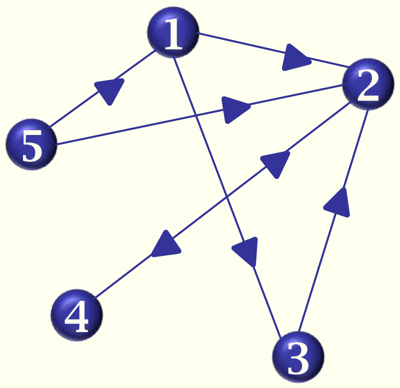


$d(1,5)=1$ → Shortest path between 1 and 5

Example

For directed graphs:

the 1 should be the same in A and D



$$A=\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} \rightarrow D=\begin{bmatrix} 0 & 1 & 1 & 2 & \infty \\ \infty & 0 & \infty & 1 & \infty \\ \infty & 1 & 0 & 2 & \infty \\ \infty & 1 & \infty & 0 & \infty \\ 1 & 1 & 2 & 2 & 0 \end{bmatrix}$$

matrix of distances

When no path links two nodes, the distance is infinite

Distance measures how efficient the interaction between two nodes is. The shortest the distance, the more efficient the interaction

Global Efficiency

global because we consider the whole network

The Global Efficency $\boldsymbol{E_g}$ of a graph is the average of the reciprocal of the distances between any pair of nodes (Latora e Marchiori, 2001)

average of the reciprocal of the distances

$$E_g = \frac{1}{N(N-1)} \sum_{i,j=1, i \neq j}^{N} \frac{1}{d_{ij}} \,, \qquad E_g \in [0,1]$$

when dij = inf, then 1/dij = 0

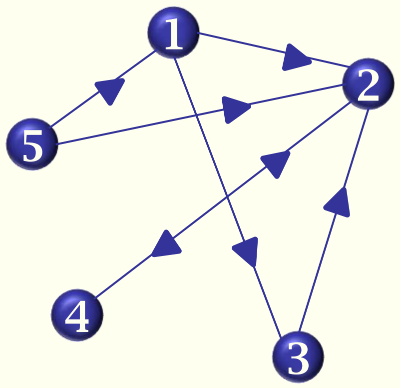
max possible number of links of edges that we can have in our network

$E_g = 1 \rightarrow$ graph fully connected

$E_g = 0 \rightarrow$ void graph

Global Efficiency measures how efficiently the information is exchanged in the network

Example



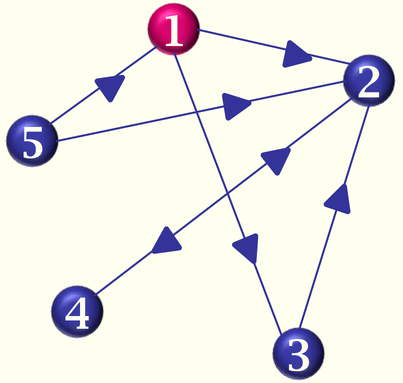
$$A=\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} \qquad D=\begin{bmatrix} 0 & 1 & 1 & 2 & \infty \\ \infty & 0 & \infty & 1 & \infty \\ \infty & 1 & 0 & 2 & \infty \\ \infty & 1 & \infty & 0 & \infty \\ 1 & 1 & 2 & 2 & 0 \end{bmatrix}$$

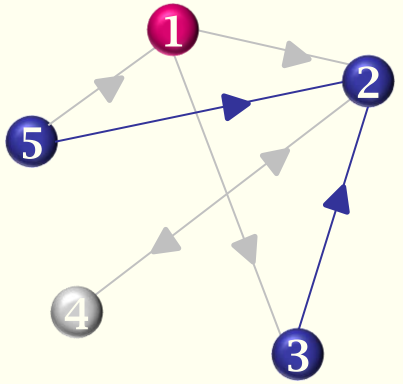
$$E_g = \frac{1}{N(N-1)} \sum_{i,j=1,i\neq j}^{N} \frac{1}{d_{ij}} = \frac{1}{5(5-1)}\left[1+1+\frac{1}{2}+1+1+\frac{1}{2}+1+1+1+\frac{1}{2}+\frac{1}{2}\right] = 0{,}45$$

(Average) Local Efficiency

it's a global measure of local properties of the network

For each node i we can extract a subnetwork S_i made of all the nodes directly connected to i (but not including i itself) and compute its Global Efficiency $E_g(S_i)$.





