Network Science Cheatsheet

UNIVERSITÉ DE LYON

Made by
Remy Cazabet

Network Science - Introduction

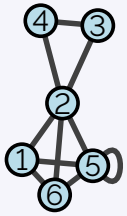
Networks: Graph notation

Graph notation : $G = (V, E)$

V	set of vertices/nodes.
E	set of edges/links.
$u \in V$	a node.
$(u, v) \in E$	an edge.

Network - Graph notation

Graph



Graph notation

$$G = (V, E)$$
$$V = \{1, 2, 3, 4, 5, 6\}$$
$$E = \{(1, 2), (1, 6), (1, 5), (2, 4), (2, 3), (2, 5), (2, 6), (6, 5), (5, 5), (4, 3)\}$$

Types of networks

Simple graph: Edges can only exist or not exist between each pair of node, and there are no self-loops, i.e., an edge connecting a node to itself.
Directed graph: Edges have a direction: $(u, v) \in V$ does not imply $(v, u) \in V$
Weighted graph: A weight is associated to every edge.

Other types of graphs (multigraphs, multipartite, hypergraphs, etc.) are introduced later

Counting nodes and edges

N, n	**size**: number of nodes $\|V\|$.
L, m	number of edges $\|E\|$
L_{max}	Maximum number of links

Undirected network: $\binom{N}{2} = N(N-1)/2$

Directed network: $= N(N-1)$

Node-Edge description

N_u	**Neighbourhood** of u, nodes sharing a link with u.
k_u	**Degree** of u, number of neighbors $\|N_u\|$.
N_u^{out}	**Successors** of u, nodes such as $(u, v) \in E$ in a directed graph
N_u^{in}	**Predecessors** of u, nodes such as $(v, u) \in E$ in a directed graph
k_u^{out}	**Out-degree** of u, number of outgoing edges $\|N_u^{out}\|$.
k_u^{in}	**In-degree** of u, number of incoming edges $\|N_u^{in}\|$
$w_{u,v}$	**Weight** of edge (u, v).
s_u	**Strength** of u, sum of weights of adjacent edges, $s_u = \sum_v w_{uv}$.

Network descriptors - Nodes/Edges

$\langle k \rangle$	**Average degree**: Real networks are sparse, i.e., typically $\langle k \rangle \ll n$. Increases slowly with network size, e.g., $\langle k \rangle \sim \log(n)^a$

$$\langle k \rangle = \frac{2m}{n}$$

$d, d(G)$	**Density**: Fraction of pairs of nodes connected by an edge in G.

$$d = L/L_{\max}$$

[a] Leskovec, J. Kleinberg, and C. Faloutsos 2005.

Paths - Walks - Distance

Walk: Sequences of adjacent edges or nodes (e.g., **1.2.1.6.5** is a valid walk)
Path: a walk in which each node is distinct.
Path length: number of **edges** encountered in a path
Weighted Path length: Sum of the weights of edges on a path
Shortest path: The shortest path between nodes u, v is a path of minimal *path length*. Not necessarily unique.
Weighted Shortest path: path of minimal *weighted path length*.
$\ell_{u,v}$: **Distance**: The distance between nodes u, v is the length of the shortest path

Network descriptors - Paths

$\ell_{\max}$	**Diameter**: maximum *distance* between any pair of nodes.
$\langle \ell \rangle$	**Average distance**:

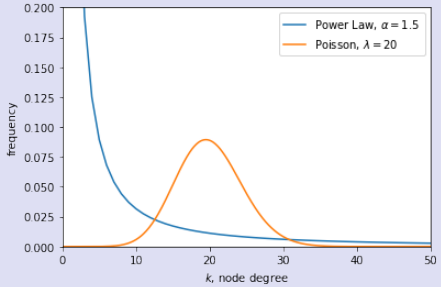
$$\langle \ell \rangle = \frac{1}{n(n-1)} \sum_{i \neq j} \ell_{ij}$$

Degree distribution

The degree distribution is considered an important network property. They can follow two typical distributions:

- **Bell-curved** shaped (Normal/Poisson/Binomial)
- **Scale-free**, also called *Power-law*

A Bell-curved distribution has a *typical scale*: as human height, it is centered on an average value. A Scale-free distribution has no typical scale: as human wealth, its average value is not representative, low values (degrees) are the most frequent, while a few very large values can be found (hubs, large degree nodes). It has a *long tail*, meaning that rare (large) values are not as rare as in a bell-curved distribution.



More details later.

Subgraphs

Subgraph $H(W)$ (induced subgraph): subset of nodes W of a graph $G = (V, E)$ and edges connecting them in G, i.e., subgraph $H(W) = (W, E'), W \subset V, (u, v) \in E' \iff u, v \in W \wedge (u, v) \in E$
Clique: subgraph with $d = 1$
Triangle: clique of size 3
Connected component: a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertex in the supergraph
Strongly Connected component: In directed networks, a subgraph in which any two vertices are connected to each other by paths
Weakly Connected component: In directed networks, a subgraph in which any two vertices are connected to each other by paths if we disregard directions

Triangles counting

δ_u - **Triads of** u**:** number of triangles containing node u
Δ - **Number of triangles in the graph** total number of triangles in the graph, $\Delta = \frac{1}{3}\sum_{u \in V} \delta_u$.

> Each **triangle** in the graph is counted as a **triad** once by each of its nodes.

$\delta_u^{\max}$ - **Triad potential of** u**:** maximum number of triangles that could exist around node u, given its degree: $\delta_u^{\max} = \binom{k_u}{2}$
$\Delta^{\max}$ - **Triangle potential of G:** maximum number of triangles that could exist in the graph, given its degree distribution: $\Delta^{\max} = \frac{1}{3}\sum_{u \in V} \delta^{\max}(u)$

Clustering Coefficents - Triadic closure

The clustering coefficient is a measure of the triadic closure of a network or of a node neighborhood. The triadic closure is a notion coming from social network analysis, often summarized by the aphorism *The friends of my friends are my friends*.

C_u - **Node clustering coefficient:** density of the subgraph induced by the neighborhood of u, $C_u = d(H(N_u))$. Also interpreted as the fraction of all possible triangles in N_u that exist, $\frac{\delta_u}{\delta_u^{\max}}$
$\langle C \rangle$ - **Average clustering coefficient:** Average clustering coefficient of all nodes in the graph, $\bar{C} = \frac{1}{N}\sum_{u \in V} C_u$.

> Be careful when interpreting this value, since all nodes contributes equally, irrespectively of their degree, and that low degree nodes tend to be much more frequent than hubs, and their C value is very sensitive, i.e., for a node u of degree 2, $C_u \in \{0, 1\}$, while nodes of higher degrees tend to have more contrasted scores.

C^g - **Global clustering coefficient:** Fraction of all possible triangles in the graph that do exist, $C^g = \frac{\Delta}{\Delta^{\max}}$

Small World Network

A network is said to have the **small world** property when it has some structural properties[a]. The notion is usually not quantitatively defined, but two properties are required:

- Average distance must be short, i.e., $\langle \ell \rangle \approx \log(N)$
- Clustering coefficient must be high, i.e., much larger than in a random network , e.g., $C^g \gg d$, with d the network density

This property is considered characteristic of *real* networks, as opposed to random networks. It is believed to be associated to particular properties (robustness to failures, efficient information flow, etc.), and to be the consequence of emergent mechanisms typical of *complex systems*.

Be careful: in some contexts, *small world network* can be used for a network that has a small Average distance, without considering its Clustering Coefficient.

---

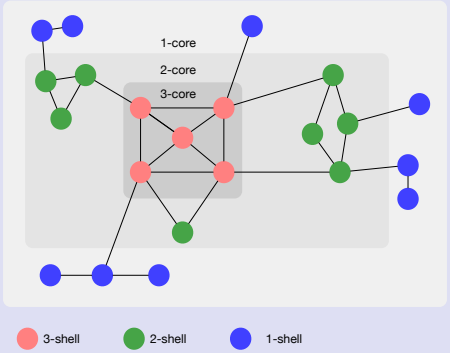
[a] Watts and Strogatz 1998.

Cores and Shells

Many real networks are known to have a **core-periphery** structure, i.e., there is a densely connected core at its center and a more peripheral zone in which nodes are loosely connected between them and to the core.

---

k-core: The k-core (core of order k) of $G(V, E)$ is the largest subgraph $H(C)$ such as all nodes have at least a degree k, i.e., $\forall u \in C, k_u^H \geq k$, with k_u^H the degree of node u in subgraph H.
coreness: A vertex u has coreness k if it belongs to the k-core but not to the $k+1$-core.
c-shell: all vertices whose coreness is exactly c.



Vocabulary

Singleton: node with a degree $k = 0$
Hub: node u with $k_u \gg \langle k \rangle$

Bridge: Edge which, when removed, split a connected component in two.
Self-loop: Edge which connects a node to itself.
Stub: A stub is an half edge, i.e., edge (u, v) has a stub connected to u and another connected to v.

Complete network: $L = L_{max}$
Sparse network: $d \ll 1, L \ll L_{max}$
Connected Graph: Graph composed of a single connected component

Going Further

Books about network science as a whole:

- Barabási et al. 2016 (free)
- Coscia 2021 (free)
- Zinoviev 2018
- Menczer, Fortunato, and Davis 2020

Networks as Matrices

Matrices in short

Matrices are mathematical objects that can be thought as *tables* of numbers. The size of a matrix is expressed as $m \times n$, for a matrix with m rows and n columns. **The order (row/column) is important**.
M_{ij} is a notation representing the element on **row** i and **column** j.

A - Adjacency matrix

The most natural way to represent a graph as a matrix is called the Adjacency matrix A. It is defined as a square matrix, such as the number of rows (and the number of columns) is equal to the number of nodes N in the graph. Nodes of the graph are numbered from 1 to N, and there is an edge between nodes i and j if the corresponding position of the matrix A_{ij} is not 0.

- A value on the diagonal means that the corresponding node has a **self-loop**
- the graph is **undirected**, the matrix is **symmetric**: $A_{ij} = A_{ji}$ for any i, j.
- In an **unweighted** network, and edge is represented by the value 1.
- In a **weighted** network, the value A_{ij} represents the **weight** of the edge (i, j)

Typical operations on A

Some operations on Adjacency matrices have straightforward interpretations and are frequently used, such as **Multiplying** A by **itself** and **Multiplying** A by a **column vector**

Multiplying A by itself

Multiplying A by **itself** allows to know the number of walks of a given length that exist between any pair of nodes: A^2_{ij} corresponds to the number of walks of length 2 from node i to node j, A^3_{ij} to the number of walks of length 3, etc.

Multiplying A by a column vector

Multiplying A by a **column vector** W of length $1 \times N$ can be thought as setting the i th value of the vector to the ith node, and each node *sending* its value to its neighbors (for undirected graphs). The result is a column vector with N elements, the ith element corresponding to the sum of the values of its neighbors in W. This is convenient when working with **random walks** or **diffusion** phenomenon.

Spectral properties of A

Spectral Graph Theory is a whole field in itself, and beyond the scope of this class. A few elements for those with a *linear algebra* background:

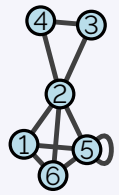
- The adjacency matrix of an undirected simple graph is symmetric, and therefore has a complete set of real eigenvalues and an orthogonal eigenvector basis.
- The set of eigenvalues of a graph is the spectrum of the graph.
- The n eigenvalues are denoted as $\lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \lambda_{\max}$
- The largest eigenvalue $\lambda_{\max}$ lies between the average and maximum degrees.
- In a large, sparse random graph, $\lambda_{\max} \approx \langle k \rangle$
- The number of closed walks of length k in G equals $\sum_{i=0}^{n} \lambda_i^k$
- A graph is bipartite if and only if its spectrum is symmetric (i.e., if λ is an eigenvalue, then so is $-\lambda$
- If G is connected, then the diameter of G is strictly less than its number of distinct eigenvalues

Graph Laplacian

The **Graph Laplacian**, or **Laplacian Matrix** of a graph is a variant of the Adjacency matrix, often used in *Spectral Graph Theory*.
It is defined as $D - A$, with D the *Degree matrix* of the graph, defined as a $N \times N$ matrix with $D_{ii} = k_i$ and zeros everywhere else.

Matrix notation - Example

Graph



A - Adjacency Mat.

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

D - Degree Matrix

$$\begin{pmatrix} 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \end{pmatrix}$$

L - Laplacian

$$\begin{pmatrix} 3 & -1 & 0 & 0 & -1 & -1 \\ -1 & 5 & -1 & -1 & -1 & -1 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 2 & 0 & 0 \\ -1 & -1 & 0 & 0 & 4 & -1 \\ -1 & -1 & 0 & 0 & -1 & 3 \end{pmatrix}$$

A^2

$$\begin{pmatrix} 3 & 2 & 1 & 1 & 3 & 2 \\ 2 & 5 & 1 & 1 & 3 & 2 \\ 1 & 1 & 2 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 1 & 1 \\ 3 & 3 & 1 & 1 & 4 & 3 \\ 2 & 2 & 1 & 1 & 3 & 3 \end{pmatrix}$$

Laplace Operator

Intuitively, the Laplace operator is a generalization of the second derivative, and is defined in discrete situations (discrete Laplace operator), for each element, as the sum of differences between the element and its "neighbors". e.g. In a B&W picture, it's the difference between the greylevel on current pixel and the greylevel of 4 or 8 closest pixels, and it performs *edge detection*. On a graph, with W a column vector representing values on nodes, LW computes for each node the difference to neighbors.

Spectral properties of L

Eigenvalues of the Laplacian have many applications, such as *spectral clustering*, *graph matching*, *embedding*, etc. Assuming G undirected with eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \dots \lambda_n$, here are some interesting properties:

- The smallest eigenvalue λ_i equals 0
- The number of 0 eigenvalues gives the number of connected components

Random Walk matrix

Another useful matrix of a graph is the **Random Walk Transition Matrix** R. It is the column normalized version of the adjacency matrix. R_{ij} can be understood as the probability for a random walker located on node i to move to j.

Going Further

- Introduction to spectral graph theory (Nica 2016)
- Survey on Graph Spectral Theory (Spielman 2012)
- Book on Graph Spectral Theory (Chung and Graham 1997)
- Spectral graph Clustering (Nascimento and De Carvalho 2011)
- Wavelets on graph (Hammond, Vandergheynst, and Gribonval 2011)

Nodes and Edges structural indices, Centrality, node similarity

Node Structural indices

Node structural indices, often called *Node centrality*, reflect how a node is characteristic of a given structural property. This is often summarized as *a measure of the node importance*, however *importance* and *centrality* are subjective/qualitative notions. Thus a centrality, despite its name, do not necessarily measure how *central* a node is, but rather how its position in the graph is typical of the property captured by this index.

Degree Centrality

Degree centrality is the most straighforward centrality. It can be interpreted as a measure of importance, of popularity, e.g., the more friends I have in a social network, the more *important* I am in this network.

Farness - Closeness

The closeness of a node measures how close a node is from all other nodes, in term of shortest paths. To interpret it, we can make a parallel with a circle: the point which has the shortest average distance to all other points of the circle is its center. The node of highest closeness is the equivalent of the center of the circle for this graph. Its formulation is easily understood as the inverse of the farness.

Farness: Average distance to all other nodes in the graph

$$\text{Farness}(u) = \frac{1}{N-1} \sum_{v \in V \setminus u} \ell_{u,v}$$

Closeness: Inverse of the farness, i.e., how close the node is to all other nodes in term of shortest paths.

$$\text{Closeness}(u) = \frac{N-1}{\sum_{v \in V \setminus u} \ell_{u,v}}$$

Harmonic Centrality

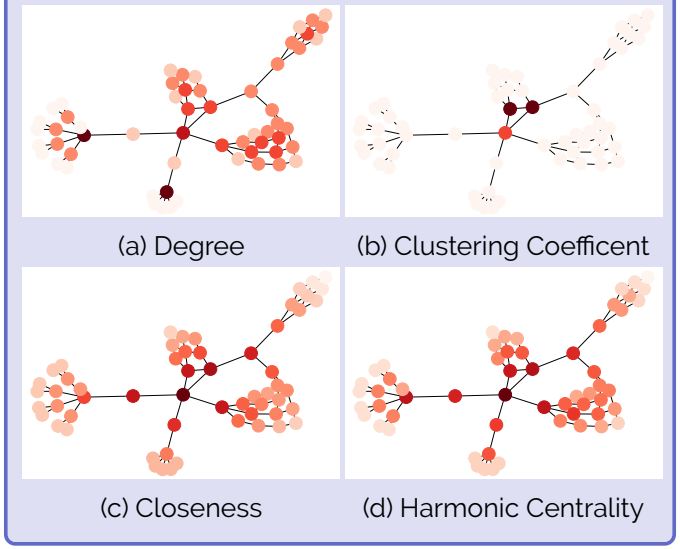
Harmonic centrality: is a variant of the closeness defined as the average of the inverse of distance to all other nodes (Harmonic mean). Well defined on disconnected networks with $\frac{1}{\infty} = 0$. Its interpretation is the same as the closeness.

$$\text{Harmonic}(u) = \frac{1}{N-1} \sum_{v \in V \setminus u} \frac{1}{\ell_{u,v}}$$

Clustering Coefficient

This score (previously defined), measures the *triadic closure* of a node. A high score is often interpreted as being well embedded in a particular community (friends of my friends are my friends because we all belong to the same group), a low score can be typical of a *bridge* node, e.g., few connections between my friends because they belong to different social circles.

