# Chapter 1

# Methodology

## 1.1 Complex networks

Many real systems are composed of a large number of elements interacting with each other. Due to interactions, without any central force, the system exhibits the emergence of collective behaviour on the macro level. Such a system is called a Complex System and its properties can not be predicted from the behaviour of the one individual. An example of a complex system is the human brain. The structure of the brain network and its properties are fundamental for brain functioning, while an emergent phenomenon is a human intelligence. In societies, people's interactions lead to civilization, economy, formation of social groups. Also, the animal populations show different levels of organization that emerge from the individual's interactions [1].

The research in complex systems focuses on the structure of the interactions between units. Knowing how branches of the system are connected, we can determine the emergence of the collective behaviour of the system. For the brain network, we can construct representation with neurons and synapses, representing the brain connectivity. Neurons in the same brain area are closely connected [2]. Similarly, we can define communication between people. The structure of these interactions gives us insights, for example, how information propagates through the system. The presence of people with many connections can lead to faster information flow.

Despite the differences between complex systems, they can be studied using complex networks; with sets of nodes (vertices) and links (edges). Elements in the system are nodes, while interactions between them are given as edges. This approximation allows us to treat equally social (graph of actors), biological (network of proteins) or even technological systems (internet, traffic) [3, 4]. In recent years, complex network theory has application in different fields, and the availability of big data incurs its development.

The complex network theory originates from the graph theory in mathematics. The first mathematical problem solved using graph theory was $Konigsberg$ problem of seven bridges. The city $Konigsberg$ had seven bridges connecting the city's parts across the river and the island in the middle. The question was, is it possible to find a walk that crosses all seven bridges only once. Representing the problem as a graph, as in figure 1.1, Euler managed to simplify the problem; the parts of the land are represented as nodes while bridges between them are links. Crossing each bridge only once is possible if each part of the land has an even number of connections. By this it is possible to enter one part of the land from one bridge and leave it by the other. As each node has odd number of connections, in this case it is not possible, see Fig. 1.1.

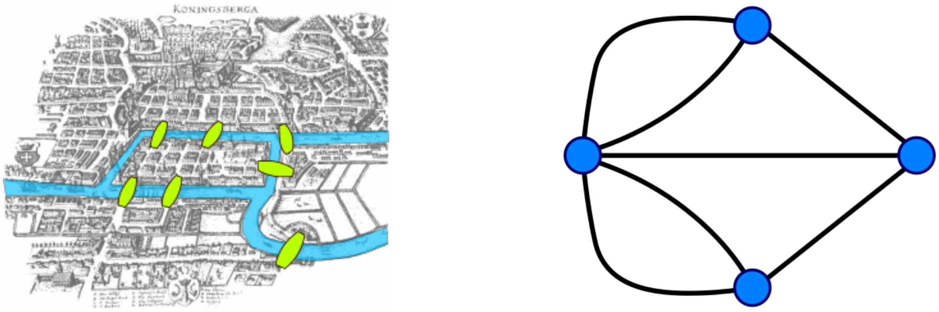

Figure 1.1: The Kronigsber problem of seven bridges.

## 1.2 Types of networks

The graph or network $G$ is defined as $G = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V} = \{v_1, v_2, ...v_N\}$ is a set of $N$ nodes (vertices), and $\boldsymbol{E} = \{e_1, ..e_L\}$ is a set of $L$ edges (links). The edge is pair of nodes $e = (v_i, v_j)$, such that $\{v_i, v_j\} \in \boldsymbol{V}$. The most basic network representation considers **unweighted and undirected** structure. The edges are unweighted, meaning that all interactions in the network are equally important. Because network is un-directed, edges are symmetric, such that $(v_i, v_j)$ implies $(v_j, v_i)$. In **directed** networks this simmetry is broken. The interaction between two nodes $v_i$ and $v_j$, can be only in one direction. A typical example is World Wide Web, where webpages are nodes and hyperlinks are directed edges. In biological networks, gene regulation and neural activation can be described as directed network. The first column a) in Figure 1.2 shows the graphical representation of two networks with equal number of nodes; the first one is underected and the second one is directed.