S_1

here we look what happen to the network when we remove a node. Since we want to know what is the robustness of the network to node 1, we remove the node 1 and we look what happens to the node linked to node 1. With efficiency we measure the linking between the nodes without the removed node

This is a measure of the robustness of the network to the removal of i.

The average of $E_g(S_i)$ across all the nodes is called (average) Local Efficiency (Latora e Marchiori, 2001)

on this subnetwork we compute the global efficiency of each network we create removing node

$$E_l = \frac{1}{N} \sum_{i=1}^{N} E_g(S_i)$$

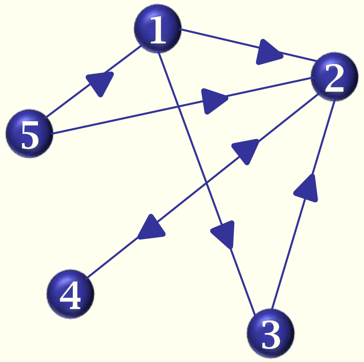
we have N of this

number of nodes

Example

the node in grey are the nodes that are not part of the subgraph, while the nodes in blue belongs to the subgraph. For ex. in the fisrt picture node 4 is in grey because it is not connected to node 1, while in picture 2 the node 4 is blue because it was linked to the node 2, but it is isolate. However it have to included in the subgraph; but the efficiency computed by considering node 4 or not is very different (if we consider node 4 and compute the efficiency the result is a very bad efficiency of the network, becuse the node not communicate with any other)

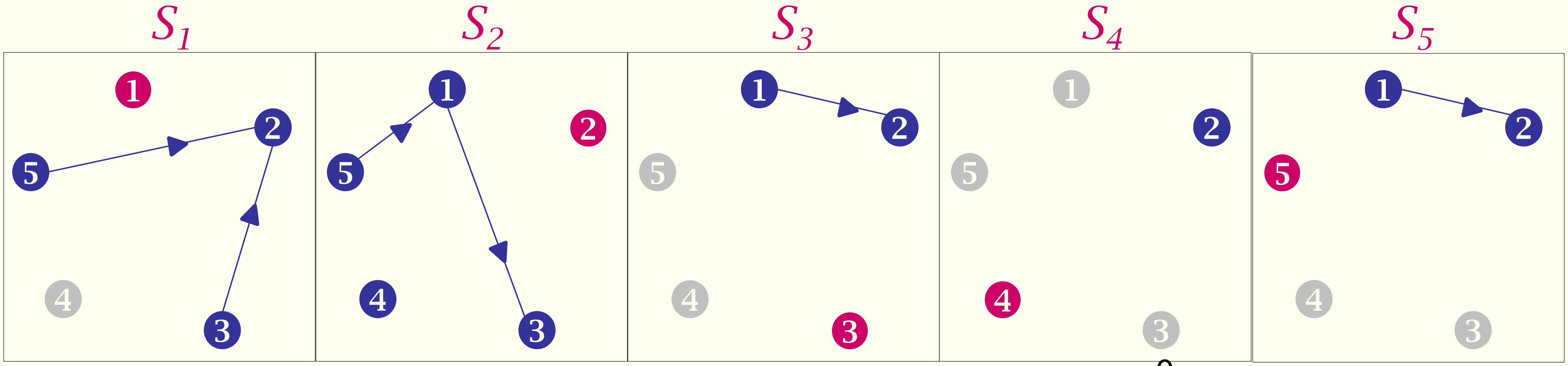
N=5



A graph with N nodes has N subgraphs:

$$E_l = \frac{1}{N}\sum_{i=1}^{N} E_g(S_i) = 0.408$$

this can be wrong because the prof condidered 1/2 rather than 0



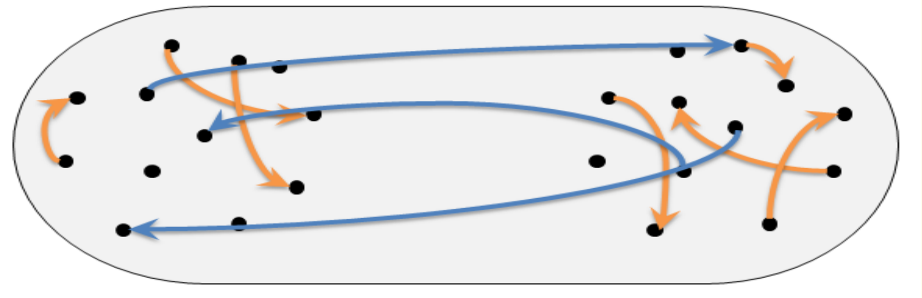
$E_g(S_1) = 1/3$ $E_g(S_2) = 5/24$ $E_g(S_3) = 1/2$ $E_g(S_4) = \cancel{1/2}$ 0 $E_g(S_5) = 1/2$

given a network and selected 2 or more subnetworks I obtain 2 or more community (group of nodes)