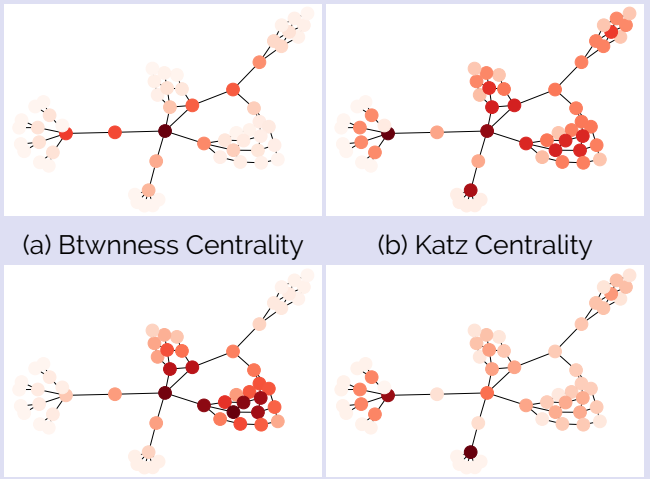
Centrality - Examples - 1



(a) Degree (b) Clustering Coefficent

(c) Closeness (d) Harmonic Centrality

Centrality - Examples - 2



(a) Btwnness Centrality (b) Katz Centrality

(c) Eigenvector Centrality (d) PageRank

Betweenness centrality

The betweenness centrality measures how much the node plays the role of a bridge. The highest the betweenness, the more the node is essential to move quickly in the graph. More formally, the betweenness of u is defined as the fraction of the shortest paths between all pairs of nodes in the graph (but u) that go through u. As a consequence, if we remove a node of high betweenness, many shortest paths will become longer, and the graph harder to navigate. The extreme situation is a node on the only path between otherwise disconnected components: if we remove this node, some nodes becomes unreachable from others. Those nodes thus tend to have high betweenness. It is defined as:

$$C_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

with σ_{st} the number of shortest paths between nodes s and t and $\sigma_{st}(v)$ the number of those paths passing through v.
The betweenness tends to grow with the network size. A normalized version can be obtained by dividing by the number of pairs of nodes, i.e., for a directed graph: $C_B^{\text{norm}}(v) = \frac{C_B(v)}{(N-1)(N-2)}$.

Eigenvector centrality

Eigenvector centrality is a recursive definition of importance: a node is important if it is connected to other important nodes. In practice, it is defined in the following way: the eigenvector centrality C_u for every node u of the graph is such that if each node *sends* its centrality score to its neighbors, then the sum of scores received by each node will be equal to λC_u (with λ a normalization constant). More formally,

$$C_u^{t+1} = \frac{1}{\lambda} \sum_{v \in N_u^{in}} C_v^t \quad (1)$$

This recursive definition can be interpreted in term of eigenvectors and eigenvalues, which is defined as $Ax = \lambda x$, with x an eigenvector, λ the corresponding eigeinvalue. The eigenvector centrality is defined as the leading invector, i.e., the eigenvector associated with the highest eigenvalue, the only solution for which all centrality values are positive.
A simple way to compute this eigenvalue is called the power method: one starts with random values on nodes, and iterate equation 1. After some time, it can be proven that the values converge to the eigenvector centrality.

Eigenvector centrality cannot in general be computed on directed networks, because of source nodes, i.e., $k^{in} = 0$. Those nodes have by definition an eigenvector centrality of 0 at t+1, and thus *send* a value of 0 at t+2, which might in turn result in a score of 0 for its successors, and so on and so forth.

Katz centrality

Katz centrality[a] is said to be a measure of the influence potential of a node. For a node u, it is defined as the sum, for all path length distance ℓ, of the number of nodes located at distance exactly ℓ of u, discounted of a factor decreasing as ℓ increases. The intuition is that, the more nodes can be accessed in few steps, the higher the value. More formally, it is expressed as

$$C_{\text{Katz}}(u) = \sum_{\ell=1}^{\infty} \sum_{v=1}^{N} \alpha^\ell (A^\ell)_{vu}$$

in which $(A^\ell)_{vu}$ means the number of paths of length ℓ from v to u, and $0 < \alpha < \frac{1}{\lambda_{\max}}$ an attenuation parameter smaller than the reciprocal of the largest eigenvalue of A, to allow computation in matrix form.

Note that in a directed network, Katz centrality as presented here represents a *vote* mechanism: a higher centrality of u means that more nodes can reach u quickly, and not that u can reach many nodes quickly. We can inverse it by using $(A^\ell)_{uv}$ instead of $(A^\ell)_{vu}$

---

[a] Katz 1953.

Katz centrality and Eigenvector centrality

Katz centrality can also be understood[a] as a generalization of the eigenvector centrality, with a recursive definition, such as:

$$C_{\text{Katz}}(u) = \alpha \sum_{v \in N_u^{in}} C_{\text{Katz}}(v) + \beta$$

with $\beta = 1$. We can see that with $\beta = 0$ and $\alpha = 1/\lambda_{\max}$ Katz centrality is equivalent to the eigenvector centrality.

---

[a] M. Newman 2018.

Pagerank centrality

Pagerank centrality is famous for being the method originally used by google to rank web-pages: all pages containing the researched words are ordered according to their Pagerank score in the graph of the WWW, in which nodes are webpages and edges are hyperlinks.
It is a variant of the Eigenvector centrality, solving the problem of source nodes.
Pagerank introduces two improvements: 1) at each step t, each node gain a small constant value. 2) The values *sent* are divided equally among successors (normalization by degree). Equation 1 thus becomes:

$$C_u^{t+1} = \alpha \sum_{v \in N_u^{in}} \frac{C_v^t}{k_v^{out}} + \beta \quad (2)$$

with, by convention, $\beta = 1$,$\alpha \in [0, 1]$ a parameter.
Pagerank centrality can also be expressed as the leading eigenvector of the so-called *Google matrix* G, defined as $G_{ij} = \alpha S_{ij} + (1 - \alpha)/n$, with S_{ij} the adjacency matrix normalized by column.

Pagerank & Random Walk

Pagerank can be interpreted in term of **random walks**. If you consider a random walker moving from nodes to nodes following randomly chosen out-going links, which starts on a random node and moves an infinite number of times. Consider that at each step, this random walker can *teleport* to any other node with a probability α instead of following an outgoing edge. Then, the probability for this random walker to be on each particular node corresponds to its Pagerank score.
We can note that the average length of a walk before restart is $\frac{\alpha}{1-\alpha}$. The *typical* value $\alpha = 0.85$ thus means that random walkers move in average 5.7 times before restart, a typical value of average distance in real graphs.

Edge Structural indices

Edges structural positions in the network can also be described using structural properties, most of them being similar to node centralities.

Edge Clustering

Edge Clustering C^e of an edge (u, v) is the fraction of the neighbors of at least one of the two nodes which are neighbors of both of them, i.e.,

$$C^e(u, v) = \frac{|N_u \cap N_v|}{|N_u \cup N_v| - 2}$$

High clustering edges are said *Integrative*, low values edges are said *Dispersive*.

Edge Betweenness

Edge betweenness Is defined exactly as node betweenness, but counting shortest paths going through each edge instead of each node, i.e.,

$$C_B(u, v) = \sum_{s \neq t \in V} \frac{\sigma_{st}(u, v)}{\sigma_{st}}$$

with σ_{st} the number of shortest paths between nodes s and t and $\sigma_{st}(u, v)$ the number of those paths passing through edge (u, v).

Node Similarity

When studying a network, one might be interested in comparing nodes between themselves, for instance to discover the most similar nodes in the network, or to assess if two nodes they are interested in share a similar network location.

A first approach is to define the similarity between nodes u and v, as their number of common neighbors, $\sigma_{u,v}$ as: $\sigma_{u,v} = |N_u \cap N_v|$.

A weakness of this approach is that high degree nodes tend to be considered similar to low degree nodes. A variant consists in normalizing by nodes degrees, thus computing the Jaccard Coefficient of neighborhoods:

$$\sigma_{u,v} = \frac{|N_u \cap N_v|}{|N_u \cup N_v| - 2}$$

Cosine Similarity

Cosine similarity $\sigma^{\cos}$ is a standard method to compare *vectors*. It is defined for two vectors x, y as :

$$\sigma_{xy}^{\cos} = \frac{x.y}{|x||y|}$$

This score can be used to measure the similarity between nodes neighborhoods by using as vector x_u of node u the row of the adjacency matrix corresponding to this node, i.e., $x_u = A_u$.
Cosine similarity of nodes then simplifies to:

$$\sigma_{uv}^{\cos} = \frac{|N_u \cap N_v|}{\sqrt{k_u k_v}}$$

Pearson coefficient

Pearson coefficient is a standard measure of correlation between variables X and Y, which is defined as :

$$r_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

with cov the covariance and σ the standard deviation.
Much as for Cosine Similarity, we can adapt this measure to nodes similarities by considering A's rows as discrete variables. The result can be understood intuitively by observing that the numerator becomes:

$$cov(u,v) = |N_u \cap N_v| - \frac{k_u k_v}{N}$$

which can be interpreted as the **number of common neighbors minus the expected number of common neighbors** in a randomized network, given nodes degrees.

$cov(u,v) = 0$ means that the number of common neighbors is exactly what we would expect by chance given their degrees, while positive values means that they have more than expected (resp. for negative values).

Going further

Books on network science (see first CheatSheet) always introduce centralities.
Survey on the topic:

- Rodrigues 2019

Random Graphs

Many elements of this course are inspired by the excellent classes by Aaron Clauset, than can be found online:
`http://tuvalu.santafe.edu/~aaronc/courses/5352/`

Synthetic networks usages

Using synthetic networks is essential in network science for several reasons. In particular, they allow to:

- Study some properties in a **controlled environment**. *What happens if we increase property X, while keeping all other properties constant?*
- Compare an observed network with a **randomized** version of it. *I observed property X in my data, is it something remarkable, or would I observe the same thing on a random network similar to my graph?*
- Explain a phenomenon. *Property X seems exceptional. It can be reproduced in random networks by simple mechanism Y.*
- Generate synthetic datasets, for instance to test the same algorithm on multiples variations of the same network.

Synthetic networks types

There are three main types of synthetic networks:

- **Deterministic models** are instances of famous graphs or, more commonly, repeated regular patters. e.g., *Caveman graph, grids, lattices*.
- **Generative models** assign to each pair of nodes a probability of having an edge according to their properties (degree, label, etc.). e.g., *Erdős Rényi, Configuration model, etc.*
- **Mechanistic models** create networks by following a set of rules, a process defined by an algorithm. e.g., *Preferential attachment, Forest fire, etc.*

Regular lattices

Regular lattices are defined as repetition of the same pattern a given (potentially infinite) number of times. Nodes all have the same degree. The pattern can be in 1, 2 or more dimensions.
The **clustering coefficient** depends on the structure, it can be large if the structure is made of triangles, for instance. It is the same for all nodes (except potentially nodes at the boundaries).
The **average distance** grows quickly with n, if $k \ll n$

Erdős-Rényi (ER) model

The **Erdős-Rényi (ER)** model is the simplest random graph model. Assuming that we know the number of nodes and the number of edges, and no other information, then edges are simply put between randomly chosen pairs of nodes.
ER models can be defined in two ways:

- in the $G(n, L)$ formulation, the number of edges of the generated graph is set to exactly L, and thus L random pairs of nodes are chosen among the set of all existing node pairs (sharp constraint, microcanonical ensemble).
- in the $G(n, p)$ formulation, an edge is added between any set of node with a probability p. (soft constraint, canonical ensemble).

Properties of both model are similar when the number of edges (defined by L or p) is large.

Random version of observed graph

When one wants to compare a real network with a **randomized** version of it (also called a **rewired** network), the usual way is not to start from the original network and to actually rewire it edge by edge, but instead to generate a new ER random graph keeping the same number of nodes and the same number of edges (or the same density) as the observed network. Properties of the observed network can then be compared with the generated network. Note that it does note make sense to compare the properties of any particular node in both networks, since nodes in the random graph have no identity. For many applications, there is not need to actually generate a random graph: one can simply compare properties of the real network with theoretical properties of the random graph.

Soft ER

In the soft ER, the number of edges is not known in advance. The distribution of the number of edges in the soft ER is described by the **binomial distribution** $\mathbb{B}(L^{max}, p)$
From the known properties of the Binomial distribution, it can be shown that:

- The **expected number of edges** is $\langle L \rangle = pL^{max}$,
- The **variance of the number of edges** is $\sigma^2 = L^{max}p(1-p)$

Binomial distribution

The **Binomial distribution** $\mathbb{B}(N_b, p_b)$ is a discrete distribution modeling the number of successes x in a sequence of N_b independent experiments with success probability p_b. For instance, it models how many times (x) one will obtain a 6 (*success*) if they throw a dice N_b times and that the probability to obtain a 6 is $\frac{1}{6}$. It is defined as $P(x) = \binom{N_b}{x} p^x (1-p_b)^{N-x}$. $\binom{N}{x}$ is the binomial coefficient, describing the number of ways, disregarding order, that x elements can be chosen among N_b.

ER: Degree distribution

Since each node has an independent probability to be connected with each other node, the degree distribution of the ER model is modeled as a binomial distribution $\mathbb{B}(N-1, p)$, i.e., the probability to have a given degree knowing that we have a probability p to have a link with each of the other nodes in the graph. From the properties of the Binomial distribution, we know that:

- The **expected average degree** is $\langle k \rangle = p(N-1)$
- The **variance of the degree** is $\sigma_k^2 = p(N-1)(1-p)$

We can note that the distribution becomes increasingly narrow as the network size increases, i.e., we are increasingly confident that the degree of a node is in the vicinity of $\langle k \rangle$:

$$\frac{\sigma_k}{\langle k \rangle} = \frac{1}{(N-1)^{1/2}}$$

ER: Approximation of degree distribution by a Poisson Distribution

When the number of nodes N is large and the average degree $\langle k \rangle$ is small, the degree distribution can be approximated by a Poisson distribution $\mathrm{Pois}(\langle k \rangle)$. From the properties of Poisson distributions, we approximate that for a network with average degree $\langle k \rangle$:

- The **variance of the degree** is $\sigma_k^2 \approx \langle k \rangle$

Poisson distribution

The **Poisson distribution** $\mathrm{Pois}(\delta)$ is a discrete distribution modeling the probability of observing exactly x occurrences of an event in a period of duration Δ_t if this event occurs randomly and that there are in average δ occurrences of it during a period Δ_t. Working with the Poisson distribution is convenient because it depends only on a single parameter δ.

ER: Clustering Coefficient

The **Global Clustering Coefficient** of a network is defined as the fraction of closed triads among all triads. Since any edge (u, v) has a fix probability to exist p independently of the existence of any other edge in the network, the probability of having edge $(a, c) \in E$ for a triad $[a, b, c]$ such as $(a, b), (b, c) \in E$ tends towards p for large graphs.
Thus, the clustering coefficient of an ER graph is $C^g \approx p$. Since we know that most real networks are sparse, p is small, thus C^g is small. A similar reasoning can be used to show that the average clustering coefficient $\langle C \rangle$ is small too.

ER: Average Distance

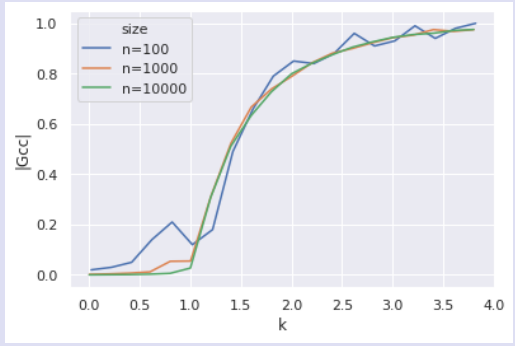
We can intuitively estimate the order of the **Average Distance** of an ER random graph as follows:
We know that the clustering coeffient of an ER graph is small. Therefore, we can approximate the graph as having a tree-like structure. As a consequence, the number of nodes located at distance d of a node u increases as $\langle k \rangle^d$. From this approximation, the relation between distance and number of nodes is $N = \langle k \rangle^d$ hops, thus the order of ℓ is $\log_{\langle k \rangle} n = \frac{\log N}{\log \langle k \rangle}$.
We can thus say that the order of the average distance of a sparse ER graph relatively to its size is $\mathcal{O}(logN)$, and thus that: **ER graphs have a short average distance**.