Even though, graphical representation can be useful for describing the network structure, mathematical representation allow us to characterize the statistical properties of the networks. The graph $G$, with $N$ nodes could be represented with **adjacency matrix** $|A| = N \times N$ [1]. The elements of the matrix are positive if there is connection between two nodes $v_i$ and $v_j$.

$$A_{ij} = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & (v_i, v_j) \notin E \end{cases} \tag{1.1}$$

The column b) on Figure 1.2 shows adjacency matrix representation of given graphs. By convention diagonal elements $A_{ii} = 0$, as self-loops are not allowed. For undirected network adjacency matrix is symmetric $A_{i,j} = A_{ji}$, but in the case of directed network matrix is not symmetric, as edges are drawn in one direction only.

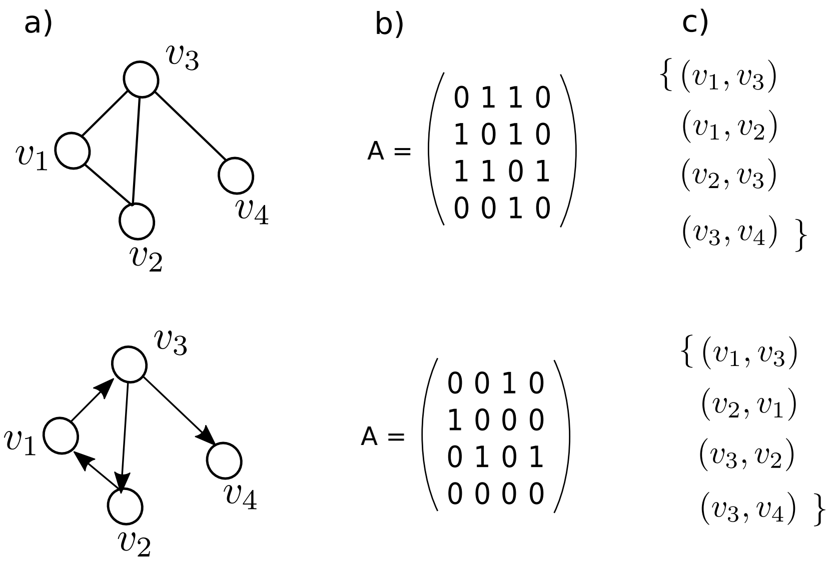

Figure 1.2: a) Graph representation of undirected (top panel) and directed (bottom panel) network. The same networks are represented with adjacency matrices column b), and edge list representation in column c).

The number of edges and nodes are dependent variables. Considering that each node can make $N-1$ connections, the maximum number of the edges in the network is $L_{max} = N(N-1)/2$, as each edge is counted twice. For directed network it is possible to draw $L_{max} = N(N-1)$ edges [5]. When it comes to large networks, they are sparse, meaning that the number of links is $L << L_{max}$. As consequence, the adjacency matrix is also sparse structure (has many zeros) that takes large portion of computer memory [6]. It is common to represent the graph as edge list. In this case, illustrated on Figure 1.2, column c), graph is described with the list of links that are in the graph, $G = \{\{v_i, v_j\}\}$. Still with this representation we are not able to distinguish between directed and undirected graph structures, so in the computational algorithm should be specified if the edges are considered symetric or not.

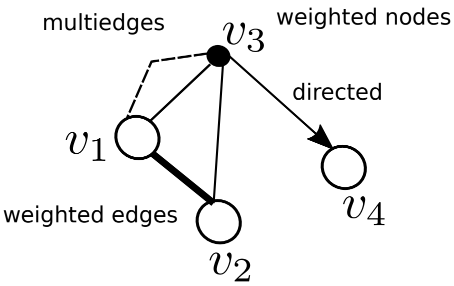

Figure 1.3: The complex networks may represent different characteristics of the system. The edges can be directed, weighted or multiply. Also nodes can be assigned with different weights or any relevant feature.