Modularity

in this case we focus on the interconnection (yellow arrows)

For some applications it may be interesting to measure ahow well it can be divided into two communities:



the 2 emisphere of the brain communicate, if we remove this connection the can't

Modularity Q measures the tendency of the subnetworks to form communities:

Undirected graphs

we consider this number only when the 2 nodes belong to the same community

$$Q = \frac{1}{L} \sum_{i,j=1}^{N} \left(a_{ij} - \frac{k_i k_j}{L}\right) \delta(C_i, C_j)$$

Directed graphs

$$Q = \frac{1}{L} \sum_{i,j=1}^{N} \left(a_{ij} - \frac{k_i^{OUT} k_j^{IN}}{L}\right) \delta(C_i, C_j)$$

L : number of edges in the network
a_{ij}: elements of the adjacency matrix
k_i: degree of node i

C_i: community to which node i belongs

$\delta(C_i, C_j)$ → 1 if $C_i = C_j$ → if i and j belong to the same community
→ 0 otherwise

Divisibility

in this case we focus on the extraconnection (blue arrows in the previous slide)

it is higher with lower number of intercommunity link. 0 links is perfect

Divisibility D is a measure of the segregation between two communities:

$$D = \frac{L}{\sum_{i,j=1}^{N} a_{ij}\left[\boxed{1-\delta\left(C_i,C_j\right)}\right]+k} = \frac{L}{L+\sum_{i,j=1}^{N} a_{ij}\left[1-\delta\left(C_i,C_j\right)\right]}$$

when the 2 nodes belong to the same communiti this is 0. So in this formula I consider only the nodes that belong to different community

L : number of edges in the network

a_{ij}: elements of the adjacency matrix

k_i: degree of node i

C_i: community to which node i belongs

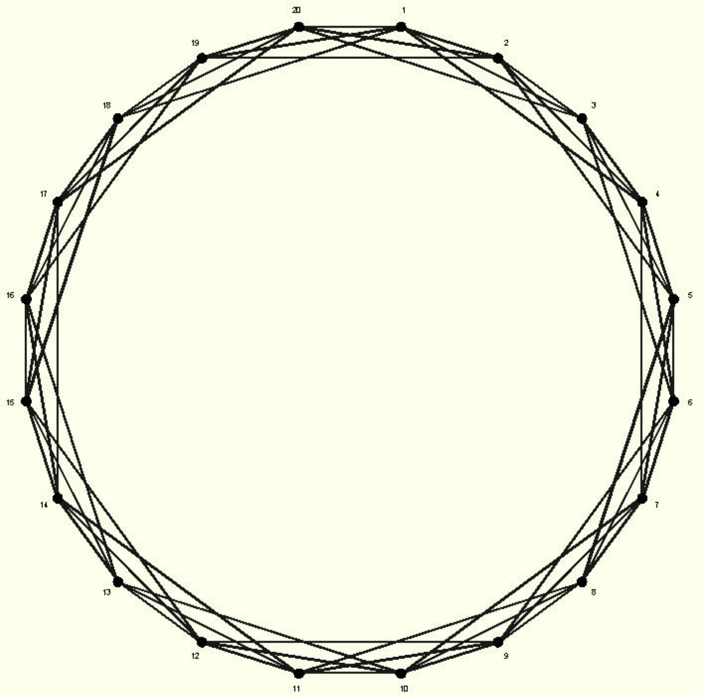
$\delta(C_i, C_j)$ → 1 if $C_i = C_j$ → 0 otherwise

k: is a constant (to avoid divergence), usualli it's equal to L so that $D \in \left[\frac{1}{2}, 1\right]$