Order of magnitude

The notation $\mathcal{O}$ is used to represent the **order of magnitude** of a value. It roughly indicates how this value is related to another one, ignoring any constant. For instance, $\mathcal{O}(x) = \mathcal{O}(10x) = \mathcal{O}(x/10)$. Typical orders of magnitude are $\mathcal{O}(\log x)$, $\mathcal{O}(x)$, $\mathcal{O}(x^2)$ and $\mathcal{O}(2^x)$.

ER: Largest connected component

The largest connected component of a graph is a way to measure its connectivity. On random networks, the relation between the density (or average degree) of a graph and the size of its largest connected component is known to undergo a *phase transition* phenomenon, i.e., a rapid change when a threshold is crossed. More precisely, as long as $\langle k \rangle < 1$, several connected components of similar sizes exist in the network, while, when $\langle k \rangle > 1$, the graph has a single *giant component* with high probability.



An intuitive way to understand this phenomenon is to use the same observation of the graph being tree-like as previously. Since the number of nodes N that can be reached after d hops can be estimated to grow as $\langle k \rangle^d$, a value of $\langle k \rangle < 1$ leads to an impossibility to reach all nodes even for a large d, while $\langle k \rangle > 1$ leads to arbitrarily large N for long enough d. Proper demonstration and more details can be found in the original paper[a].

You can explore this property using this interactive *explorable*: `https://www.complexity-explorables.org/explorables/the-blob/`

[a] Erdős and Rényi 1960.

Configuration Model (CM)

The **Configuration Model** is another classic random graph model in which the degree of each node –or the degree distribution– is preserved. In general terms, a configuration model is defined by the number of nodes in the graph, the number (or probability) of edges, and a distribution of degrees of nodes.
This degree distribution can either be chosen *a priori*, for instance following a *Poisson* or a *Power-law* distribution, or by taking the observed distribution of a real network we would like to obtain a randomized-version of.
Note that in the later case, nodes can be considered to retain their identity: one can compare the local properties of the node of highest degree between the two graphs, for instance.

Why the configuration model

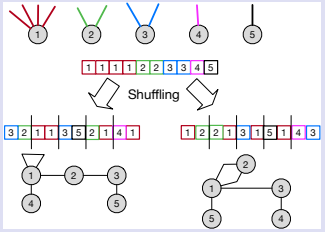
For many real graphs, nodes represent real entities, and the degree of those nodes is due to an intrinsic property of those nodes, which is known in advance and should be taken into account. For instance, let's consider a network representing flight connections between airports: each node represents an airport, and there is an edge between two airports if a direct flight exist between them. *JFK* international airport in New-York will likely be a Hub in this network, having a very large degree. This large degree is a consequence of the properties of the city it belongs to: large population, touristic attraction, etc. So, *if connections between airports were random*, it could nevertheless be relevant to keep the degree of this node.
Furthermore, the degree distribution itself is also a characteristic of the network: the fact that hubs *do exist* in the network change its properties, compared with a network in which such nodes do not exist.

Approximate/Soft Configuration model

In the approximate version of the Configuration model, each pair of node is connected by an edge with a given probability, which depends on their **objective degrees**.
More precisely, the probability of having an edge (i, j) is defined as $p_{uv} = \frac{k_u k_v}{2L}$. Note that this is a well defined probability only if $\max(k_u)^2 < 2m$, otherwise it can be higher than 1. p_{uv} should therefore rather be understood as the *expected number of edges* in a multigraph.
Intuitively, this definition can be understood as follows: each node u has k_u stubs. The total number of stubs in the graph is $2L$. Knowing that node v has k_v stubs, the probability for each stub of u to connect to a stub of v is $\frac{k_v}{2L}$.
Note that this model is defined such as self-loops can exist.

Rewired exact configuration model

When the objective of a configuration model is to obtain a randomized version of an observed graph, a common approach is to fix the exact degree of each node, and to connect *stubs* randomly. An efficient way to do so is to use the following algorithm:
1) Create a list s such as it contains k_u times the index of node u - 2) Randomize s - 3)For each i in $[0, L]$, create an edge between nodes of index s_{2i} and s_{2i+1}.



Note that this method can create self-loops and multiple links between the same nodes, even if the original network was a *simple graph*. However, the number of multiple links and self-links decreases when the number of nodes increases, for sparse graphs. The probability of an edge to exist between two nodes depends on their degree, and is the same as in the soft CM.
For more details on configuration models with fixed degree sequences, see[a].

[a] Fosdick et al. 2018.

CM: Clustering Coefficient

The clustering coefficient of the configuration model can also be studied theoretically. Its derivation is beyond the scope of this class and can be found in the literature[a]. Intuitively, we can use the same reasoning as for the ER model: the probability of having edge $(a, c) \in E$ for a triad $[a, b, c]$ such as $(a, b), (b, c) \in E$ is $\frac{k_a k_c}{2L}$. However, the probability of observing (a, b) and (b, c) and thus to have such a triad also depends on k_a, k_b, k_c. In the end, the clustering coefficient is

$$C = \frac{1}{L} \frac{[\langle k^2 \rangle - \langle k \rangle]^2}{\langle k \rangle^3}$$

where $\langle k^2 \rangle$ correspond to the expected *variance* (second moment) of the degree.
Since the right part of the equation is a constant depending only on the average degree, the order of the clustering coefficient is $\mathcal{O}(1/L)$, and thus small for large graphs. This is true as long as $\langle k^2 \rangle$ is definite, which might not be the case if the degree distribution is a *power law*.

[a] M. Newman 2018.

CM: Friendship paradox

An interesting property of the Configuration Model with heterogeneous degree distribution arises when we study the **average degree of random neighbors**. Let's call p_k the probability to pick a node of degree k when we pick a node at random. This probability represents the *degree distribution* chosen for the configuration model. Now, if we choose one node at random, and then pick one of its neighbors at random, what is $p_{neighb,k}$, the degree distribution of *random neighbors*? It is different, because nodes with a higher degree have, by definition, a higher probability of being chosen. More formally,

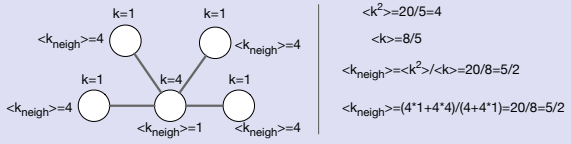
$$p_{neighb,k} = \frac{k}{2m} n p_k = \frac{k p_k}{\langle k \rangle}$$

because np_k is the number of nodes of degree k in the graph, and $\frac{k}{2m}$ is the probability to pick at random a *stub* of a particular node of degree k among all stubs.

We can now compute the **average degree of neighbors** of a node chosen at random, as:

$$\langle k_{neighb} \rangle = \sum_k k p_{neighb,k} = \frac{\langle k^2 \rangle}{\langle k \rangle}$$

Thus if all degrees are the same (*homogeneous*), $\langle k_{neighb} \rangle = \langle k \rangle$, but if it is **heterogeneous**, $\langle k_{neighb} \rangle > \langle k \rangle$ due to the comparatively larger influence of high degrees.



CM: Average distance

We use the same logic as for the ER model of the graph being locally tree-like due to the low Clustering Coefficient to show intuitively that the *average distance* is short. This property is verified experimentally.

E

xamples of differences in Clustering and average path length for a few real graphs, compared with randomized versions of it.

graph	N	L	k	C_g	$\langle \ell \rangle$	ER-C_g	ER-$\langle \ell \rangle$	CM-C_g	CM-$\langle \ell \rangle$
karate	34	77	4.53	**0.26**	2.42	*0.14*	2.42	*0.14*	2.55
football	115	613	10.66	**0.41**	2.51	*0.10*	2.25	*0.07*	2.28
wiki-science	687	6523	18.99	**0.47**	3.43	*0.03*	2.55	*0.08*	2.65
euroroad	1174	1417	2.41	0.03	**18.40**	0.00	7.66	0.00	9.55

Differences btw. Real & Random networks

When comparing real networks to ER and CM networks of similar properties, we observe that they tend to disagree on one of two key properties: on real graphs, usually, the graph has a high clustering coefficient and a short average distance (or sometimes the opposite).
On the contrary, random networks have both a low clustering coefficient and a short average distance.

Watts-Strogatz (WS) Model

The Watts-Strogatz model was introduced[a] to show how a simple phenomenon could create networks having both a large clustering coefficient and a short average distance.
The model has 3 parameters:

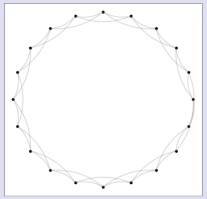
- N: number of nodes
- K: initial number of neighbors
- p: rewiring probability

The network is created following a 2-step processes: first N nodes are disposed on a ring, and each node is connected to its K closest neighbors. Then each edge is replaced by a random edge with probability p. It can be interpreted as a network combining the properties of a (1-dimentional) **regular lattice** and of an **ER network**.

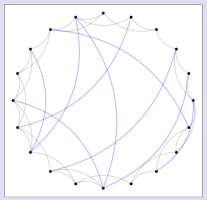
[a] Watts and Strogatz 1998.

WS - Illustration

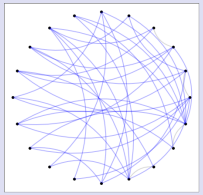
From left to right: WS graphs when increasing the probability of rewiring. $N = 20, K = 4$



(a) $p = 0$
Regular



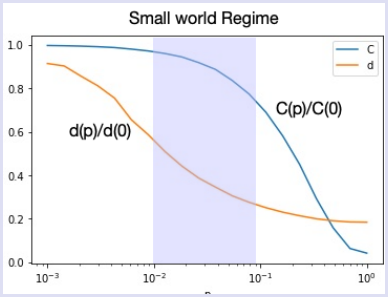
(b) $p = 0.3$
Small world



(c) $p = 1$
Random

WS - Small World Regime

If p is small, the network has properties similar to a regular lattice, and if p is large, properties of an ER graph.
We can observe this transition by comparing how the Clustering (C) and average distance (d) change when varying p, compared with the network when $p = 0$, i.e., a regular lattice.



Example with $N = 200$, $K = 6$.

WS - Clustering

Properties of the WS model are not as simple to study theoretically as previous random graphs, so most details are not presented here. It can be shown however, that the global clustering coefficient can be approximated by:

$$C^g = \frac{3(K-2)}{4(K-1) + 8Kp + 4Kp^2}$$

which is independent of N, thus can be large even for large graphs.

WS - Average Path length

The average path length of the WS model has been studied through approximations and numerical simulations[a] and can be shown to become small quickly with the increase in p.

[a] M. E. Newman 2000.

WS - Degree distribution

Without entering into details, it can be shown the the degree distribution range from a fixed degree for all nodes to a Poisson distribution, since each rewired edge is decreasing the degree of some nodes and increasing the degree of some others in a random way.

Barabási-Albert (BA) Model

The **Barabási-Albert** model of random graphs was introduced[a] to illustrate how a simple mechanism could explain a common property of real graphs, the **power-law degree distribution**. This mechanism is though to somewhat mimic what is happening in real life, at least for some networks. It is often called **preferential attachment**, and mimic the **rich get richer phenomena**: nodes that already have a large degree are more *attractive*, and thus are more likely to become connected with other nodes creating links.

[a] Barabási and Albert 1999.

BA - Preferential attachment

The preferential attachment process has two parameters, the number of edges to create at each step m and the initial number of nodes m_0, with $m \leq m_0$. It is defined by the following iterative process:

- Start with a connected graph with m_0 nodes
- At each step, add a new node and m links connecting it to m other nodes chosen randomly proportionnaly to their degree, i.e., with probability $p_i = \frac{k_i}{\sum_j k_j}$

BA - Degree distribution

The degree distribution created by the preferential attachment mechanism is a power law of exponent $\alpha = 3$. The exponent of the distribution does not depend on parameters m and m_0. The degree exponent is *stationary in time*, i.e., it stays the same while we add new nodes and edges.

Nodes degree increase with time: the earlier a node was added, the larger its degree tends to be.

BA - Average Path Length

Networks generated by the BA process have a power-law degree distribution of exponent $\alpha = 3$. It is known that such networks have a short average path length, more formally:

$$\langle \ell \rangle = \frac{\ln N}{\ln \ln N}$$

BA - Clustering Coefficient

Although the demonstration is beyond the scope of this class[a], it can be shown that the clustering coefficient of BA graphs is:

$$C = \frac{L}{4}\frac{(\ln N)^2}{N}$$

This is more than for a random network, but still decreases with the network size, and tends toward 0 for large graphs. It is thus considered a **small** clustering coefficient.

[a] Barabási and Albert 1999.

Other random graph models

Many other graph models have been proposed in the literature, either *mechanistic models* to mimic common properties of some graphs, as with BA and WS models, or *statistical models* to generate random graphs with imposed constraints, as the Configuration model does with degree distributions.
Some examples of mechanistic models:

- Vertex copying model (J. M. Kleinberg et al. 1999)
- Tunable-clustering scale-free model (Holme and Kim 2002)
- Forest fire model (Leskovec, J. Kleinberg, and C. Faloutsos 2005)

Some examples of statistical models:

- Exponential Random Graphs (Robins et al. 2007)
- Stochastic Block Models (Peixoto 2019)
- A survey on the topic (Orbanz and Roy 2014)

Scale-Free Networks

Scale-Free: Definition

A network is said to be **Scale-Free** when its degree distribution follows a Power-Law distribution, or can be approximated by a Power-Law distribution.

Power-Law (PL)- Approximate Distribution

A **Power-Law distribution** is defined as follows:

$$P(k) \sim k^{-\alpha} = \frac{1}{k^{\alpha}}$$

α is called the *exponent* of the distribution.
Intuitively, the more α is large, the more large values are rare. For instance, with $\alpha = 0$, it corresponds to a uniform distribution (any degree is equivalently probable). With $\alpha = 1$, the probability of a node taken at random to have degree k is $\sim \frac{1}{k}$. Usually, a distribution is considered scale free when $2 \leq \alpha \leq 3$, as we will see.

PL - Boundaries

In most settings, the Power-Law degree distribution exists only for a certain range of degrees.
This makes sense in real networks: few people have 0 or 1 social contacts, for instance, few websites have no incoming nor outgoing hypertext links, or we wouldn't even be aware of them. Thus there is a **lower bound** k_{min} from which the distribution exists.
Similarly, real networks represent entities of the real world, which are in finite numbers, therefore the number of elements itself is a limit. But in many situations, even lower thresholds exist: Social networks often impose a limit to the number of connections to avoid spammers, time and space also typically impose limits to what is possible or not in a network. An **upper bound** k_{max} can be used to limit the distribution.

Power-Law - Exact Distribution

For a distribution to be properly defined, the sum of all probability must be equal to one, we therefore add a normalization constant C to ensure this property.

$$\int P(k) = 1 = \int Ck^{-\alpha} = C \int k^{-\alpha}$$

From which we can define C:

$$C = \frac{1}{\int_{k_{\min}}^{\infty} k^{-\alpha} dk} = (\alpha - 1) k_{\min}^{\alpha - 1}$$

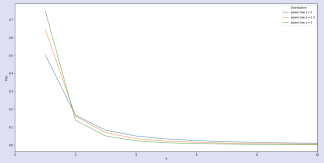
And finally, the exact definition of the Power-Law degree distribution with lower bound:

$$P(k) = (\alpha - 1) k_{\min}^{\alpha - 1} k^{-\alpha} P(k) = \frac{\alpha - 1}{k_{\min}} \left(\frac{k}{k_{\min}} \right)^{-\alpha}$$

Power-Law - Plotting

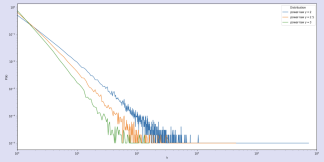
A famous property of the Power-Law distribution is that it looks like a line when plotted in a log-log plot, i.e., a plot in which the x-axis (degrees) and the y-axis (frequency of degrees) are represented using a logarithmic scale.

Power-Law distributions in linear scale for degrees [1-10] (100 000 samples)



Power-Law distributions in linear scale for degrees [1-100000] (100 000 samples). The distribution is so heterogeneous that is is not readable.