To create the more realistic models, sometimes is essential to include the specific properties of the system in the network representation. For example, to emphasis the frequent interactions between nodes, edges can be assigned with different values, such networks are **weighted**. They can be described with adjacency matrix, whose elements can take any real number $A_{ij} = w_{ij}$ and $w_{ij} > 0$. In general edges may be associated with any categorical variable. Similarly additional properties can be added to nodes, or even to the whole network structure. To include the **temporal** component in the network, edges are characterized with the time when the interaction between nodes happen. Finally, if two nodes interact in different ways, the **multigraph** is appropriate configuration where multiply edges are allowed. The graphical representation of discussed network representations is given on the Figure 1.3.

A **bipartite network** consists of two types nodes. The nodes in the same partition are not connected, while links exist only between partitions. For many real systems, a bipartite graph is a natural representation[6, 2]. For example, the bipartite network of people and groups has two distinct node partitions while links indicate the memberships. Another example is a system of customers and products. The link between user and item is created when the user buys an item. The bipartite networks find their application in the algorithms for recommended systems, whose goal is to recommend items that may interest the user. Actually, to find the most probable missing links in the network.

In a bipartite network, nodes in one partition are not connected. Still, we can analyse a single node type if we project the bipartite network on one partition. The primary assumption is that two nodes in one partition could be connected if they point to the same node in another partition. Consider the network of movies and actors. The one mode projection of movies is an undirected network whose links indicate that two movies share the same actors. On the other hand, another projection is a network of actors. The links exist if two actors appear in the same movie [7, 6].

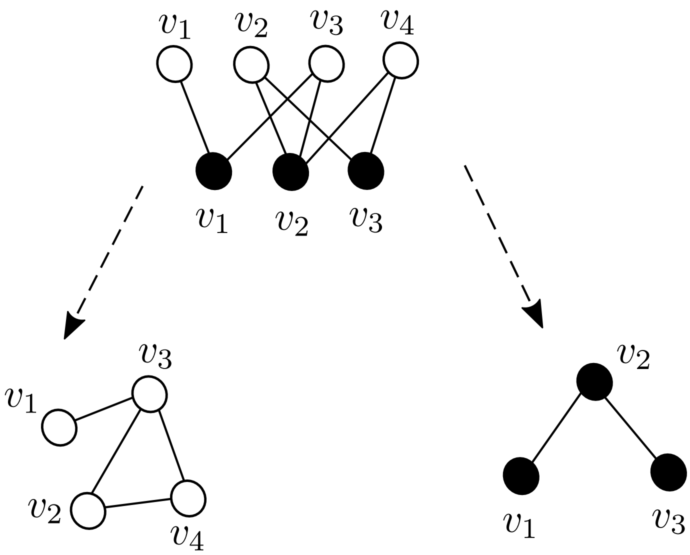

Figure 1.4: Bipartite graph and two partition projections.

We should be aware that important information is lost when creating a one-mode projection. First of all, without having weighted edges in the network of actors, it is impossible to have information on how many movies two actors appear in. From the one-mode projection, we can not reconstruct the original network. Moreover, two different bipartite networks may have the same projected networks. The important consequence of the network projection is the creation of cliques; subgraphs where all nodes are connected.
In general, it is possible to define the k-bipartite network. The same rules apply as before. There are $k$ distinct node partitions, while the edges exist only between different types of nodes.

**Temporal networks.** Studying the real systems as static networks can give us a lot of insight into the system properties. Still, real systems are not static; they evolve not only in the number of elements but also in the number of interactions between them. Some interactions in the system may repeat in different intervals and could be described with complex activity patterns. Including time dimension in the network representation allows us to study the properties of the system closely. The temporal information may matter a lot [8]. For example if interaction between nodes $(v_1, v_2)$ happened before in time than $(v_2, v_3)$, then nodes $v_1, v_3$ would not be connected, as it is the case in the static network.

