Session I (9 am – 10.30 am)

UNIVERSITÉ de Cergy-Pontoise

UNIVERSITÉ DE NAMUR

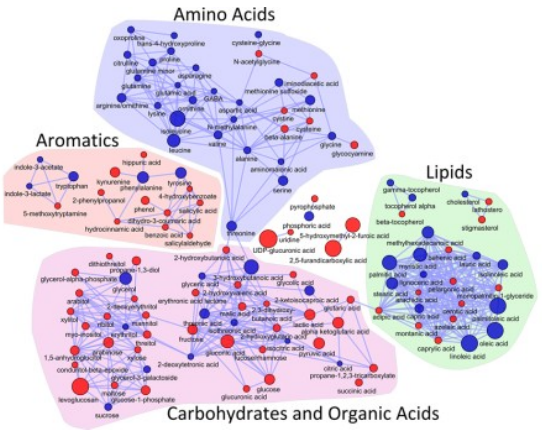
Introduction to Network Science

By Yérali Gandica

IC²S²

5th International Conference on Computational Social Science
July 17-20, 2019, University of Amsterdam, The Netherlands

I. Main types of networks found in nature

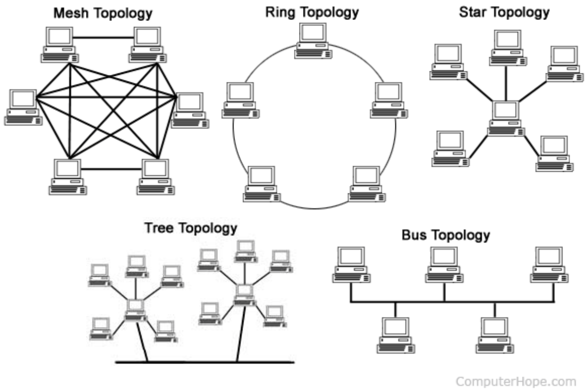


Example of **biological networks**: Nodes represent metabolites and edges can be many things.

Source: https://www.r-bloggers.com/tutorial-building-biological-networks/

Telecommunication networks:

Collection of computers, servers, mainframes, network devices.

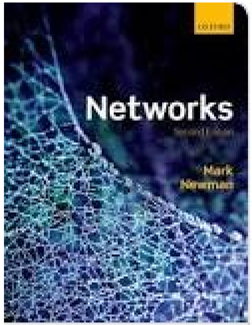


Source: https://www.computerhope.com/jargon/n/network.htm

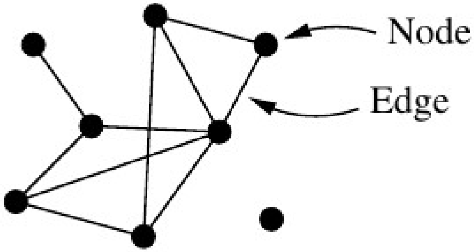
Social networks:

Facebook, Twitter, Wikipedia, etc.

Source: https://blog.stateofthedapps.com/blockchain-based-social-networks-1b7b729beb4d



I. Main types of networks found in nature



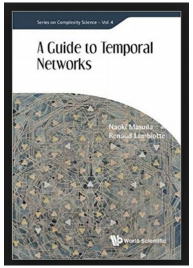
Network	Node	Edge
Internet	Computer or router	Cable or wireless data connection
World Wide Web	Web page	Hyperlink
Citation network	Article, patent, or legal case	Citation
Power grid	Generating station or substation	Transmission line
Friendship network	Person	Friendship
Metabolic network	Metabolite	Metabolic reaction
Neural network	Neuron	Synapse
Food web	Species	Predation

Session I (9 am – 10.30 am)

I. Main types of networks found in nature

Which other examples can we think of?

Session I (9 am – 10.30 am)



Definitions and basic properties:

A **network**: System made of nodes connected by links (they can be undirected or directed, and unweighted or weighted).

In the mathematical literature, a network is called a **graph: G** = (V, E).

V: Set of nodes (also called vertices).

E: Set of links (also called edges).

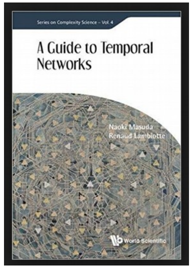
Each link is defined by a pair of nodes, e = (v, v’) ∈ E.

In the case of undirected networks, the order of v and v’ does not matter.

In the case of directed networks, (v, v’) is a link from v to v’.

If (v, v’) ∈ E and (v’, v) ∈ E, the two nodes are reciprocally connected.

In the case of weighted networks, links are also assigned with a weight function, characterising the importance or weight of the link.