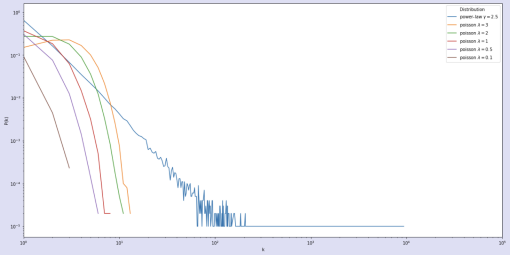
Power-Law distributions in log-log scale for degrees [1-100000] (100 000 samples).



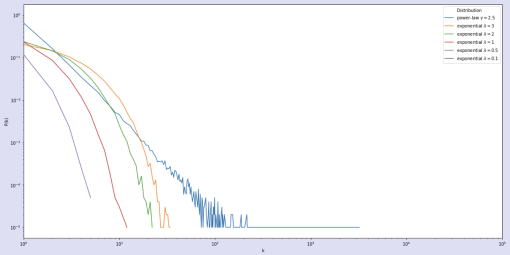
Power-Law - Long tail

Compared with other well-known distributions such as Poisson or exponential distribution, a key difference is what is called the **long-tail** property: very large values are rare, but possible. We can observe this long tail by comparing with other distributions on log-log plots.

Comparing power-law with Poisson distributions



Comparing power-law with Exponential distributions



SF networks - universality

Scale-Free networks are widely studied because they are considered to be very frequent in the real world. Some important papers discovered the existence of Power-Law degree distribution in a variety of large real networks, notably:

1. The World Wide Web (webpages) (Barabási and Albert 1999)
2. The internet (physical network) (M. Faloutsos, P. Faloutsos, and C. Faloutsos 1999)
3. Airline connections (Guimera and Amaral 2004)
4. Scientific collaborations (M. E. Newman 2001)
5. Romantic interactions (Liljeros et al. 2001)

It must be noted, however, that many real world networks **are not scale-free**. A typical counter-example is a road-network, in which nodes correspond to intersection and edges to roads: for practical reasons, intersections with large degrees do not make sense.

Why is it called *scale free*

Because they have no (typical) scale!
It is defined in opposition to Poisson and other Bell-Shaped distributions, which are **centered around their average value**. Let's take a typical example: The height of humans follow a Bell-shaped distribution: the average height is 1.65m, and most humans are quite close to this value, there is a **typical scale** of human height. On the contrary, human wealth distribution follows approximately a power-law [a]: a few humans are extremely wealthy (Billions of $), while more than half the world population posses less than 10 000$. As a consequence, the average human wealth (70 000$) is not at all representative of human wealth.

[a] `https://en.wikipedia.org/wiki/Distribution_of_wealth`

Central moments

The first two central moments of a distribution are the mean $\langle k^1 \rangle$ and the variance $\langle k^2 \rangle$. They are defined as

$$\langle k^m \rangle = \int_{k_{\min}}^{\infty} k^m p(k) dk = (\alpha - 1) k_{\min}^{\alpha - 1} \int_{k_{\min}}^{\infty} k^{-\alpha + m} dk$$

From this, we can conclude that **central moments are defined only if** $\alpha > m + 1$, otherwise they diverge towards infinity, they are not properly defined.
Thus:

1. $\langle k^2 \rangle = \frac{\alpha - 1}{\alpha - 2} k_{\min}$, **if and only if** $\alpha \geq 2$
2. $\langle k^3 \rangle = \frac{\alpha - 1}{\alpha - 3} k_{\min}^2$, **if and only if** $\alpha \geq 3$

Divergence in practice

In practice, one can always compute the mean and variance of a provided, observed degree distribution. So what does it mean that they *diverge*?
The problem arises when we are not certain to observe the whole network. Usually, a large sample of a population has the same mean and variation than the whole population, and the largest the sample, the more precise the value.
But in a power law, moments are **dominated** by the largest values in the long tail: some rare values are *so large* that they shift the moments. So the more data we observe, the higher the moments.

Divergence: consequences

The consequence of diverging moments is that **if the distribution follows a power law**, then if the exponent is below 2, you should now rely on the mean degree or the variance. If the exponent is between 2 and 3, you can (relatively) rely on the mean, but not on the variance. Be careful though, even if $\alpha > 2$, the mean converges slowly, i.e., you need a very large sample for your mean to be close to the real value.

Fitting power laws

When confronted with a power law degree distribution, we might want to **fit** the distribution, i.e., to **find the exponent** of the distribution. A naive and simple way to do it is to plot the distribution on a log-log plot and to find the slope of the line, either graphically or through least-square regression on the log-transformed values of degrees and frequencies.
This however suffers from a strong bias: values in the tail are based on a few samples, and introduce noise.
The most appropriate method is to use Maximum Likelihood Estimation (MLE[a]), taking into account min and max-boundaries, as described in[b]

[a] https://towardsdatascience.com/a-gentle-introduction-to-maximum-likelihood-estimation-9fbff27ea12f

[b] Goldstein, Morris, and Yen 2004.

Exponent limits

In real networks, we consider that we should have $\alpha \geq 2$, because a lower exponent would mean that the distribution is so skewed that we expect to find nodes with a degree larger than the size of the network.
Furthermore, if the exponent is too large, large degree nodes becomes so rare, that the network would need to be enormous to observe such a node. For instance, with $\alpha = 5$, we need to observe $N = 10^{12}$ nodes to expect to observe a single node of size 1000.

Exponent and shortest-paths

Random networks with Poisson degree distribution already have a *short average distance*. However, it is possible to define classes of networks with even smaller average distance based on the exponent α:

- $\alpha = 2$: The biggest hub degree is of order $\mathcal{O}(N)$, thus most nodes are at distance 2. The average path length can be considered a small constant, independent of N
- $2 < \alpha < 3$: **Ultra Small World**: $\langle \ell \rangle = \frac{\log \log N}{\log(\alpha - 1)}$
- $\alpha = 3$: $\langle \ell \rangle = \frac{\log N}{\log \log N}$
- $\alpha > 3$: $\langle \ell \rangle = \log N$, the network behaves approximately like an ER network.

Scale-free network controversy

There is an on-going debate in the network science community over the prevalence of scale-free networks. For some authors[a], most real networks follow to some extent a power-law degree distribution, while, for some others, scale-free networks are rare[b]. The controversy has been studied[c] and can be interpreted as differences between scientific approaches: one popular among (some) physicists (*scale-freeness is the sign of a universal law*) and another one common among statisticians *(scale-freeness is an empirical characterization)*.

[a] Barabási and Bonabeau 2003.
[b] Broido and Clauset 2019.
[c] Jacomy 2020.

SF networks: what to do

Is your network Scale-Free? The first question you might ask yourself is: *why do you need to know?*.

- If the goal is to characterize a network, then plotting the degree distribution might be more useful than fitting a power-law exponent to it
- If the goal is to show that the distribution is broad, significantly different from a bell-shaped distribution, then plotting the distribution might be enough
- If the goal is to show that the distribution is *approximately* a power-law, for instance because an algorithm complexity or a proof can be made for such cases, then fitting a line on a log-log plot and talking about *power-law-ish* might be enough
- If on the contrary it is scientifically important to argue that the network is, without doubts, a scale-free networks, then you need to be fully aware of the controversy and to position your work relatively to it.

Community Structure

Blocks and Communities: Definition

The general idea of blocks and communities is that nodes of a network can be grouped together in homogeneous sets, based on the network topology. The problem of automatically discovering those groups is one of the most studied problem of network science, but also one of the most difficult to properly define.

Partitions/Overlap

We must differentiate two types of node grouping:

1. A **Partition** of a graph is a division of its nodes such as each of them belongs to one and only one group.
2. Overlapping communities/blocks allow, on the contrary, nodes belonging to several groups. Unless specified differently, they also allow nodes to belong to no group.

Algorithms looking for partitions are much more common than those searching for overlapping groups, due to the increased complexity of the later task. Overlapping community detection is, nevertheless, an active field of research.

Community structure

The idea of having a network structured in **communities** is defined as an analogy with communities in social networks. Communities are therefore defined (informally) as groups of nodes that are strongly connected between themselves (**high internal density**) and more weakly connected to the rest of the network (**low external density**).
This definition however cannot be translated unambiguously into a mathematical formulation. The problem of **community detection**, or community discovery, is therefore complex to define.

Block structure

The general idea of the block structure is that the probability to observe an edge between two nodes is a function of the blocks they belong to. Contrary to communities, no assumption is made *a priori* about the respective values of those probabilities: **they can be high between nodes belonging to the same blocks or to different blocks**, and can **differ for each pair of block**.

Definition

C	a *community partition*, or, more generally, a set of set of nodes
c_i	community i, a set of nodes

Modularity

The most famous quality function to measure the *quality* of partitions is the **Modularity**. Introduced in[a], it is defined for a partition C and a graph G as the difference between the fraction of observed internal edges and the expected fraction of internal edges if G were rewired according to a configuration model, i.e., preserving the degrees of nodes.

$$Q = \frac{1}{2L} \sum_{uv} \left[A_{uv} - \frac{k_u k_v}{2L} \right] \delta(c_u, c_v)$$

with $\delta(c_u, c_v)$ the kronecker delta, i.e., $\delta(c_u, c_v) = 1$ if u, v belong to the same community, 0 otherwise. It can be rewritten for convenience as a sum over communities:

$$Q = \frac{1}{L} \sum_{i=1}^{|C|} (L_i - \frac{K_i^2}{4L})$$

with $L_i = L(H(c_i))$ the number of edges inside community i and $K_i = \sum_{u \in c_i} k_u$ the sum of degrees of nodes in community i.

[a] Girvan and M. E. Newman 2002.

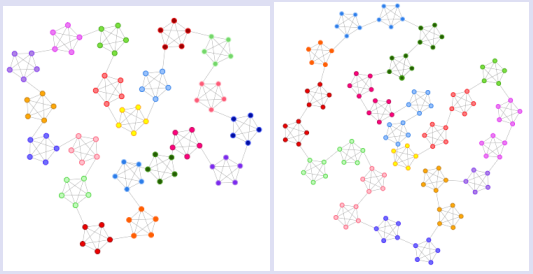
Modularity: null model

The modularity as expressed above compares the number of edges inside communities to the expected number of edges in a **null model**, i.e., a randomized version of the graph. In the original version, this null model is the **configuration model** (as easily recognized in the $\frac{k_u k_v}{2L}$ of the original formula).
Variants of the modularity have been proposed using different null models[a], for instance an ER null model, or a gravity model to take into account the effect of geographic distance[b]

[a] Jutla, Jeub, and Mucha 2011.
[b] Expert et al. 2011.

Modularity: resolution limit

It is important to remember that the Modularity is (only a) **quality function**, **not a definition** of the quality of communities. An important drawback of Modularity is known as the **resolution limit**[a]. It says that partitions of maximal modularity are biased toward a particular *scale*, i.e., for a graph of a give size (#nodes, #edges), communities smaller than a certain size cannot be found. The typical example of this limit is the clique-ring structure (set of cliques connected by a single edge), in which the expected partition is to have one community by clique, while the solution of highest modularity put several cliques in the same community, when we increase the number of cliques.



Examples of networks with 20(left) and 26(right) cliques of size 5. Colors represent communities found by a modularity maximization algorithm.

[a] Fortunato and Barthelemy 2007.

Modularity and random networks

Another well known limitation of a Modularity maximization approach is that it finds communities with high scores in random networks: since it is not *adjusted for chance*, random flucutations in a random network are mistaken for meaningful structure in the network.

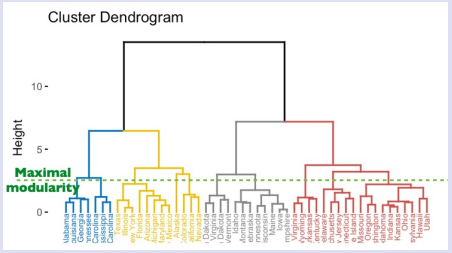
Multi-resolution Modularity

A simple solution has been proposed to the limit of resolution, consisting in adding a resolution parameter λ to *tune* the desired resolution[a], i.e., $(L_i - \frac{1}{2}K_i^2)$ becomes $(L_i - \lambda\frac{1}{2}K_i^2)$. It raises or shrinks the expected number of edges inside communities. It requires, however, to choose a proper value for λ, i.e., to choose arbitrarily a scale for communities.

[a] Reichardt and Bornholdt 2006.

Mod. maximization: Girvan Newman

Several of the most popular community detection algorithms have as objective to discover the partition of highest modularity. This is a difficult problem, and thus existing approaches are based on heuristics.
The original method by Girvan and Newman[a] first builds a dendrogram by iteratively removing the edge of highest betweenness. It is called a *divise* approach: At the top of the dendrogram, there is a single community, then 2, 3, 4 etc., until each node is in its own community. Modularity is used as a criterium to *cut* the dendrogram.



[a]Girvan and M. E. Newman 2002.

Mod. maximization: Louvain method

The Louvain method[a] is certainly the most used method for community detection. Its objective is to optimize the modularity using a greedy, agglomerative approach:
Step 1: Optimizing modularity at a hierarchical level

- Each node starts in its own community
- Repeat until convergence:
  1. **FOR** each node, compute the gain in modularity of adding it to the community of each of its neighbors
  2. choose the decision that increases the most the modularity (the best decision can be to keep the node in the same community)

Step 2: Global algorithm

- Repeat until convergence:
  1. Optimize modularity for the current hierarchical level according to Step 1
  2. Move to a higher hierarchical level by computing an **induced network**: each community becomes a node, the weight of the edge between nodes/communities i and j corresponds to the number (sum of weights) of edges between nodes of c_i and nodes of c_j.

The result of Louvain algorithm is a **hierarchy** of communities.

[a]Blondel et al. 2008.

Louvain method strengths and weaknesses

The main reason explaining the popularity of the Louvain method to this day is its **scalability**: The algorithm is very scalable in practice on real graphs, for several reasons: 1)It is a greedy approach, 2) By checking only the interest of moving to neighbor's communities, it benefits from the sparsity of networks, 3)Modularity gains of a partition change can be computed locally, using its definition as a sum of independent values for each community.
Another advantage of the Louvain method is that results at lower hierarchical levels can naturally mitigate the problem of the resolution limit, for instance on the ring clique example, Louvain find each clique in its community at the first level, and only in a second level yield the problematic partition.
However, it has also be shown in[a] that the greedy nature of the algorithm could lead to having counter-intuitive structures, such as disconnected communities. The authors of the paper proceed to introduce a variant of the algorithm called Leiden, solving this problem.

[a]Traag, Waltman, and Eck 2019.

Infomap

Infomap[a] is a method based on an objective function different from the Modularity. Its objective is to **Minimize the description length of an average random walk** in the network, i.e. maximize the **compression** of the description of such a walk. More formally, the code length to minimize for partition C is described as:

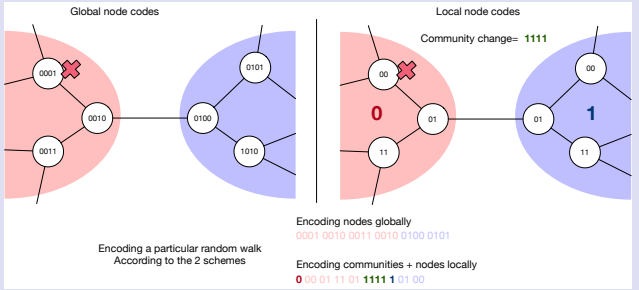
$$H(C) = qH(\curvearrowright) + \sum_{i}^{|C|} p^i H(\circlearrowright_i)$$

with q the probability for a move to be between modules,$H(\curvearrowright)$ the amount of information (bits) required to encode a move between modules, p^i the probability for a move to be inside community i and $H(\circlearrowright_i)$ the amount of information required to encode a move inside community i.
A greedy optimization algorithm, for instance one similar in nature to the one of Louvain, can then be used to minimize this description length.
Compared with Modularity, the main advantage of this approach is that it does not systematically find communities in random networks. It is known also to suffer from a form of resolution limit. Several improvements have been proposed, for instance to discover hierarchical partitions.

[a]Rosvall and Bergstrom 2008.

Infomap Algorithm intuition

Illustration of the intuition behind Infomap random walk compression. For a more accurate depiction, check the excellent illustration by the authors[a]. The real encoding length is not computed explicitly, but estimated for an infinite random walk based on information theory principles.



[a]Rosvall and Bergstrom 2008.