The temporal network is a collection of timestamped edges. Each edge is defined as $(v_i, v_j, t, \Delta t)$, where $v_i$ and $v_j$, are nodes $t$ is time when interaction happen, and $\Delta t$ is event duration [9]. The duration of the events may vary, as in the phone-call network. Also, for many systems, the time resolution of event duration is too small. For example, this parameter may be neglected when people interact on social platforms or email each other because the event time is too short, it scales in seconds.

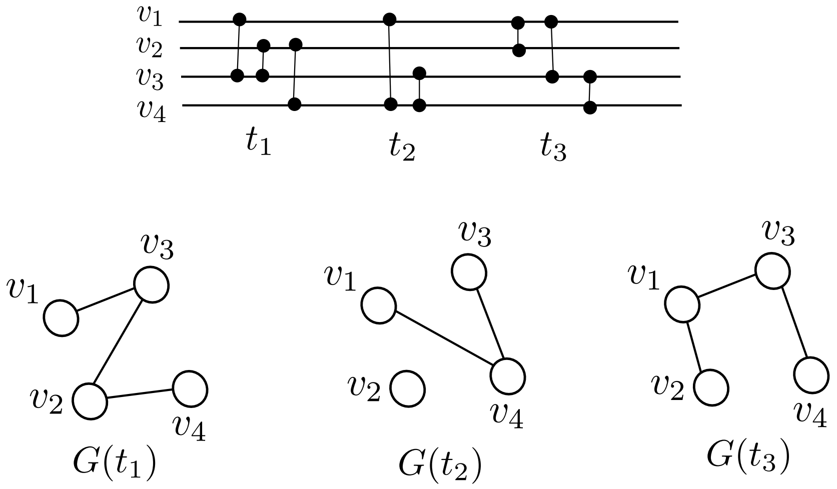

Figure 1.5: Temporal network.

The temporal network can be represented as sequence of static networks that evolve in time, $G = \{G(t_1), G(t_2), ..., G(t_{max})\}$. At each time step, we can create the network and analyze the macroscopic properties of the given network snapshot. With this, we can end up with graph snapshots with many disconnected components or empty graphs for some points [10]. Sometimes, a much better approach is to aggregate the links that over time-windows. Here, we need to specify the time window length $w$. Interactions in the time interval $0 \leq t < w$ enter the first snapshot. The next snapshot takes edges $w \leq t < 2w$, and so on. The time windows are not overlapping, but generally, it is possible to slide the time window for different periods $1 \leq \delta t \geq w$. The downside of this method is that we can not recover original data points. The larger the time window is, the more information is lost. If the time window is set to $w = t_{max}$, there is only one snapshot, and the temporal data are no more available [11, 12].

**Multilayer networks** were introduced for studying systems in which different types of interaction exist. This formalism allows one to investigate diverse network systems and to combine different types of data into one model [13]. In a multilayer or multiplex network, all nodes are present in each layer, but their interactions among layers differ. Two nodes may be connected in one layer but not in the other. Different online social systems may be an example of a multiplex network when users are connected on one platform but not on the other [14]. Or the airline transportation network, where each layer represents the flights of different airline companies [15].

## 1.3 The structure of complex networks

### 1.3.1 Degree distribution

The simplest network measure is **node degree**, $k$. The degree of node $i$ gives the number of nodes attached to node $i$, $k_i = \sum_j A_{ij}$. The density of the network is average degree divided by $N - 1$, where $N$ is number of nodes. It is relative fraction of nodes in the network.

In the case of regular networks, such as grids, each node has an equal degree, meaning that nodes in the network have similar roles. In the general case, the networks have more complicated structure. If degree sequence is skewed, we are able to identify nodes with high degree, hubs. Removing hubs may partition a connected network into several components. Finally, if we are able to test isomorphism between two graphs, the starting point would be to compare their degree sequences are the same. If they are not same, then graphs can not be isomorphic.

To calculate the degree distribution we can consider the fraction of $k$ degree nodes $N_k$, $p(k) = N_k/N$. It is the probability , $P(k)$, that randomly chosen node has degree $k$. Similarly, we can order nodes according to their degree and plot the node degree.