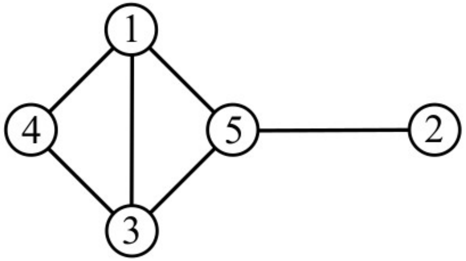
Definitions and basic properties:

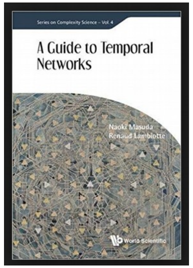
Two main representations of networks

- $A_{N \times N}$ adjacency matrix. $A_{i,j} = 1$ if there is a link between nodes I and j.

- If the network is weighted, A ij can take positive values $\neq 1$, representing the weight of the link.

- In general, undirected and directed networks will yield symmetric and asymmetric adjacency matrices, respectively.





Session I (9 am – 10.30 am)

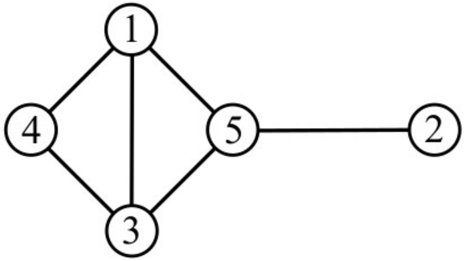
Definitions and basic properties:

Two main representations of networks

- $A_{N \times N}$ adjacency matrix. $A_{i,j} = 1$ if there is a link between nodes I and j.

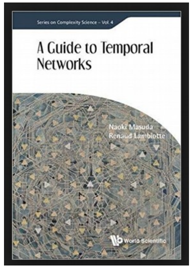
- If the network is weighted, A ij can take positive values ≠ 1, representing the weight of the link.

- In general, undirected and directed networks will yield symmetric and asymmetric adjacency matrices, respectively.



$$A_{N \times N} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Session I (9 am – 10.30 am)



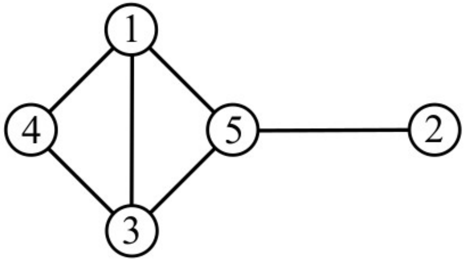
Definitions and basic properties:

Two main representations of networks

- $A_{N \times N}$ adjacency matrix. $A_{i,j} = 1$ if there is a link between nodes I and j.

- If the network is weighted, A ij can take positive values $\neq 1$, representing the weight of the link.

- In general, undirected and directed networks will yield symmetric and asymmetric adjacency matrices, respectively.

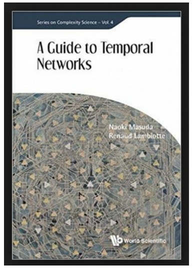


$$A_{N \times N} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

A representation of static networks alternative to the adjacency matrix is called the **link list**: A graph is described as a list of pairs of nodes

Session I (9 am – 10.30 am)

Definitions and basic properties:



1.2.1: Degree distribution: Number of links incident to a node. We denote the degree of the ith node by k i . For undirected networks, the degree is given by

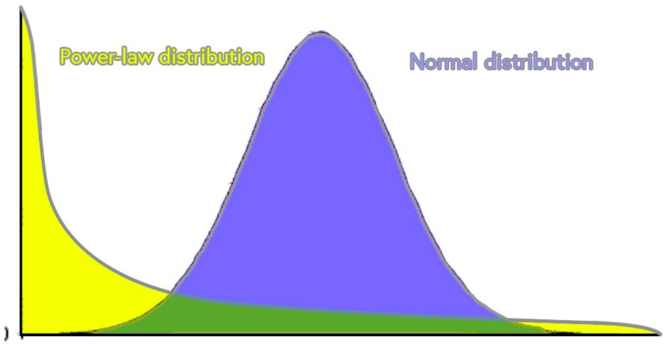
$$k_i = \sum_{j=1}^{N} A_{ij} \left(= \sum_{j=1}^{N} A_{ji} \right)$$

A network is called regular if all nodes have the same degree, i.e., $k_i = k_j$ for all i and j.

For directed networks, we distinguish the in-degree (number of links incoming to the node), and the out-degree (number of links outgoing from the node).

The degree distribution of a network is the frequency distribution of the degree and denoted by p(k). A majority of networks in different domains possesses long-tailed degree distributions. In many situations, their tail is described by a power-law:

$$p(k) \propto k^{-\gamma},$$



How can we transform a distribution with a bounded variance into one without characteristic scale? → Let's think of an example!!!

How can we transform a distribution with a bounded variance into one without characteristic scale? → Let's think of an example!!!