Stochastic Block Models (SBM)

A stochastic block model is a random graph model defined by:

b	$n \times 1$ vector such as b_i describes the index of the block of node i.
E	$k \times k$ **stochastic block matrix**, such as E_{ij} gives the number of edges between blocks i and j (or the probability to observe an edge between any pair of nodes chosen with one node in each of the two blocks).
θ	$n \times 1$ vector representing the node degrees (optional)

SBM inference

The objective of a community/block detection algorithm based on the SBM principle is thus to perform **SBM inference**, i.e., to find the parameters of the SBM that best explain the observed graph, usually in term of maximizing the likelihood. Said differently, we search –among a certain class of models– the model that has the highest probability to generate the observed graph. Note that for an observed graph, for each partition in blocks b, there is a single block matrix E that is relevant to consider, that can be found simply by counting the number of edges actually present between blocks in the graph.
More formally, the objective is:

$$b := \underset{b}{\operatorname{argmax}}\, P(A|b)$$

SBM: Simple Graphs

Assuming a simple graph, the probability to observe a graph A given a partition b is computed as the product, for each pair of node, of the probability to obtain the observed situation: edge or no-edges. It uses a Bernouilli distribution.

$$p(A|b,E) = \prod_{i<j} \begin{cases} E_{b_i b_j} & \text{if } A_{ij} = 1 \\ 1 - E_{b_i b_j} & \text{if } A_{ij} = 0 \end{cases}$$

SBM: Poisson

Other assumptions on the distribution can be made. For scalability reasons, a common one is to assume a poisson distribution of edges

$$P(A|b,E) = \prod_{i<j} \frac{(E_{b_i b_j})^{A_{ij}}}{A_{ij}!} e^{-E_{b_i b_j}}$$

Assuming that A is sparse, Poisson or Bernouilli tends toward the same results for large graphs.

DC-SBM: Degree Correction

Much as the Modularity null model preserving nodes degrees, modern SBM usually integrate a degree correction

$$P(A|b,E,\theta) = \prod_{i<j} \frac{(\theta_i \theta_j E_{b_i b_j})^{A_{ij}}}{A_{ij}!} e^{-\theta_i \theta_j E_{b_i b_j}}$$

as E, the optima θ is deducted from b, a:

$$\hat{\theta}_i = \frac{k_i}{\kappa_{b_i}}$$

with κ_{b_i} the sum of degrees in i's cluster

SBM: quality function

The probability to observe a graph can be simplified[a], using log transformation and getting rid of constants, into objective functions as simple to compute as the modularity.
Poisson, with self-loops (m=nb edges, n=nb nodes)

$$\mathcal{L}(A|b) = \sum_{rs} m_{rs} \log \frac{m_{rs}}{n_r n_s}$$

Poisson, with self-loops, degree corrected (κ=sum of degrees)

$$\mathcal{L}(A|b) = \sum_{rs} m_{rs} \log \frac{m_{rs}}{\kappa_r \kappa_s}$$

[a] Karrer and M. E. Newman 2011.

SBM: number of blocks

As defined until now, the inference has a trivial solution: each node being in its own block, and $A = E$. The probability of such a model to generate the observed graph is maximal (1 for simple graphs). The solution to this problem is often to fix the number of blocks. This approach is not satisfying in the general case, when we do not know the expected number of blocks.

SBM: infer the number of blocks

More recently, approaches[a] have been proposed to infer also automatically the number of blocks. They adopt an approach from Information Theory called the Minimum Description Length (MDL), whose principle is to find the description which reduces the total cost of describing a graph, by minimizing both 1)The quantity of information needed to encode the graph, knowing that it is generated by a given model, and 2)The quantity of information needed to encode the model itself. Intuitively, a model with few blocks requires little information to be described, contrary to a model with many blocks. But a model with many blocks is more **constrained**, the graphs it generates are more *specific*, and therefore can be described at a lesser cost, knowing the model.

[a] Peixoto 2019.

SBM infering blocks: equation

More formally, we can decompose the probability of observing a graph and a model as:

$$P(A|b) = P(A|\theta,E,b)P(\theta|E,b)P(E|b)P(b)$$

with the last three probabilities being *priors*. Said differently, we can define the number of bits required to encode a model as $L = -log_2 P(\theta,E,b)$, the number of bits necessary to encode a graph knowing the model as $S = -log_2 P(A|\theta,E,b)$ and thus the total cost to minimize as $S + L$.
The objective thus becomes:

$$b := \underset{b}{\operatorname{argmin}} -log_2 P(\theta,E,b) - log_2 P(A|\theta,E,b)$$

Variants of the SBM

Group inference using SBM is a very active field of research, and many variants have been proposed, including degree-corrected, nested, Overlapping, Mixed membership SBM, etc.
An introduction to the state of the art can be found for instance in[a].
A python library[b] exists to apply recent methods to observed graphs.

[a] Lee and Wilkinson 2019.
[b] https://graph-tool.skewed.de

Evaluation of Community structures

Since there isn't a unique accepted definition of what are good communities, the evaluation of the quality of a partition or set of communities is not a trivial task.
There are two main approaches:

- **Internal evaluation** consists in using *quality functions* (e.g., Modularity) to give a score for a pair partition/graph
- **External evaluation** consists in comparing a computed partition to a **ground truth** reference partition.

Internal evaluation can be used to evaluate the quality of communities found on a network of interest, while External evaluation is mostly used to asses the quality of algorithms on benchmarks.

Internal Evaluation - Global

Objective quality function themselves can be understood as different **definitions of community structures**. While some methods try directly to optimize one of those quality functions, some other methods are based on different principles (e.g., clique-based communities, consensus reaching based on game-theory, etc.). Quality functions can therefore be used a posteriori to assess the quality of a partition.
Typical examples are:

- **Modularity**
- **Description lenght**, as in Infomap or SBM
- **Surprise**[a] evaluates the departure of the observed partition from the expected distribution of nodes and links into communities given a null model.

[a] Aldecoa and Marin 2013.

Internal Evaluation - By community

Some quality functions are defined at the level of **individual communities**, instead of having one score for the whole partition. Those individual scores can however be combined to provide a global score, for instance using a weighted average. Some of the most popular are[a]:

- **Conductance**, the fraction of all stubs of nodes in the community that points outside of it
- **ODF**, Out Degree Fraction, the average for every node of its fraction of neighbors inside the community
- **Internal Transitivity**, the clustering coefficient inside the community
- **Scaled density**, the ratio of the node density to the total graph density

[a] leskovec2010empirical

External Evaluation

Partitions obtained by a given method can be compared with a ground truth. This approach is used on real networks, with a ground truth coming from metadata (e.g., classes in a network of social interactions between students), and on synthetic networks, with communities known by construction.
Although this is still discussed in the literature, it is mostly accepted that the evaluation on real networks using this approach is problematic[a], because there is no guarantee that the labels used as ground truth are indeed related to the **topological structure** of the network, which is what communities are about.
Most popular methods for partitions comparisons are:

- **NMI**, Normalized Mutual Information, and its adjusted for chance variant, **AMI**.
- **ARI**, Adjusted Rand Index

But more generally, any method for cluster comparison can be used[b]

[a] Peel, Larremore, and Clauset 2017.
[b] Dao, Bothorel, and Lenca 2020.

Overlapping communities

For many types of networks, the real organization of networks is thought to be **overlapping**, i.e., each node can belong to several communities. Think for instance of your personal social network: some of your family members might also be part of a group of friends, or some of your friends from high school might also be part of your friends from university, which are otherwise distinct groups.
Detecting overlapping clusters is considered harder than non-overlapping ones, for at least two reasons: the search space (number of possible solutions) is much larger (and even infinite), and defining what good communities are is even harder, since there isn't the natural limit for each edge to be either internal or external.
A large number of methods have nevertheless been proposed[a]. Extensions of non-overlapping quality functions have been proposed, such as the overlapping Modularity[b], or overlapping NMI[c], but they are not as widely used as their more constrained counterparts.

[a] Xie, Kelley, and Szymanski 2013.
[b] Nicosia et al. 2009.
[c] McDaid, Greene, and Hurley 2011.

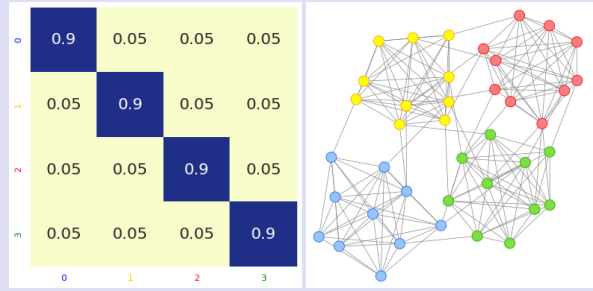
Other meso-scale structures

Beyond the usual community structure, other types of network structural organizations have been proposed and studied. Some of the most widely known are:

- **Link communities**, in which communities are defined as *sets of links*. Searching for (non-overlapping) partitions of edges yield a structure in which nodes naturally belong to several groups, i.e., a community can corresponds to *familial* edges, another to *professional* edges, etc. (Ahn, Bagrow, and Lehmann 2010)
- **Fuzzy communities**, in which nodes belong to (often several) communities with a certain probability or strength (Liu 2010)
- **Core-Periphery structure**, already defined when we introduced the notion of *k-cores*
- **Nestedness**, corresponding to a network with a hierarchical organization such as elements with few connections tends to be connected to a subset of the neighbors of a *parent* node. (Pawar 2014)
- **Spatial organization**, in which the probability of observing an edge between nodes depends on their distance. (Barthélemy 2011)

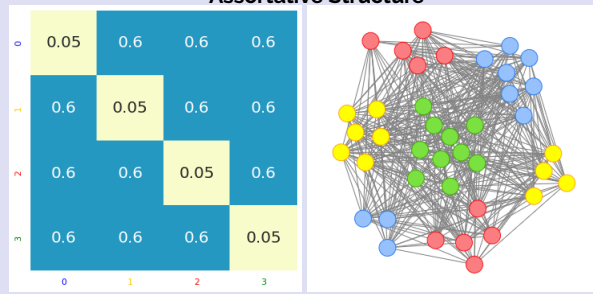
Meso-scale organization -1

Examples of different types of organization that can be obtained using block structure



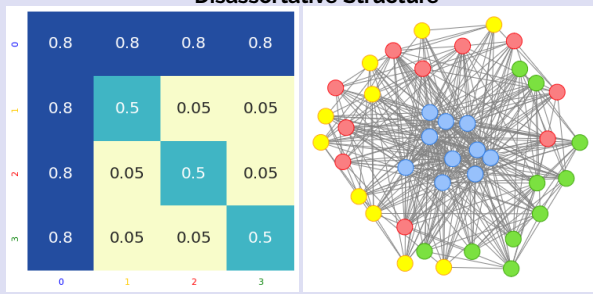
(1)

Assortative Structure



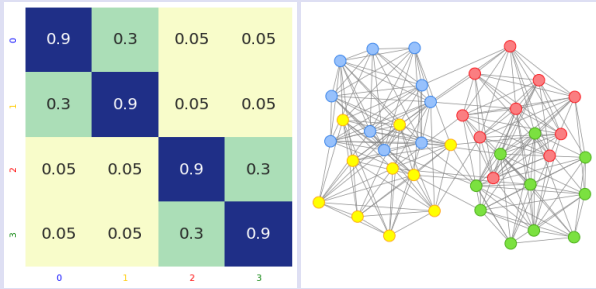
(2)

Disassortative Structure



(3)

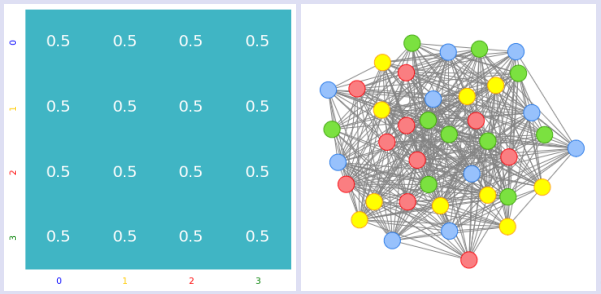
Core Periphery Structure



(4)

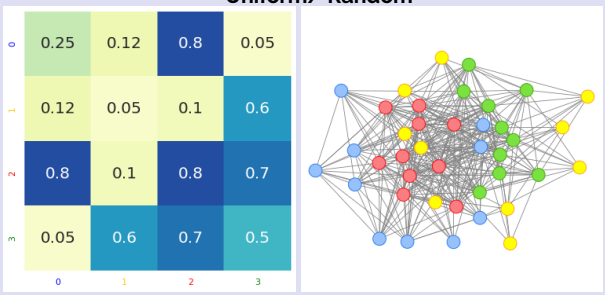
Hierarchical Structure

Meso-scale organization -2



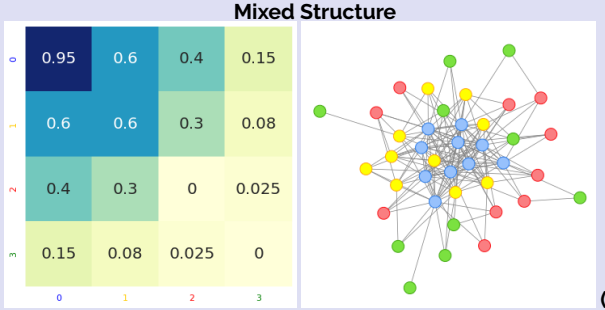
(5)

Uniform/ Random



(6)

Mixed Structure



(7)

Nested Structure

Going Further

Surveys: (Fortunato 2010) (Fortunato and Hric 2016)
On community detection approaches: (Rosvall, Delvenne, et al. 2019)
On stochastic Block Models:Funke and Becker 2019
Survey overlapping communities: (Xie, Kelley, and Szymanski 2013)
Community detection in dynamic networks (Rossetti and Cazabet 2018)
On ground Truth and community detection: (Peel, Larremore, and Clauset 2017)
Community detection in neuroscience (Betzel 2020)

Assortativity

Assortativity - Homophily - Mixing Patterns

A network is said to be **assortative** or to demonstrate **homophily** if its nodes tend to connect more with other nodes that are **similar** than to nodes that are different.
Similarity in this case must be understood in term of nodes properties. Some typical examples can be age, gender, language, political beliefs, etc.
Homophily is considered a common feature of many networks, in particular social networks[a], as reflected in the aphorism *Birds of a feather flock together*.
Typical examples would be *age, gender, ethnicity* or *politicla opinions* in social networks networks such as Twitter[b]

[a] McPherson, Smith-Lovin, and Cook 2001.
[b] McPherson, Smith-Lovin, and Cook 2001.

Disassortativity - Heterophily

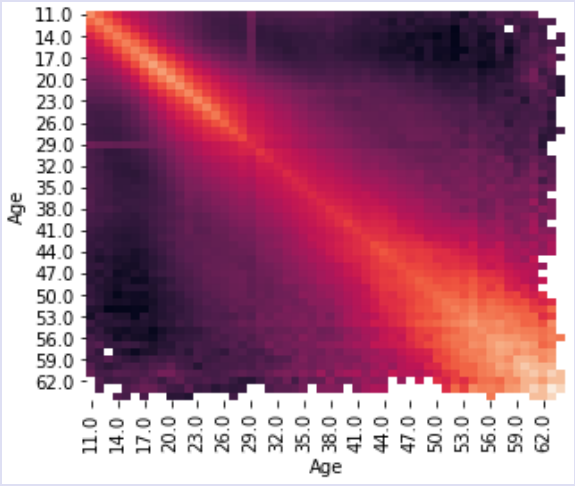
Some networks can also demonstrate **heterophily**, or **disassortativity**, i.e., a greater number of connections with nodes that are different (for instance, in a sentimental relationship network, women tend to connect more with men than with other women, and reciprocally).

Mixing Patterns

The notion of nodes connecting to each other with preferences based on their attributes can be generalized to the concept of **Mixing Patterns**. Beyond homophily/heterophily, nodes with property p_1 can be preferentially connected to nodes with property p_2 (and not p_3 or p_4) while nodes having property p_3 can have a preference for nodes having the same property, for instance.

Mixing Patterns - example

Example of mixing patterns of age in the Pokec[a] social network. Better examples of rich mixing patterns can be found in[b].



We can see that there is some level of assortativity (high values on the diagonal), but that there are also some more complex mixing patterns, for instance between age 10 and 40, approximately, here interpreted as child-parents relationships.

[a] https://snap.stanford.edu/data/soc-pokec.html
[b] Del Valle et al. 2007.

Note on interpreting homophily

Homophily can be a link creation mechanism (nodes have a preference to connect with similar ones, so the network end up to be assortative), or a consequence of influence phenomenons (because nodes are connected, they tend to influence each other and thus become more similar).
Without access to the dynamic of the network and its properties, it is not possible to differentiate those effects.

Categorical or Numerical homophily

Attributes of nodes can be either categorical (no natural order between values, discrete number of possible values), or numerical (natural order, discrete or continuous). Although the general idea remains the same, the way to compute homophily differs according to type of attributes we are interested in.