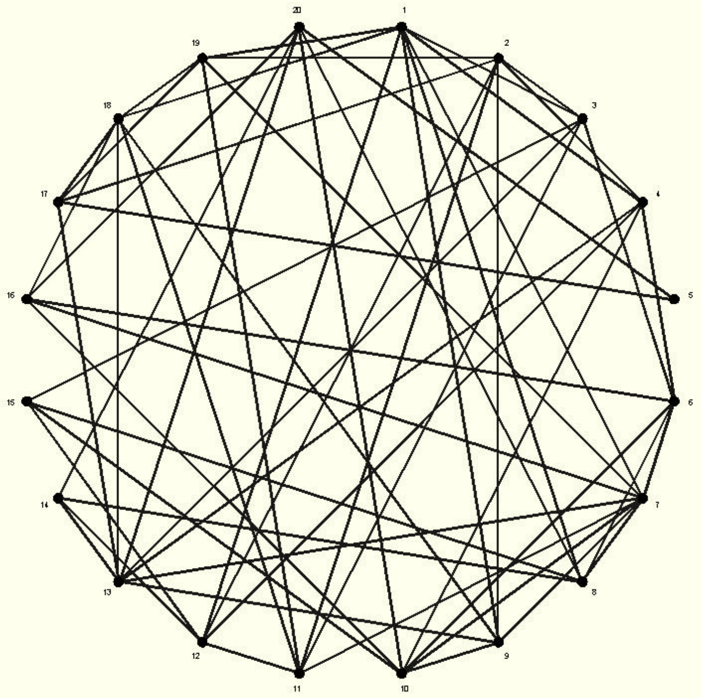
Reference networks

Two reference structures:

Regular

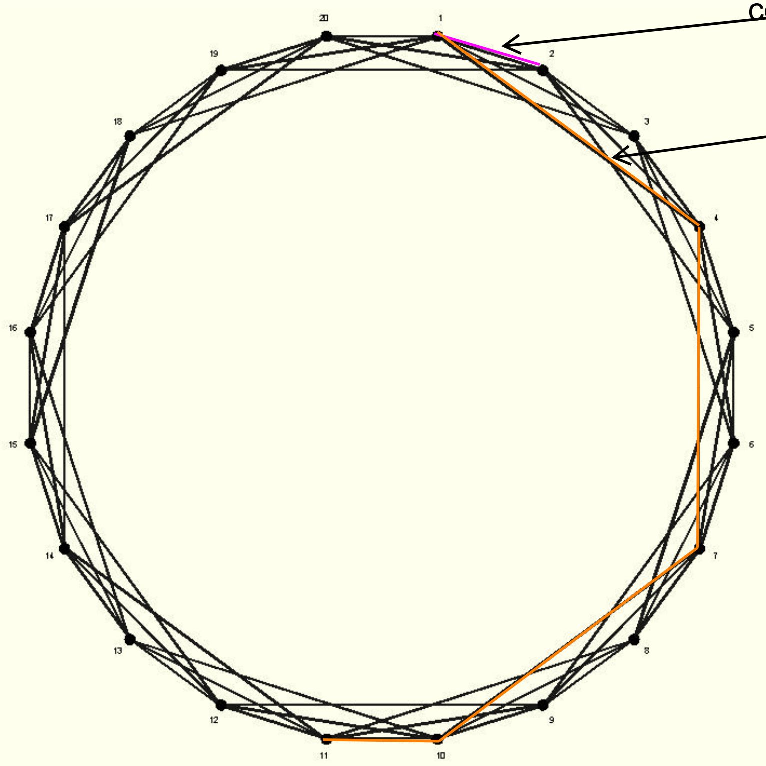


Random



Regular networks

we have a very strong efficient communication between close nodes and a very unefficient communications between far nodes. far and close are logical distances