If the nodes of the graph are statistically independent, the degree distribution completely determines the properties of a network. Here we summarize the forms of degree distributions that are mostly found in the complex network theory:

- The Poisson distribution. The degree distribution in random network, where all nodes have the same connecting probability, follows Poisson distribution $P(k) = \frac{(Np)^k e^{-Np}}{k!}$, where $k$ is the mean degree distribution.
- Exponential distribution. $P(k) = e^{-k/k}$. This is degree distribution of the growing random graph. Even for infinite networks all moments of distributions are finite, and have natural scale of the order of average degree.
- In many real networks degree distribution follows a power law. $P(k) = k^{-\gamma}$, where $\gamma$ is exponent of the distribution. In this distribution there is no natural scale, so they are called scale-free networks. In infinite networks all higher moments diverge. If the average degree of scale-free networks is finite, than $\gamma$ exponent should be $\gamma > 2$. Therefore, real networks have a scale-free structure with the emergence of the hubs [7].

When plotting the degree distribution, it is common to use scaling of the axis. As many nodes have low degree, like for power-law or exponential distribution it is more useful to use logarithmic scale. Now it is more easily notices that data-points follow straight line, meaning that degree distribution is some kind of exponential function.

### 1.3.2 Degree correlations

Correlation is defined through a correlation coefficient r. If x and y are two stochastic variables, for which we have a series of observation pairs $(x_1, y_1), (x_2, y_2), ...(x_n, y_n)$. The correlation coefficient $r(x, y)$ between $x$ and $y$ is defined as:

$$r(x,y) = \frac{\frac{1}{n}\sum_{i=1}^{n}((x_i - \bar{x})(y_i - \bar{y}))}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1.2}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$, is the average over variable $x$.

Taking the definition of correlation coefficient we can define it for vertex degrees. For simple graph G with vertex set $V(G) = \{v_1, ..v_n\}$, $\boldsymbol{A}[i,j] = 1$ if there is a link between nodes $v_i$ and $v_j$. If G is a simple graph with adjacency matrix $\boldsymbol{A}$ and degree sequence $\boldsymbol{d} = [d_1, ..., d_n]$

$$r_{deg}(G) = \frac{\sum_{i=1}^{n}\sum_{i=1+1}^{n}((d_i - \bar{d})(d_i - \bar{d})\boldsymbol{A}[i,j])}{\sum_{i=1}^{n}(d_i - \bar{d})^2} \tag{1.3}$$

Using adjacency matrix, allow us to calculate the correlations between neighboring nodes. If two nodes are not connected $A[i,j] = 0$, the degree correlation between them does not have contribution to the $r$.

The **degree-degree correlations** in the network are measured by **assortativity**. If correlations are positive, networks are assortative; there is a tendency that connections exist between similar degree nodes. The negative correlations indicate that large degree nodes have preference to connect nodes with small degree; dissasortative networks. The average first neighbor degree $k_{nn}$ can be calculated as $k_{nn} = \sum_{k'} k' P(k'|k)$. The $P$ is conditional probability that an edge of degree $k$ points to node with degree $k$. The norm is $\sum_{k'} P(k'|k) = 1$, and detailed balance conditions [1], $kP(k'|k)P(k) = k'P(k|k')P(k')$ [1]. If the node degrees are uncorrelated, $k_{nn}$ does not depend on the degree, otherwise increasing/decreasing function indicates on positive/negative correlations in the network.

The Newman defined the assortativity index $r$ in slightly different way:

$$r = \sum_{kl} kl(e_{kl} - q_l q_k)/\sigma_q^2 \tag{1.4}$$

where $e_{kl}$ is the probability that randomly selected link connect nodes with degrees $k$ and $l$, $q_k$ is probability that randomly choosen node is connected to node $k$ and equals $q_k = kp_k/\langle k \rangle$, while $\sigma_q$ is variance of the distribution $q_k$.