Assortativity Index - Definition

When the property for which we study homophily is **categorical**, homophily can be defined[a] by comparing the fraction of edges that connect nodes of the same category, and the expected value of such edges if the network was random. More formally, it is expressed as:

$$r = \frac{\sum_i e_{ii} - \sum_i a_i^2}{1 - \sum_i a_i^2}$$

where e_ii is the fraction of edges connecting two nodes of category i, and a_i the fraction of all edges connected to a node of category i (sum of degrees divided by number of edges).

[a] M. E. Newman 2003.

Assortativity index - Example

Let's see a fictional example of how to compute the assortativity index. Nodes are individuals, edges represent for instance some social interaction. Columns/Rows correspond to blood types, and numbers are expressed in fraction of the total number of edges.

Blood Types	A	AB	B	O	a_i
A	0.30	0.05	0.1	0.05	*0.5*
AB	0.05	0.05	0	0	*0.1*
B	0.1	0	0.2	0	*0.3*
O	0.05	0	0	0.05	*0.1*
a_i	*0.5*	*0.1*	*0.3*	*0.1*	**1**

$$r = \frac{(0.3+0.05+0.2+0.05) - (0.5^2+0.1^2+0.3^2+0.1^2)}{1-(0.5^2+0.1^2+0.3^2+0.1^2)} = \frac{0.6+0.36}{1-0.36} = 0.375$$

Asortativity index - Properties

An assortativity index of $r = 0$ means that the network has no assortative mixing, $r = 1$ corresponds to a perfectly assortative network (edges exist only between nodes of the same category), and $r = -1$ to a perfectly disassortative network (no edge between nodes of the same category).

Assortativity and Modularity

Assortativity is related to the Modularity, a measure of the quality of *communities*, by the following relation:

$$r = \frac{Q}{Q_{max}}$$

Indeed, $\sum_i e_{ii} - \sum_i a_i^2$ corresponds to the definition of the Modularity, while $1 - \sum_i a_i^2$ corresponds to the maximal value that the Modularity could reach if all nodes were in the same communities.

Homophily for numeric variables

When the property for which we study homophily is **numeric**, homophily r can be defined as the **Pearson Correlation Coefficient** between values at both end of each edge. For details, see M. E. Newman 2003.

Numeric Assortativity index - Properties

Homophily $r = 0$ means that the network has no assortative mixing, $r > 0$ corresponds to an assortative network (nodes with high values tend to connect to high values), and $r < 0$ to a disassortative network (nodes with high values are preferably connected to low values).

Degree assortativity

Degree assortativity[a], sometimes simply called *assortativity*, is a particular case of homophily measured in term of node degrees, i.e., the numerical value associated to each node is its degree.
The existence of a degree assortativity can be interpreted in term of a *rich club phenomenon*: hubs prefer to connect to other hubs.
ER, Configuration and BA random graph models have a degree assortativity equals to 0, while many real networks have positive values, and some negative ones.

[a] M. E. Newman 2003.

Limits of Assortativity

A limit of assortativity coefficients as we have defined them is that they summarize the whole network as a single value. However, different parts of the network might have different types of assortativity. For instance, the 3 following graphs have a similar value of assortativity (≈ 0).

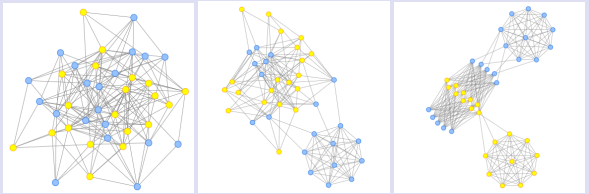


Figure inspired from[a], in which the authors propose a local measure of **multiscale assortativity**. Another longer distance assortativity is defined in .

[a] Peel, Delvenne, and Lambiotte 2018.

Going Further

- A survey on the topic (Noldus and Van Mieghem 2015)
- Non-local assortativity (Peel, Delvenne, and Lambiotte 2018) (Rossetti, Citraro, and Milli 2021)

Spreading Processes

Spreading - Diffusion

In many real world situations, networks can be seen as **support** of **spreading** or **diffusion** processes. Typical examples are diffusion of information, innovation, rumors, biological or computer virus, adoption of ideas or products, etc. The support can be social networks of interactions, social media platforms of computer networks, to name a few.
A typical way to model such a process is to assign **properties** to nodes, categorical or numerical, to represent the current status of the node relatively to the process (e.g., *Susceptible, Infected*).
We can also call such a process **Dynamic On Networks**, as opposed to **Dynamics Of Networks**.

SI - SIR - SIS

Three of the most popular models of diffusion in epidemiology are the **SI**, **SIR** and **SIS** models. Letters correspond to the states in which individuals can be according to the model:

- **Susceptible**: Individual is not Infected
- **Infected**: Individual is Infected
- **Recovered/Removed**. Individual cannot be infected again (Considered cured or dead)

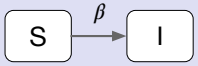
All individuals are in one of the states allowed by the model, and we define:

$s(t)$	Fraction of individuals in Susceptible state at time t
$i(t)$	Fraction of individuals in Infected state at time t
$r(t)$	Fraction of individuals in Recovered state at time t
i_0	Initial($t = 0$) fraction of infected individuals

SI

In the SI model, individuals can be only in two states, Susceptible and Infected. Susceptible ones can become Infected, and Infected individuals rest in this state indefinitely. Parameters are:

τ	**Infectivity**: probability that the contact between an *Infected* individual and a *Susceptible* one results in the infection of the Susceptible.
$\hat{c}$	*Contact rate*: average number of contact per person per time
β	**Effective contact rate**, $\beta = \tau\hat{c}$, number of newly infected individuals by each infected individual in a population in which everyone else is susceptible.



SI - characteristics

Each of the i infected individuals infects in average β contacts, but only $s = (1 - i)$ of its contacts are indeed susceptible. More formally using differential equations:

$\frac{di}{dt}$	**Rate of new infection**: $\frac{di}{dt} = \beta is = \beta(1 - i)i$
$i(t)$	**Infected fraction**[a]: $i(t) = \frac{i_0 e^{\beta t}}{1 - i_0 + i_0 e^{\beta t}}$
$s(t)$	**Susceptible fraction**: $1 - i(t)$

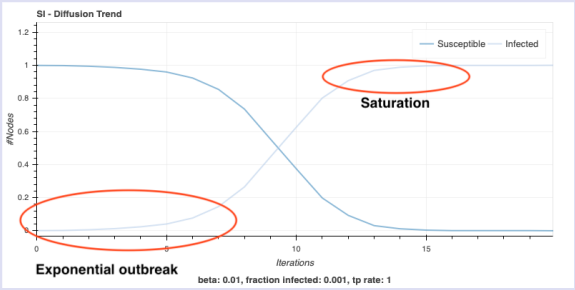
The process can be separated in three steps:

- At first, the fraction of infected individuals **Grows exponentially** until a large fraction of the population is infected. (i is small, $\frac{di}{dt} \approx \beta i \Rightarrow$ exponential)
- Due to **saturation**, the infection of the last individuals is slow
- The growth is faster and faster until half the population is infected ($\text{argmax}_{x,y}(x(1 - x)) : x = y = 0.5$).

If $\beta > 0$, everyone is infected at the end of the process.

[a] Barrat, Barthelemy, and Vespignani 2008.

SI - Sketch



SI - Application

An example in which this model can be appropriate is for diffusion of innovation: being infected means buying or using a new service, product or technology whose usage becomes widespread in society, e.g., television, cell-phone, internet, Netflix, etc.

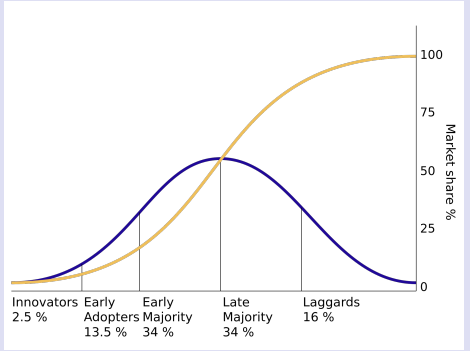


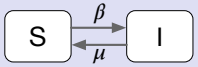
Illustration of Rogers's innovation adoption model. Image source[a]

[a] `https://en.wikipedia.org/wiki/Diffusion_of_innovations`

SIS

When modeling real viruses, it is often useful to consider that infected individuals can **recover**, i.e., go back to the susceptible state, without becoming immune, such as for common cold or influenza. This can be modeled using the **SIS** model.
Additionally to β, the SIS model requires another parameter:

μ	**recovery rate**: probability that an *Infected* individual go back to the susceptible state per unit of time.



SIS - characteristics

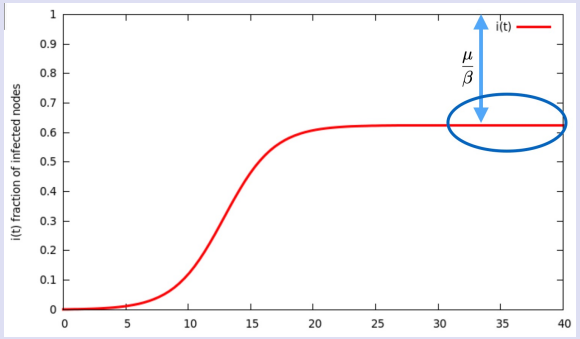
Intuitively, the fraction of infected individuals is now reduced by those switching to the susceptible state, more formally:

$\frac{di}{dt}$	**Rate of new infection**: $\beta i(1 - i) - \mu i = i(\beta - \mu - \beta i)$
$i(t)$	**Infected fraction**[a]: $\left(1 - \frac{\mu}{\beta}\right) \frac{Ce^{(\beta-\mu)t}}{1+Ce^{\beta-\mu)t}}$

For large times, $i(t) \to 1 - \frac{\mu}{\beta}$, i.e., the fraction of infected individuals stabilize around a value which depends only of parameters μ and β.

[a] Barrat, Barthelemy, and Vespignani 2008.

SIS - Sketch



λ ratio or (R_0)

In the SIS model, an important notion is the λ ratio, also called R_0.

$$R_0 = \frac{\beta}{\mu}$$

R_0 can be understood as the average number of individuals that will be infected by an infected individual, **in a population in which all other nodes are Susceptible**. R_0 is a property of the model and **do not change with time**.
Looking at the R_0 is important in the early stage of the epidemic:

- if $R_0 > 1$, **there will be an outbreak**
- if $R_0 < 1$, **the epidemic will disappear naturally**.

If R_0 is just above 1, the outbreak also can stop naturally by chance in the early stage.

SIR

In many spreading situations, infected individuals can themselves switch to a new state, usually called **Recovered**, which can represent either an acquired immunization or its removal (death, computer failure, etc.). In any case, Recovered individuals cannot infect nor be infected.
Additionally to β, the SIR model requires another parameter:

γ	**recovery rate**: probability that an *Infected* individual switch to the Recovered state per unit of time.

SIR - characteristics

Intuitively, the fraction of infected individuals is now reduced by those switching to the recoved state, more formally:

$$\frac{ds}{dt} = -\beta is, \frac{di}{dt} = \beta is - \gamma i, \frac{dr}{dt} = \gamma i$$

- The initial steps of the outbreak still follow an exponential growth
- The fraction of infected nodes reach a peak and then decreases
- The fraction of recovered can saturate below 1
- The λ ratio is defined as $\lambda = \frac{\beta}{\gamma}$

SIR - sketch

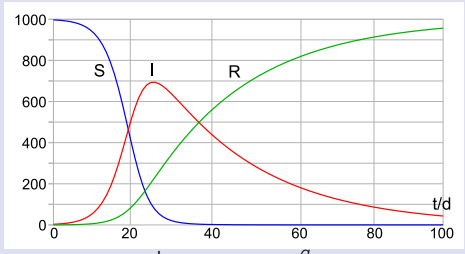


Image source: [a]

[a] `https://en.wikipedia.org/wiki/Compartmental_models_in_epidemiology`

Other non-network models

Epidemic modeling is a large and rich scientific topic, thus those models are nowadays considered *toy models*, too simple to model real epidemics. Most used model thus include other factors, such as **natural population dynamic** (birth, natural death, etc. for long term dynamics), **population segmentation** (tau might differ among subsets of populations, e.g., elderly, maried couples, etc.), **population mixing** ($\hat{c}$ might vary between members of a subpopulation and another), etc.

Network models

A natural way to add more details to a spreading process is, instead of considering an homogeneous population, or a population composed of homogeneous subpopulations, to study the diffusion on a network representing the structure of the population.

Notation change on networks

$\hat{c}$ has no meaning in networks (its role is played by the network structure), so by convention we use $\beta = \tau$: the probability for a node to infect each of its neighbor at each step.

Homogeneous Networks

If we consider an **homogeneous random network** in which all nodes have degree exactly k, then we can consider the spreading on this network as similar to the non-network models, with $\hat{c} = k$. For instance, the SI model becomes:

$$\frac{di}{dt} = \beta\langle k\rangle(1-i)i$$

Note that in practice, there are a few differences, such as a switch from a continuous to a discrete setting, but they have no consequences on large graphs.

ER Networks

In ER random graphs, for large graphs, we have seen that $k \approx \langle k\rangle$, thus the same model description still holds, with the approximation being better for larger networks.

R_0 on networks

In homogeneous or ER networks, R_0 is naturally defined as $\frac{\beta\langle k\rangle}{\mu}$
Another way to express the same thing is that, if we define $R_0 = \frac{\beta}{\mu}$, then the epidemic threshold is not equal to 1 but to $\frac{1}{\langle k\rangle}$

Heterogeneous Degrees - Degree blocks

We have seen that most real networks have an heterogeneous degree distribution. To study analytically the effect of this property, a method is to use a **degree block approximation**: we consider all nodes with a given degree as an homogeneous groups(degree block), and analyze each of these groups separately.
i_k, s_k, r_k: fractions of nodes of degree k that are infected, susceptible, recovered, respectively.
The global average is the average for all degree block, weighted by the fraction of nodes in each block. $i = \sum_k P(k) i_k$ with $P(k)$ the fraction of nodes having degree k.
The same holds for global s and r.

Heterogeneous Degrees - SI

For the SI model, we know that all nodes are infected in the end, but what may vary is **speed** of the process.
The speed of diffusion by degree block can be expressed as:

$$\frac{di_k}{dt} = \beta k(1 - i_k)\Theta_k$$

with Θ_k being the fraction of infected neighbors of a node with degree k.

Heterogeneous Degrees - Θ_k

Θ_k represents the fraction of infected neighbors of a node with degree k. If the network is homogeneous, $\Theta_k(t) = i(t)$. If the graph is heterogeneous, nodes with different degrees have different probabilities of being infected: higher degree nodes are, by definition, more exposed, since they have more chances of being infected at each step.
This is important for two reasons:

- Due to the friendship paradox, nodes are more likely to be connected to large nodes than to small ones
- In real networks, we have seen that there is often a degree assortativity, thus nodes of a given degree have different degrees in their neighborhood.

We can thus define Θ_k as the probability to be connected to nodes of a given degree and or their respective probability of being infected $\Theta_k = \sum_{k'} P(k'|k) i_{k'}$, with $P(k'|k)$ the probability that a node with degree k connects to a node with degree k'.
For simplicity, **we assume no degree-degree correlation**.
$P(k'|k)$ can then be expressed as the fraction of all edge stubs attached to nodes of degree k', indepedent of the k under study:

$$P(k'|k) = \frac{k'P(k')}{\sum_{k''} k''P(k'')} = \frac{k'P(k')}{\langle k \rangle}$$

And:

$$\Theta_k = \Theta = \frac{\sum_{k'} k'P(k')i_{k'}}{\langle k \rangle}$$

Heterogeneous Degrees - SI - time scale

From previous equations, it can be shown[a] that the **time scale** τ of the process, i.e., a measure inversely proportional to its speed, is $\tau = \frac{\langle k \rangle}{\beta(\langle k^2 \rangle - \langle k \rangle)}$.
Thus, for a given average degree $\langle k \rangle$ and a given β, **the more heterogeneous the degrees, the faster the diffusion**.
If the degree distribution follows a power law of exponent $\alpha < 3$, we have seen that $\langle k^2 \rangle$ diverge towards infinity, thus τ tends toward 0, thus the diffusion is nearly instantaneous.
This can be understood as follows: if a node is connected to nearly every other node, then it has an extremely high probability to become infected immediately, and can then infect the rest of the network extremely quickly.

[a] Barrat, Barthelemy, and Vespignani 2008.

Heterogeneous Degrees - λ

For SIS and SIR models, it can also be shown[a] that the epidemic threshold λ (or R_0) is not reached when $\lambda = \frac{\beta\langle k \rangle}{\mu} > 1$ as in homogeneous networks, but when $\lambda > \frac{\langle k \rangle^2}{\langle k^2 \rangle}$.
This means that in a very heterogeneous network, **an outbreak can start even if λ is very small, and below 1**.
Intuitively, even if people recover faster than they spread the virus in average, some nodes (hubs) will nevertheless become infected, and since they can infect many others, the contagion will spread.