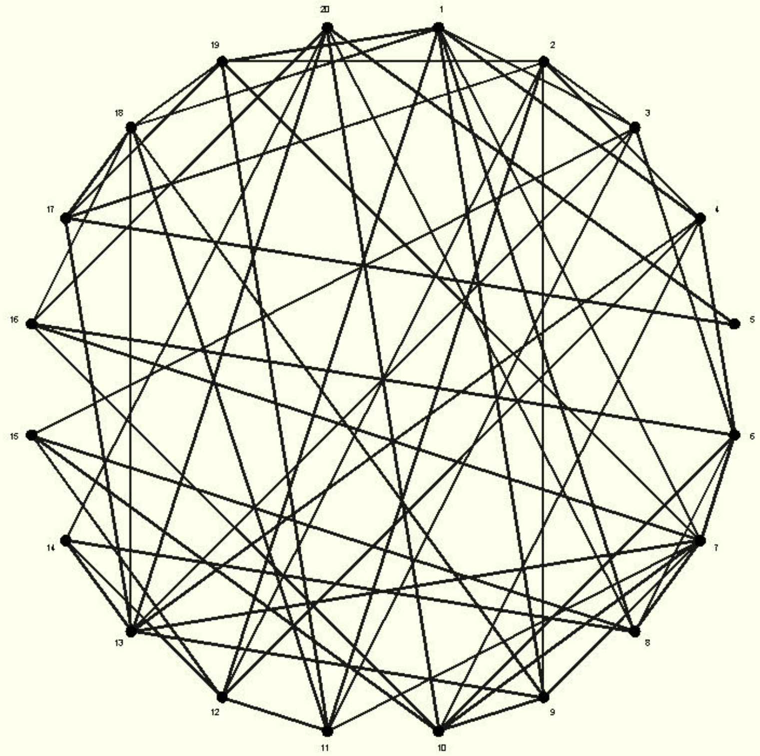


Each node is linked to a small number of other nodes.
The degree is the same for all nodes.
Efficient communications between small groups, unefficient communications at the entire network level

Lattice structure

- LOW Global Efficiency
- HIGH Local Efficiency

Random ~~Regular~~ networks



Each node is linked to the others randomly. There are no small groups with a strong internal communication. (Erdòs Rènyi, 1959)

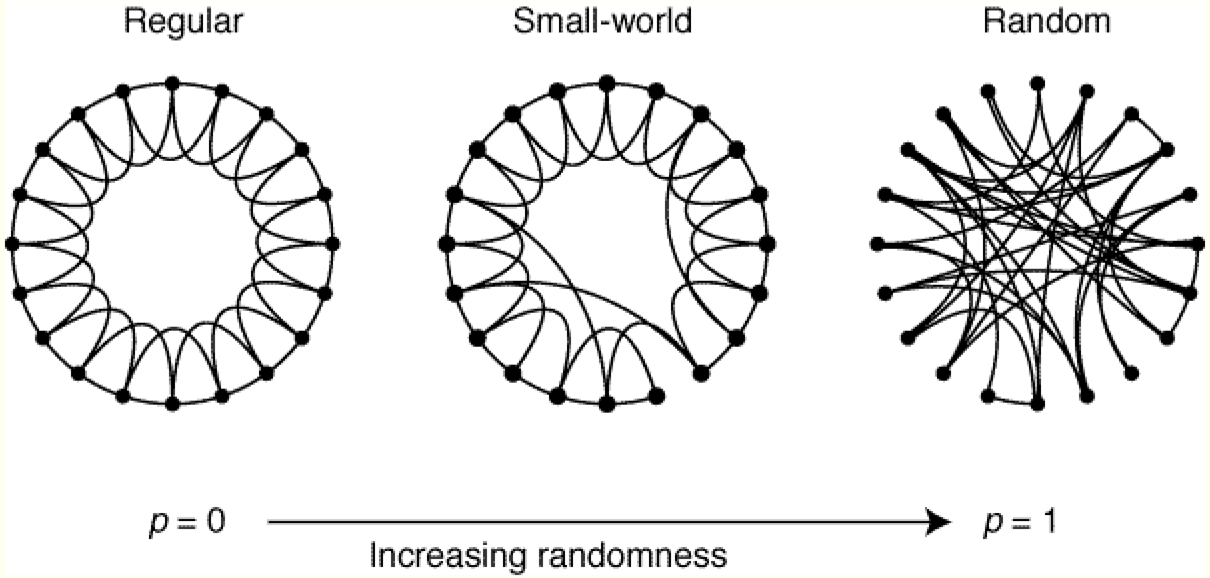
If we are looking for small groups or organized nodes every time we remove a node we will demage the entire network.