### 1.3.3 Clustering coefficient

The **clustering coefficient** is a measure describing the neighbourhood's structure. In networks exist tendency to form triangles or clusters. This is common in friendship networks where two friends of one person have a high probability of being friends. The clustering can be measured by computing the number of links between neighbours of one node,

$$c_i = 2e_i/(k_i(k_i - 1)) \tag{1.5}$$

Averaging it over all network nodes, we can calculate the mean clustering coefficient. It ranges from $\langle c \rangle = 0$ where connections between neighbouring nodes do not exist, network has the structure of three. On the other hand, $\langle c \rangle = 1$ indicates a fully connected network.

Newman proposed the alternative definition for the clustering coefficient based on the number of triples and triangles in a graph. A triangle at node $v$ is complete subgraph with 3 nodes, including $v$. A triple on the node v is a subgraph of exactly three nodes and two edges, where v is incident with two edges. The network transitivity is defined as the ratio of number of triangles in the network over the number of triples. The network transitivity is seen as global clustering, as it considers the whole network.

### 1.3.4 Network paths

In the network structure, the interacting nodes are directly connected with the edge. In this representation we can say that distance between them is $d_{v_i,v_j} = 1$. Distance defined like this does not have any physical meaning. Its purpose is to describe how the position of nodes in the network structure influences the other distant nodes.

The **path** between two nodes, $v_i$ and $v_j$ is a sequence of edges $\{(v_1, v_2), (v_2, v_3), ...(v_k, v_{k+1})$ $, ...(v_{n-1}, v_n)\}$, where $v_1 = v_i$, $v_n = v_j$. In the path, the nodes are distinct. Otherwise, the sequence is called a **walk**, where each node can be visited many times. Also, it is possible to define a **cycle**, a path that starts and ends on the same node while other nodes in the cycle are distinct. The length of the path, walk or cycle is the number of links in the sequence. Using the adjacency matrix we can easily calculate the number of walks between two nodes. The $A^2$ gives us walks of length 2, the $A^3$, number of walks of length 3, and so on.

The network is connected if it is possible to define the path between every two nodes in the network. When it is not the case, the network is disconnected into two or more connected components. Note that the component can be an isolated node. Also, in directed networks may happen that node $v_i$ is reachable from node $v_j$, but if we start from $v_j$ we can not find the path to the $v_i$. Such a graph is connected but is called a weakly connected component.

We can find different paths between two nodes in the network, but the most important one is the **shortest path**. The distance between two nodes $d(v_i, v_j)$ is defined as the shortest path length between two nodes. In the case of weighted networks, it is the path with minimal weight, and the length of such path does not have to be minimal. Distances on the network can give us insight into how similar networks are and indicate the node's relative importance in the network.

The **radius** is the minimum overall eccentricity values, while the **diameter** defines the largest distance between nodes in the network. These definitions apply to directed and undirected graphs.

If G is a connected graph with vertex set V and $\bar{d}(u)$ is the average length of the shortest paths from node u, to any other node v in network G.

$$\bar{d}(u) = \frac{1}{|V| - 1} \sum_{v \in V, v \notin u} d(u, v) \tag{1.6}$$

From there, the **average path length** is mean value over $\bar{d}(u)$.

$$\bar{d}(G) = \frac{1}{|V|} \sum_{u \in V} \bar{d}(u) \tag{1.7}$$

while the **characteristic path** length of G is median over all $\bar{d}(u)$.

### 1.3.5 D-measure

For each node $i$ we can define the distribution of the shortest paths between node $i$ and all others nodes in the network, $P_i = \{p_i(j)\}$, where $p_i(j)$ is percent of nodes at distance $j$ from node $i$. The connectivity patterns can efficiently describe difference between two networks. To specify how much $G$ and $G^{'}$ are similar we use D-measure [16]

$$D(G, G^{'}) = \omega \left| \sqrt{\frac{J(P_1, ..P_N)}{log(d)}} - \sqrt{\frac{J(P_1^{'}, ..P_N^{'})}{log(d^{'})}} \right| + (1 - \omega) \sqrt{\frac{J(\mu_G, \mu_{G^{'}})}{log2}} \tag{1.8}$$

D-measure calculates Jensen-Shannon divergence between $N$ shortest path distributions,

$$J(P_1, .., P_N)) = \sum_{i,j} p_i(j) log(\frac{p_i(j)}{\mu_j}) \tag{1.9}$$

where $\mu_j = (\sum_{i=1}^{N} p_i(j))/N$ is mean shortest path distribution.