Suggestion:

Starting point: Distribution of student's height in the classroom:

Distribution of everything's height:

Definitions and basic properties:

1.2.2: **Average path length**: Average number of steps along the shortest path for all possible pairs of network nodes. It is a measurement of the efficiency of information or mass transport on a network.

The distance between nodes v_i and v_j , $d(v_i , v_j)$, is defined as the smallest number of jumps in a path necessary to go from v_i to v_j

$$L = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{i-1} d(v_i, v_j).$$

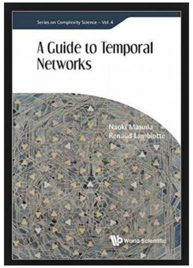
6 degrees of separation: https://www.youtube.com/watch?v=a99ry70CnRs

Another example for later on:

https://www.youtube.com/watch?v=8j1wX5Nznro

Session I (9 am – 10.30 am)

Definitions and basic properties:



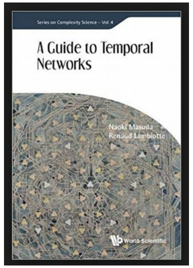
1.2.3: **Clustering coefficient**: it quantifies the amount of triangles in a network.

$$C \equiv \frac{1}{N} \sum_{i=1}^{N} C_i.$$

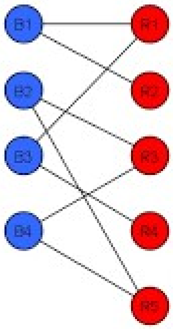
Where the node's clustering:

$$C_i \equiv \frac{\text{number of triangles including the } i\text{th node}}{k_i(k_i - 1)/2},$$

Session I (9 am – 10.30 am)



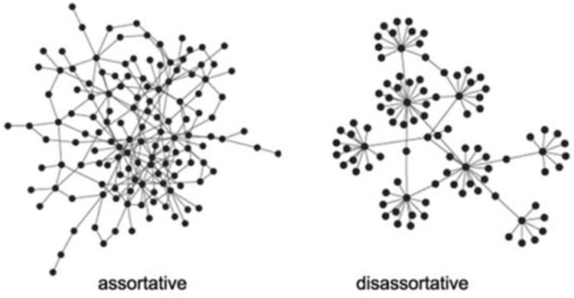
Particularities of some networks



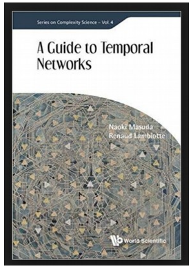
Bi-partite networks: Complex networks, whose nodes are divided into two sets X and Y, and only connections between two nodes in different sets are allowed. Examples?

Assortativity Vs. disassortativity:
Assortativity: when there is a preference for a network's nodes to attach to others with similar degree. Examples?

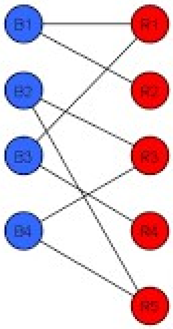
Disassortativity, when high degree nodes tend to attach to low degree nodes. Technological and biological networks typically show disassortative mixing. Examples?



Homophily: When nodes have the tendency to connect with similar ones



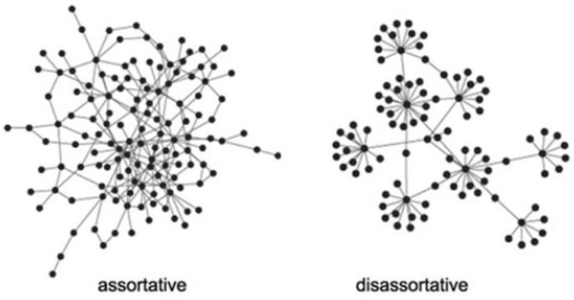
Particularities of some networks



Bi-partite networks: Complex networks, whose nodes are divided into two sets X and Y, and only connections between two nodes in different sets are allowed. Examples?

Assortativity Vs. disassortativity:

Assortativity: when there is a preference for a network's nodes to attach to others with similar degree. Ex. social networks.



Disassortativity, when high degree nodes tend to attach to low degree nodes. Technological and biological networks typically show disassortative mixing. Ex. technological and biological networks

Homophily: When nodes have the tendency to connect with similar ones

Particularities of some networks

Size n and assortativity coefficient r for various networks

Network	n	r
Physics coauthorship (a)	52 909	0.363
Biology coauthorship (a)	1 520 251	0.127
Mathematics coauthorship (b)	253 339	0.120
Film actor collaborations (c)	449 913	0.208
Company directors (d)	7 673	0.276
Internet (e)	10 697	−0.189
World-Wide Web (f)	269 504	−0.065
Protein interactions (g)	2 115	−0.156
Neural network (h)	307	−0.163
Marine food web (i)	134	−0.247
Freshwater food web (j)	92	−0.276
Random graph (u)		0
Callaway *et al.* (v)		$\delta/(1+2\delta)$
Barabási and Albert (w)		0

Source: https://en.wikipedia.org/wiki/Assortativity

Networks Visualizations

https://networkx.github.io/ → Python module

https://graph-tool.skewed.de/ → Python module (running in C)
Interactive visualization

https://igraph.org/r/ → R

https://www.smrfoundation.org/nodexl/ → Microsoft Excel

http://mrvar.fdv.uni-lj.si/pajek/ → Pajek for all OS

Let us check whether everyone has installed:

- Python 2.7. You can also use Anaconda Python.