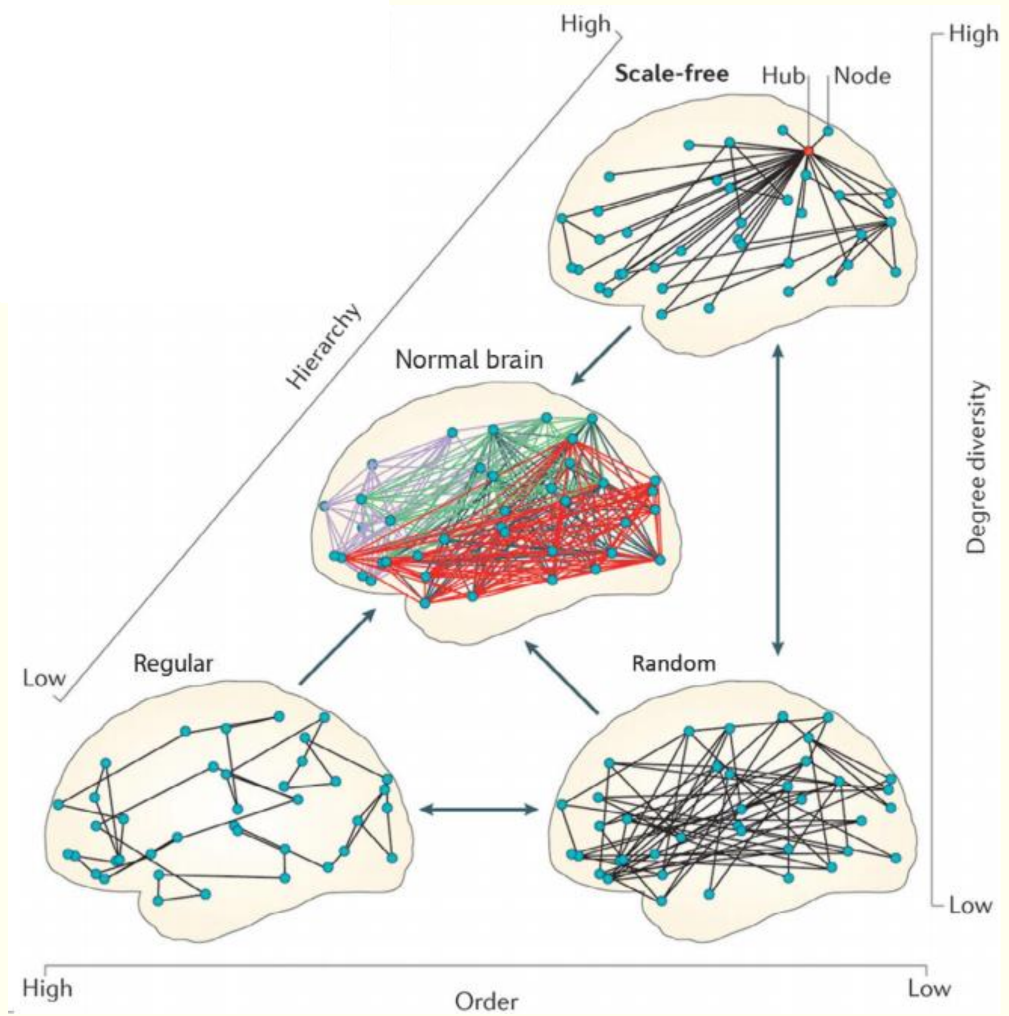
Lattice structure

- HIGH Global Efficiency
- LOW Local Efficiency

Small-world networks

Real networks are neither like the random nor like the regular graphs:





Eg regular < **Eg** < Eg random

El random << **El** < El regular

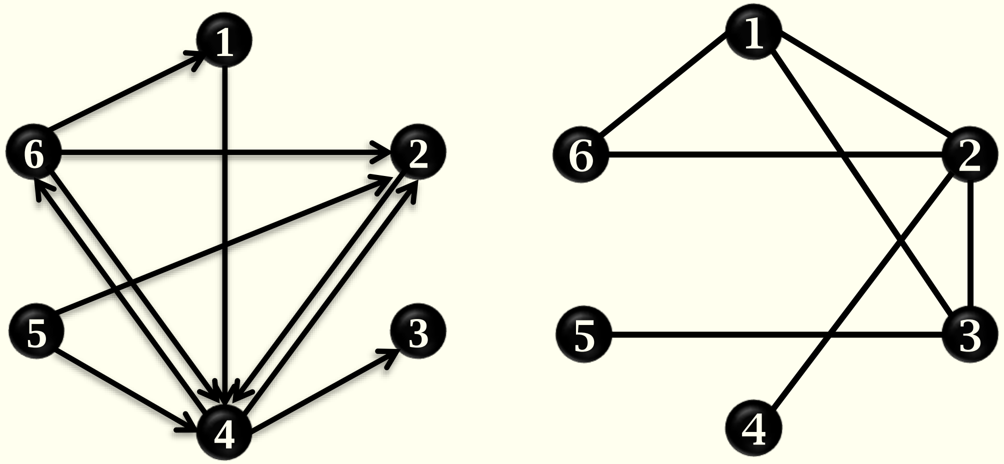
(Watts and Strogatz, Nature, 1998)

References

- Cohen, Chapter 31

Self-evaluation

1. Given the following graphs:



2. Write down their adjacency matrices
3. Compute their densities
4. Compute their degrees
5. Write down their distance matrices
6. Compute their Global Efficiency