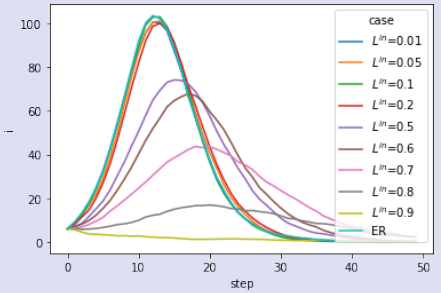
[a] Barrat, Barthelemy, and Vespignani 2008.

SIR - Experimental

When analytical solutions cannot be simply derived, empirical simulations can be used to observe the effect of network properties on diffusion processes.
In particular, these properties can be used to asses the effect of typical heterogeneity: degree-heterogeneity, belonging to blocks, spatial heterogeneity, etc.

SIR - Community Structure

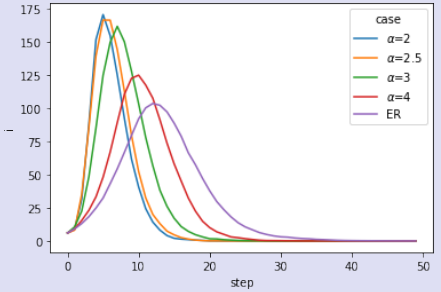
In this experiment, we compare an ER network to Stochastic Block Models.
Network parameters: $n = 1000, \langle k \rangle = 5$.
SBM parameters Number of blocks $|C| = 100$. We vary L^{in}, the fraction of all edges that are inside blocks. When $L^{in} = 0.01, p^{in} \approx p^{out} = 0.005$. When $L^{in} = 0.9, p^{in} = 0.5, p^{out} \approx 0.0005$
SIR parameters: $\theta = 0.2, \gamma = 0.5$. The initial number of infected nodes is 5, all of them in the same community structure.



We observe that the more marked the communities, the less efficient the spreading process.

SIR - Scale Free

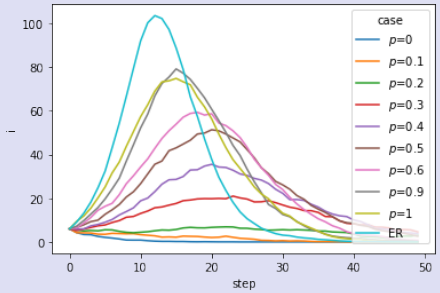
In this experiment, we compare an ER network to Configuration Models with power law degree distributions.
Network parameters:$n = 1000, \langle k \rangle = 5$. We vary the exponent of the distribution, while keeping $\langle k \rangle = 5$ constant.
SIR parameters: $\theta = 0.2, \gamma = 0.5$. The initial number of infected nodes is 5, all of them in the same community structure.



The highest the exponent of the degree distribution, the faster is the diffusion.

SIR - Spatial effect - WS

In this experiment, we compare an ER network to Watts Strogatz random graphs, varying the probability of rewiring edges. It can be understood as a model of spatial proximity: with $p = 0$, each node is connected only to its direct neighbors in the 1 dimensional space. If $p = 1$, each node is connected to exactly k random nodes.
Network parameters:$n = 1000$, $\langle k \rangle = 5$
SIR parameters: $\theta = 0.2$,$\gamma = 0.5$. The initial number of infected nodes is 5, being 5 direct neighbors.



The more nodes tend to be connected to direct neighbors in space, the slower the diffusion.

Application of diffusion models

Diffusion models can be used for several applications:

- Model fitting: better understand an actual epidemic by fitting parameters on real observations
- Predicting trends of evolution
- Control of epidemics: Given an epidemic model and a supporting network, find an optimal solution to control (accelerate or slow-down) the epidemic.

Optimal node/edge removal

One way to slow-down an epidemic consist in **removing nodes**(e.g., vaccination). The problem can be formulated as a budget constrained removal, i.e., if we can remove only x nodes/edges, which one should we choose? Based on theoretical and experimental results, heuristic solutions consist in removing: **highest degree nodes**, **highest betweeness** nodes/edges (isolating communities), long-distance edges (shortcuts) in spatial networks.

Friendship paradox and node removal

It has been proposed[a] that the friendship paradox could be used to apply budget-constrained high degree nodes preferential vaccination in real networks where finding such nodes is not possible because the whole network is unknown: instead of targeting random individuals, one could vaccinate random contacts of random individuals, thus greatly increasing the average degree of vaccinated persons.

[a] Cohen, Havlin, and Ben-Avraham 2003.

Going further

Book Dynamic processes on Networks: Barrat, Barthelemy, and Vespignani 2008
Surveys:
Analysis and Control of Epidemics: Nowzari, Preciado, and Pappas 2016
Diffusion in networks: Lamberson 2016
Impact of community structure: Stegehuis, Van Der Hofstad, and Van Leeuwaarden 2016

Dynamic Networks

Disclaimer

Dynamic network analysis as introduced here is a recent and still not fully mature field, with a large number of contributions, for which we cannot know yet which one will stand the test of time. This is therefore *my* vision of the dynamic network field *as of today*.

Ubiquity of Dynamic Networks

Most real networks are in fact dynamic: nodes and edges appear and disappear with time. Think of friendship in social networks, flight routes or any human interactions. Networks are often analyzed as static objects because 1)it's harder to obtain dynamic information, 2)Taking dynamic into account makes some analysis more difficult.
While more and more aspects of our life become linked to digital technology, datasets with fine temporal information also become more and more common.

Snapshots & Aggregated Networks

Dynamic networks can sometimes be represented as sequences of static graphs. These graphs can be obtained by two processes:

- **Snapshots** correspond to the state of a network at a particular point in time, e.g., all follower/followees relationship at a particular second
- **Aggregated Networks** are obtained by cumulating information over a period of time, e.g., in a phone call network, in the snapshot representing year 2020, an edge exists between two individuals if they called each other at least once over the year 2020.

Interactions or Relation?

Dynamic networks can be used to represent different types of real data. In particular, they can be used to represent networks of **relations** and networks of **interactions**. For instance, friendships, acquaintances, physical wires, roads, etc. can be thought as *relations*, while e-mails, phone calls, instant messages, physical contacts, etc. are *interactions*.
There is often a relation between these two notions: interactions tend to occur between entities having a relation, and/or relations tend to form between entities having interactions.

Dynamic Network Properties

Independently of the studied data, dynamic networks can have various properties:

- **Edge** presence can be **punctual** or **with duration**
- **Node** presence can be **unspecified**, **punctual** or **continuous**
- If **time is continuous**, it can be **bounded** on a period of analysis or **unbounded**
- If **nodes** have attributes, they can be **constant** or **time-dependent**
- If **edges** have weights, they can be **constant** or **time-dependent**

Vocabulary

Many different names have been used for networks changing with time, but there is no broad consensus in the literature on the meaning of those terms, unless they are used with an explicit reference to a paper defining them. Here is a list of the most popular:

- **Dynamic Networks** and **Dynamic Graphs**
- **Longitudinal Networks**
- **Evolving Graphs**
- **Link Streams** & **Stream Graphs** (Latapy, Viard, and Magnien 2018)
- **Temporal Networks**, **Contact Sequences** and **Interval Graphs** (Holme and Saramäki 2012)
- **Time Varying Graphs** (Casteigts et al. 2012)

Slowly Evolving/Degenerate

Beyond the nature of the data and the chosen representation, a critical difference defining how a dynamic network can be analyzed is whether it is a **Slowly Evolving Network (SEN)** or **Degenerate**. In a SEN network, to each instant corresponds a well defined graph, that can be studied with usual tools of network science. In a degenerate temporal network, a meaningful graph can be obtained only when aggregating it over a period Δ.

Analyzing SEN

A slowly evolving network can easily be studied by the tools already defined on static graphs. For any instant (discrete or continuous), one can compute network descriptors (density, clustering coefficient, etc.), node descriptors (centralities), reachability, etc.

Analyzing degenerate networks

A degenerate network can always be transformed into a SEN by aggregating it using time windows, fixed (yielding snapshots, i.e., discrete SEN) or sliding (yielding continuous SEN). But a more powerful solution is to study them in their original form, without loosing any temporal information through aggregation. This however requires new definitions.

Stream Graph (SG)- Definition

Stream Graphs have been proposed in[a] as a generic formalism – it can represent any type of dynamic networks, continuous, discrete, with or without duration, with the objective or redefining typical notions of graphs on dynamic networks, including degenerate ones.
Let's define a Stream Graph

$$S = (T, V, W, E)$$

T	**Set of Possible times** (Discrete or Time intervals)
V	**Set of Nodes**
W	**Vertices presence time** $V \times T$
E	**Edges presence time** $V \times V \times T$

[a] Latapy, Viard, and Magnien 2018.

SG - Time-Entity designation

It is useful to work with Stream Graphs to introduce some new notions mixing entities (nodes, edges) and time:

V_t	**Nodes At Time**: set of nodes present at time t
E_t	**Edges At Time**: set of edges present at time t
G_t	**Snapshot**: Graph at time t, $G_t = (V_t, E_t)$
v_t	**Node-time**: v_t exists if node v is present at time t
$(u, v)_t$	**Edge-time**: $(u, v)_t$ exists if edge (u, v) is present at time t
T_u	**Times Of Node**: the set of times during which u is present
T_{uv}	**Times Of Edge**: the set of times during which edge (u, v) is present

SG - Node/Edge presence

Nodes and Edges are typically present in the graph only for a fraction of its total duration, Node/Edge presence is computed as the fraction of the total times during which it is present. Note that if time is continuous and edges are discrete, we take by convention $|T| = 1$, i.e., we simply count nodes/edges presence time.

N_u	**Node presence**: The fraction of the total time during which u is present in the network $\frac{\|T_u\|}{\|T\|}$
L_{uv}	**Edge presence**: The fraction of the total time during which (u, v) is present in the network $\frac{\|T_{uv}\|}{\|T\|}$

SG - Redefining Graph notions

The general idea of redefining static network properties on Stream Graphs is that if the network stays unchanged along time, then properties computed on the stream graph should yield the same values as the same properties computed on the aggregated graph.

SG - N

The number/quantity of nodes in a stream graph is defined as the total presence time of nodes divided by the dataset duration. In general, it isn't an integer.
More formally:

$$N = \sum_{v \in V} N_v = \frac{|W|}{|T|}$$

For instance, $N = 2$ if there are 4 nodes present half the time, or two nodes present all the time.

SG - L

The number of edges is defined as the total presence of edges divided by the total dataset duration.
More formally:

$$L = \sum_{(u,v),u,v \in V} L_{uv} = \frac{|E|}{|T|}$$

For instance, $L = 2$ if there are 4 edges present half the time, or two edges present all the time.

SG - Edge domain - $L_{\max}$

In Stream Graphs, several possible definitions of $L_{\max}$ could exist:

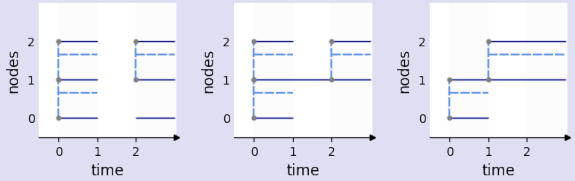
- Ignoring nodes duration: $L^1_{\max} = |V|^2$
- Ignoring nodes co-presence $L^2_{\max} = N^2$
- Taking nodes co-presence into account: $L^3_{\max} = \sum_{(u,v),u,v \in V} |T_u \bigcap T_v|$

SG - Density - d

The density in static networks can be understood as the fraction of existing edges among all possible edges,

$$d = \frac{L}{L_{\max}}.$$

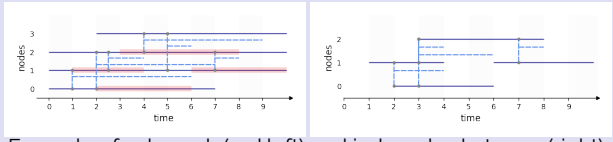
The definition can naturally be extended by using the definitions of L and $L_{\max}$ introduced on Stream Graph. In (Latapy, Viard, and Magnien 2018), the authors use $L^3_{\max}$. This definition can also be understood as the probability, if we take a time at random, and two nodes alive a that time at random, for them to be connected. Note that a common way to define the density in static networks is $d = N^2$, because N^2 is the only way to define $L_{\max}$ in static networks, unlike in Stream Graphs.



Examples of graphs with $N = 2$ nodes, $L = 1$ link, but with different densities, respectively $\frac{1}{2}$ (left), $\frac{3}{4}$ (center) and 1 (right).

SG - Clusters & Substreams

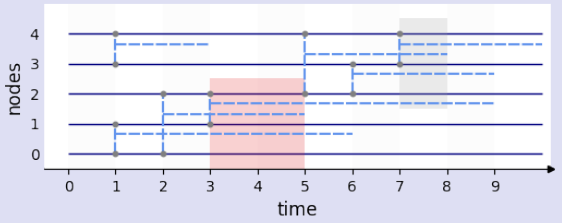
In static networks, a cluster is a set of nodes, and we have defined an (induced) subgraph of this cluster as a graph composed of the nodes of the cluster and the edges existing between those nodes. In Stream Graphs, a clusters C is as subset of W, and the corresponding (induced) substream $S(C) = (T, V, C, E(C))$, with $E(C) = \{(t, (u, v)) \in E, (t, u), (t, v) \in C\}$.



Example of subgraph (red,left) and induced substream (right).

SG - Cliques

Having defined substreams and density, we can now naturally define a clique by analogy with static networks as a substream of density 1. A clique is said to be a **maximal clique** if it is not included in any other clique.



Red and Grey are the two maximal cliques of size three in this Stream Graph.

SG - Neighborhood $N(u)$

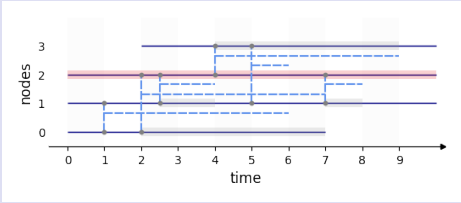
The neighborhood $N(u)$ of node u is defined as the cluster composed of node-times such as an edge-time exists between it and a node-time of u, i.e.,

$$N(u) = \{v_t, (u, v)_t \in E\}$$

SG - Degree $k(u)$

The degree $k(u)$ of node u is defined as the quantity of node in the Neighborhood of node u, i.e.

$$k(u) = |N(u)|$$



Example, the neighborhood of node 2 is highlighted in grey. $k(c) = \frac{5+2.5+5}{10} = 1.25$.

SG - Ego-network

The Ego network G_u of node u is defined as the substream induced by its neighborhood, i.e., $G_u = (T, V, N(u), E(N(u)))$.

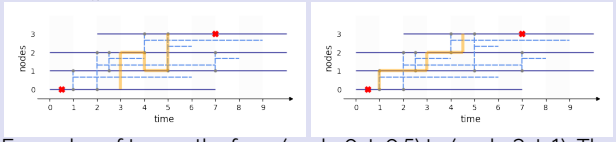
SG - Clustering coefficient

The clustering coefficient $C(u)$ of node u is defined as the density of the ego-network of u, i.e.,

$$C(u) = d(N(u))$$

SG - Paths

In a Stream Graph S=(T,V,W,E), a **path** P from node-time x_α to node-time y_ω is a sequence $(t_0, x, v_0), (t_1, v_0, v_1), ..., (t_k, v_k, y)$ of elements of $T \times V \times V$ such that $t_0 \geq \alpha, t_k \leq \omega$, $((t_i, u_i, v_i)) \in E$. We say that P **starts at** t_0, **arrives at** t_k, has **length** $k + 1$ and **duration** $t_k - t_0$.



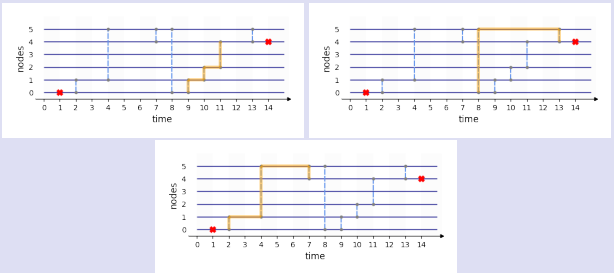
Examples of two paths from (node 0, t=0.5) to (node 3, t=1). The left one starts at 3, arrives at 5, has length 3 and duration 2. The right one starts at 1, arrives at 4.5, has length 3 and duration 3.5.