- NetworkX pip install networkx==2.2 (https://pypi.org/project/networkx/2.2/)

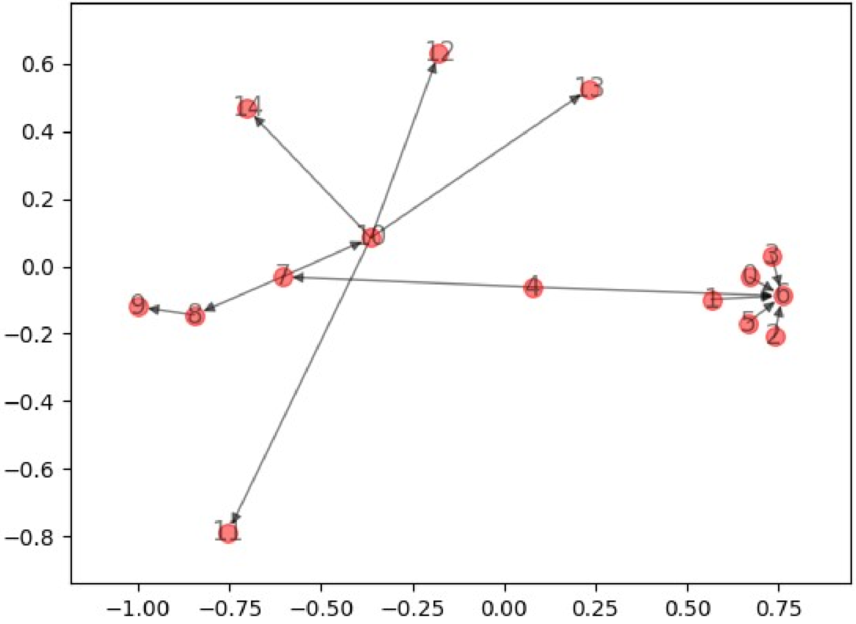
sudo apt install python-pip → if you haven' t installed pip yet

- pip install matplotlib

- pip install numpy

sudo apt-get install python-tk (Linux)

Let us visualize a small network



Closeness centrality: The inverse of the mean distance from node in question to any other node in the network. The closeness centrality is well-defined only for connected networks

$$\text{closeness}_i = \frac{N-1}{\sum_{j=1; j \neq i}^{N} d(v_i, v_j)},$$

Betweenness centrality: is defined as the fraction of the shortest paths passing through the node in question. This quantity is averaged over all possible pairs of nodes.

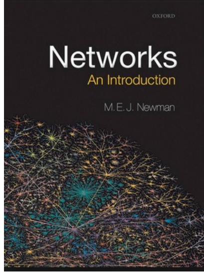
$$\text{betweenness}_i = \frac{2}{(N-1)(N-2)} \sum_{j=1;j\neq i}^{N} \sum_{\ell=1;\ell\neq i}^{j-1} \frac{\sigma_{j\ell}^{i}}{\sigma_{j\ell}},$$

where $\sigma_{j\ell}$ is the number of the shortest paths connecting the jth and ℓth nodes, and $\sigma_{j\ell}^{i}$ is the number of such shortest paths that pass through the ith node.

The summation excludes the shortest paths that start or end at the ith node.

The normalisation factor 2/ [(N − 1)(N − 2)] comes from the combinations of j and ℓ.

Session I (9 am – 10.30 am)



Katz centrality: It measures the relative degree of influence of a node within a network. Katz centrality takes into account that the importance of a node is increased by having connections to other vertices that are themselves important. It measures influence by taking into account the total number of walks between a pair of nodes.

$$\mathrm{Katz}_i = \sum_{j=1}^{N} \left[(I - \alpha A)^{-1} \right]_{ij} .$$

Now let us calculate the centrality measures on our small network !!!!

Now let us check the packages for the Louvain Method:

Coffee-break

- Community detection for NetworkX's (louvain method)
https://python-louvain.readthedocs.io/en/latest/

https://github.com/taynaud/python-louvain → clone or download → extract → python setup.py install Or

pip install python-louvain

Repositories for networks:

http://konect.cc/

https://snap.stanford.edu/data/

http://networkrepository.com/network-data.php