SG - Shortest - Fastest - Foremost

- **Shortest Paths**, as in static networks, are paths of **minimal length**.
- **Fastest Paths** are paths of **minimal duration**.
- **Foremost Paths** are paths **arriving first**.

Furthermore, one can combine those properties, defining for instance:
Fastest shortest paths (paths of minimum duration among those of minimal length)
Shortest fastest paths (paths of minimal length among those of minimal duration)

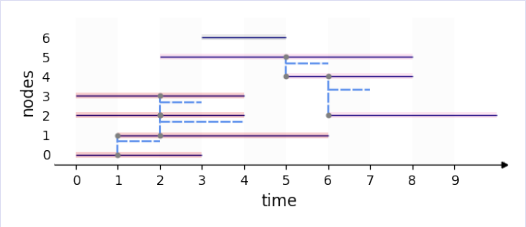
SG - Shortest - Fastest - Foremost



Fastest (top left), Shortest (top right), Foremost (bottom),

SG - Connected Components

Various definitions for connected components have been proposed for temporal networks, see (Latapy, Viard, and Magnien 2018) for details. One of the simplest one is the **weakly connected component**, defined such as two node-times belong to the same connected component if and only if there is a path from one to the other, *ignoring time*.



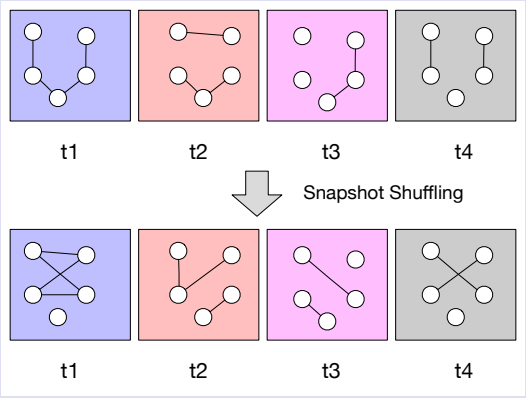
Example of a Stream Graph decomposed in 3 weakly connected components (including one composed of the singleton node 6)

Random Models

We have seen that comparing an observed network with a randomized version of it has many applications. In dynamic networks, many variants have been proposed. In (Gauvin et al. 2022), the authors consider methods defined on sequences of snapshots that conserve nodes and number of events, and grouped them in 4 main families, **Snapshot Shuffling, Sequence Shuffling, Link Shuffling** and **Timeline Shuffling**.

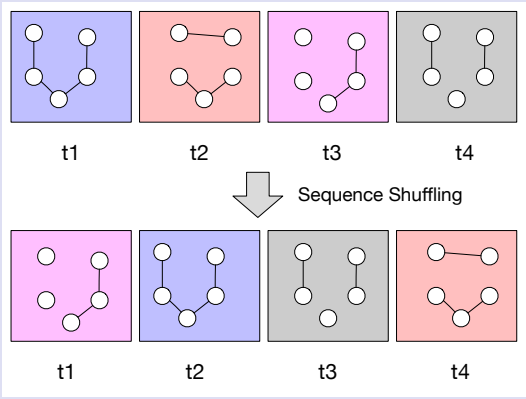
Snapshot Shuffling

Snapshot Shuffling keeps the order of snapshots, randomize edges inside snapshots. Any random model for static network can be used, such as ER random graphs or a degree preserving randomization.



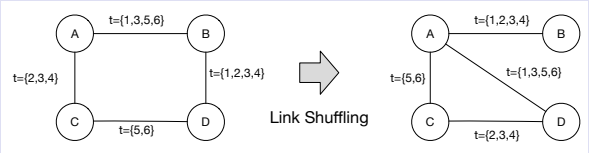
Sequence Shuffling

Sequence Shuffling keeps each snapshot identical, switch randomly their order.



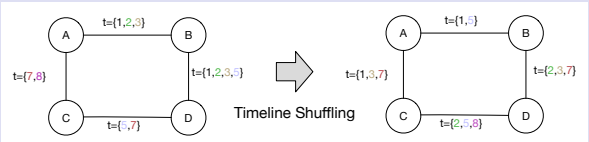
Link Shuffling

Link Shuffling keeps activation time per node pairs, randomize the aggregated graph. For instance, a simple way to achieve this is to pick two node pairs at random (connected or not) of the aggregated graph, and to exchange activation time of these node pairs, e.g.:



Timeline Shuffling

Timeline Shuffling keeps the aggregated graph, randomize edges activation time. For instance, a simple way to achieve this is to redistribute randomly activation period among all edges, e.g.:



More constrained Shuffling

Variants of these shufflings with additional constraints have been proposed, for instance the **Local timeline shuffling**, randomizing events time edge by edge, or the **Weight constrained timeline shuffling**, randomizing events while conserving the number of observations for each edge. See (Gauvin et al. 2022) for more.

Going Further

Book: Holme and Saramäki 2019
Stream Graph definition: Latapy, Viard, and Magnien 2018
Transforming dynamic networks into static networks: Kivelä et al. 2018
Dynamic Communities: Rossetti and Cazabet 2018

References

[1] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. "Link communities reveal multiscale complexity in networks". In: *nature* 466.7307 (2010), pp. 761–764.

[2] Rodrigo Aldecoa and Ignacio Marin. "Surprise maximization reveals the community structure of complex networks". In: *Scientific reports* 3.1 (2013), pp. 1–9.

[3] Albert-László Barabási et al. *Network science*. Cambridge university press, 2016.

[4] Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks". In: *science* 286.5439 (1999), pp. 509–512.

[5] Albert-László Barabási and Eric Bonabeau. "Scale-free networks". In: *Scientific american* 288.5 (2003), pp. 60–69.

[6] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. *Dynamical processes on complex networks*. Cambridge university press, 2008.

[7] Marc Barthélemy. "Spatial networks". In: *Physics Reports* 499.1-3 (2011), pp. 1–101.

[8] Richard F Betzel. "Community detection in network neuroscience". In: *arXiv preprint arXiv:2011.06723* (2020).

[9] Vincent D Blondel et al. "Fast unfolding of communities in large networks". In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.

[10] Anna D Broido and Aaron Clauset. "Scale-free networks are rare". In: *Nature communications* 10.1 (2019), pp. 1–10.

[11] Arnaud Casteigts et al. "Time-varying graphs and dynamic networks". In: *International Journal of Parallel, Emergent and Distributed Systems* 27.5 (2012), pp. 387–408.

[12] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. 92. American Mathematical Soc., 1997.

[13] Reuven Cohen, Shlomo Havlin, and Daniel Ben-Avraham. "Efficient immunization strategies for computer networks and populations". In: *Physical review letters* 91.24 (2003), p. 247901.

[14] Michele Coscia. *The Atlas for the Aspiring Network Scientist*. 2021. arXiv: `2101.00863 [cs.CY]`.

[15] Vinh Loc Dao, Cécile Bothorel, and Philippe Lenca. "Community structure: A comparative evaluation of community detection methods". In: *Network Science* 8.1 (2020), pp. 1–41.

[16] Sara Y Del Valle et al. "Mixing patterns between age groups in social networks". In: *Social Networks* 29.4 (2007), pp. 539–554.

[17] Paul Erdős and Alfréd Rényi. "On the evolution of random graphs". In: *Publ. Math. Inst. Hung. Acad. Sci* 5.1 (1960), pp. 17–60.

[18] Paul Expert et al. "Uncovering space-independent communities in spatial networks". In: *Proceedings of the National Academy of Sciences* 108.19 (2011), pp. 7663–7668.

[19] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. "On power-law relationships of the internet topology". In: *ACM SIGCOMM computer communication review* 29.4 (1999), pp. 251–262.

[20] Santo Fortunato. "Community detection in graphs". In: *Physics reports* 486.3-5 (2010), pp. 75–174.

[21] Santo Fortunato and Marc Barthelemy. "Resolution limit in community detection". In: *Proceedings of the national academy of sciences* 104.1 (2007), pp. 36–41.

[22] Santo Fortunato and Darko Hric. "Community detection in networks: A user guide". In: *Physics reports* 659 (2016), pp. 1–44.

[23] Bailey K Fosdick et al. "Configuring random graph models with fixed degree sequences". In: *Siam Review* 60.2 (2018), pp. 315–355.

[24] Thorben Funke and Till Becker. "Stochastic block models: A comparison of variants and inference methods". In: *PloS one* 14.4 (2019), e0215296.

[25] Laetitia Gauvin et al. "Randomized reference models for temporal networks". In: *SIAM Review* 64.4 (Nov. 2022). DOI: `10.1137/19M1242252`. URL: `https://hal.archives-ouvertes.fr/hal-01817633`.

[26] Michelle Girvan and Mark EJ Newman. "Community structure in social and biological networks". In: *Proceedings of the national academy of sciences* 99.12 (2002), pp. 7821–7826.

[27] Michel L Goldstein, Steven A Morris, and Gary G Yen. "Problems with fitting to the power-law distribution". In: *The European Physical Journal B-Condensed Matter and Complex Systems* 41.2 (2004), pp. 255–258.

[28] Roger Guimera and Luıs A Nunes Amaral. "Modeling the world-wide airport network". In: *The European Physical Journal B* 38.2 (2004), pp. 381–385.

[29] David K Hammond, Pierre Vandergheynst, and Rémi Gribonval. "Wavelets on graphs via spectral graph theory". In: *Applied and Computational Harmonic Analysis* 30.2 (2011), pp. 129–150.

[30] Petter Holme and Beom Jun Kim. "Growing scale-free networks with tunable clustering". In: *Physical review E* 65.2 (2002), p. 026107.

[31] Petter Holme and Jari Saramäki. *Temporal Network Theory*. Springer, 2019.

[32] Petter Holme and Jari Saramäki. "Temporal networks". In: *Physics reports* 519.3 (2012), pp. 97–125.

[33] Mathieu Jacomy. "Epistemic clashes in network science: Mapping the tensions between idiographic and nomothetic subcultures". In: *Big Data & Society* 7.2 (2020), p. 2053951720949577.

[34] Inderjit S Jutla, Lucas GS Jeub, and Peter J Mucha. "A generalized Louvain method for community detection implemented in MATLAB". In: *URL http://netwiki. amath. unc. edu/GenLouvain* (2011).

[35] Brian Karrer and Mark EJ Newman. "Stochastic blockmodels and community structure in networks". In: *Physical review E* 83.1 (2011), p. 016107.

[36] Leo Katz. "A new status index derived from sociometric analysis". In: *Psychometrika* 18.1 (1953), pp. 39–43.

[37] Mikko Kivelä et al. "Mapping temporal-network percolation to weighted, static event graphs". In: *Scientific reports* 8.1 (2018), pp. 1–9.

[38] Jon M Kleinberg et al. "The web as a graph: measurements, models, and methods". In: *International Computing and Combinatorics Conference*. Springer. 1999, pp. 1–17.

[39] PJ Lamberson. "Diffusion in networks". In: *The Oxford handbook of the economics of networks. Oxford University Press, Oxford* (2016), pp. 479–503.

[40] Matthieu Latapy, Tiphaine Viard, and Clémence Magnien. "Stream graphs and link streams for the modeling of interactions over time". In: *Social Network Analysis and Mining* 8.1 (2018), p. 61.

[41] Clement Lee and Darren J Wilkinson. "A review of stochastic block models and extensions for graph clustering". In: *Applied Network Science* 4.1 (2019), p. 122.

[42] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. "Graphs over time: densification laws, shrinking diameters and possible explanations". In: *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 2005, pp. 177–187.

[43] Fredrik Liljeros et al. "The web of human sexual contacts". In: *Nature* 411.6840 (2001), pp. 907–908.

[44] Jian Liu. "Fuzzy modularity and fuzzy community structure in networks". In: *The European Physical Journal B* 77.4 (2010), pp. 547–557.

[45] Aaron F McDaid, Derek Greene, and Neil Hurley. "Normalized mutual information to evaluate overlapping community finding algorithms". In: *arXiv preprint arXiv:1110.2515* (2011).

[46] Miller McPherson, Lynn Smith-Lovin, and James M Cook. "Birds of a feather: Homophily in social networks". In: *Annual review of sociology* 27.1 (2001), pp. 415–444.

[47] Filippo Menczer, Santo Fortunato, and Clayton A Davis. *A First Course in Network Science*. Cambridge University Press, 2020.

[48] Maria CV Nascimento and Andre CPLF De Carvalho. "Spectral methods for graph clustering–a survey". In: *European Journal of Operational Research* 211.2 (2011), pp. 221–231.

[49] Mark Newman. *Networks*. Oxford university press, 2018.

[50] Mark EJ Newman. "Mixing patterns in networks". In: *Physical review E* 67.2 (2003), p. 026126.

[51] Mark EJ Newman. "Models of the small world". In: *Journal of Statistical Physics* 101.3-4 (2000), pp. 819–841.

[52] Mark EJ Newman. "The structure of scientific collaboration networks". In: *Proceedings of the national academy of sciences* 98.2 (2001), pp. 404–409.

[53] Bogdan Nica. "A brief introduction to spectral graph theory". In: *arXiv preprint arXiv:1609.08072* (2016).

[54] Vincenzo Nicosia et al. "Extending the definition of modularity to directed graphs with overlapping communities". In: *Journal of Statistical Mechanics: Theory and Experiment* 2009.03 (2009), P03024.

[55] Rogier Noldus and Piet Van Mieghem. "Assortativity in complex networks". In: *Journal of Complex Networks* 3.4 (2015), pp. 507–542.

[56] Cameron Nowzari, Victor M Preciado, and George J Pappas. "Analysis and control of epidemics: A survey of spreading processes on complex networks". In: *IEEE Control Systems Magazine* 36.1 (2016), pp. 26–46.

[57] Peter Orbanz and Daniel M Roy. "Bayesian models of graphs, arrays and other exchangeable random structures". In: *IEEE transactions on pattern analysis and machine intelligence* 37.2 (2014), pp. 437–461.

[58] Samraat Pawar. "Why are plant-pollinator networks nested?" In: *science* 345.6195 (2014), pp. 383–383.

[59] Leto Peel, Jean-Charles Delvenne, and Renaud Lambiotte. "Multiscale mixing patterns in networks". In: *Proceedings of the National Academy of Sciences* 115.16 (2018), pp. 4057–4062.

[60] Leto Peel, Daniel B Larremore, and Aaron Clauset. "The ground truth about metadata and community detection in networks". In: *Science advances* 3.5 (2017), e1602548.

[61] Tiago P Peixoto. "Bayesian stochastic blockmodeling". In: *Advances in network clustering and blockmodeling* (2019), pp. 289–332.

[62] Jörg Reichardt and Stefan Bornholdt. "Statistical mechanics of community detection". In: *Physical review E* 74.1 (2006), p. 016110.

[63] Garry Robins et al. "An introduction to exponential random graph (p*) models for social networks". In: *Social networks* 29.2 (2007), pp. 173–191.

[64] Francisco Aparecido Rodrigues. "Network centrality: an introduction". In: *A mathematical modeling approach from nonlinear dynamics to complex systems*. Springer, 2019, pp. 177–196.

[65] Giulio Rossetti and Rémy Cazabet. "Community discovery in dynamic networks: a survey". In: *ACM Computing Surveys (CSUR)* 51.2 (2018), pp. 1–37.

[66] Giulio Rossetti, Salvatore Citraro, and Letizia Milli. "Conformity: A path-aware homophily measure for node-attributed networks". In: *IEEE Intelligent Systems* 36.1 (2021), pp. 25–34.

[67] Martin Rosvall and Carl T Bergstrom. "Maps of random walks on complex networks reveal community structure". In: *Proceedings of the National Academy of Sciences* 105.4 (2008), pp. 1118–1123.

[68] Martin Rosvall, Jean-Charles Delvenne, et al. "Different approaches to community detection". In: *Advances in network clustering and blockmodeling* (2019), pp. 105–119.

[69] Daniel Spielman. "Spectral graph theory". In: *Combinatorial scientific computing*. 18. Citeseer, 2012.

[70] Clara Stegehuis, Remco Van Der Hofstad, and Johan SH Van Leeuwaarden. "Epidemic spreading on complex networks with community structures". In: *Scientific reports* 6.1 (2016), pp. 1–7.

[71] Vincent A Traag, Ludo Waltman, and Nees Jan van Eck. "From Louvain to Leiden: guaranteeing well-connected communities". In: *Scientific reports* 9.1 (2019), pp. 1–12.

[72] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world'networks". In: *nature* 393.6684 (1998), pp. 440–442.

[73] Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. "Overlapping community detection in networks: The state-of-the-art and comparative study". In: *Acm computing surveys (csur)* 45.4 (2013), pp. 1–35.

[74] Dmitry Zinoviev. *Complex network analysis in Python: Recognize-construct-visualize-analyze-interpret*. Pragmatic Bookshelf, 2018.