The first term in equation 1.8 compares local differences between two networks, and Jensen-Shannon divergence between $N$ shortest path distributions $J(P_1, ..., P_N)$ is normed with network diameter $d(G)$. The second part determines global differences, computing $J(\mu_G, \mu_{G'})$ between mean shortest path distributions. The D-measure ranges from $0$ to $1$. The lower D-measure is, networks are more similar and for D-measure $D = 0$, structures are isomorphic.

### 1.3.6 Community structure

Nodes can be organized into groups, called communities. Identifying these hidden blocks can lead to interesting insights into the network. The communities are expected in social networks, as people tend to organize into different groups.

However, the community detection problem does not give a precise definition of what a community is. A common definition of a community is that it is densely connected subgraph [17], [18]. In community detection the number of communities is not predefined. The number of possible communities in the network could be large number and we can not analyse all combinations, so we need algorithms to help us to identify potential communities in the network.

**Modularity**. Comparing the link density of the community by the link density obtained for the same group of nodes randomly connected we could conclude if the community corresponds to the dense subgraph or the structure is created completely random. The modularity is function that measures the randomness of the each partition. With modularity we can compare the communities and decide which one is better.

For the network with $N$ nodes and $L$ links that partitions into $n_c$ communities. Each community has $N_c$ nodes and $L_c$ links. If number of links is larger than the expected number of links between $N_c$ nodes given in the expected node sequence than these nodes may form the community. We calculate the difference between real network connectivity $A_{ij}$ and the expected number of links between nodes if the network is randomly connected, $p_{ij}$. The $p_{ij}$ can be obtained by randomizing the original network, but keeping the expected degree of each node unchanged, so $p_{ij} = \frac{k_i k_j}{2L}$.

$$M_c = \frac{1}{2L} \sum (A_{ij} - p_{ij}) \tag{1.10}$$

If modularity is positive, the selected nodes may be community as their connectivity is far from random. If $M_c$ is zero, then the connectivity between nodes is random, and if $M_c$ is negative the nodes do not form the community.

The same idea can be generalized to the whole network: The modularity of network partitioned into $n_c$ communities is then defined as:

$$M = \sum_{c=1}^{n} [\frac{L_c}{L} - (\frac{k_c}{2L})^2] \tag{1.11}$$

The higher modularity indicates that nodes are partitioned in better communities. When we put all nodes into only one community $M = 0$, otherwise if each node is community itself $L_c = 0$ and the sum is negative.

Maximum network modularity indicates the best partions. As there are too many partitions, it is not possible to construct all possible partitions and calculate their modularity. For that we need algorithm that could identify the the best partition.

The first algorithm that was proposed for modularity optimization was greedy algorithm. First it assign each node to community, and start with N communities. Then, we try to merge each pair of communities and calculate the moddularity difference $\Delta M$. Then indentify the community pair for which the difference is largest and merge those two communities. This is repeated untill all nodes merge into single community. The best partition is one with largest $M$.

**Louvain algorithm** is optimization algorithm with better scalability than gready algorithm, so it can operate on very large networks. Initially each node is assigned to different community, and similar as before we calculate the difference in the modularity if we move the community of one of its neighbours. Then we move the node i to the community such that modularity becomes larger. This is applyed to all nodes untill no further improvement could be made. In the second step we create weighter network whose nodes are communities identified during first step. The weight of the link between communities is the sum of the weights between the links in the communities, and the number of links inside the community is given as weighted self-loop. Then the first and secound steps are repeated, until there is no more change in the modularity, otherwise until we find the maximum, optimal modularity.

**Core-periphery** structure describes a network whose nodes are divided into two community, densely connected core and less connected periphery. If we consider the average probabilities of edges within each group as $p_{11}$ and $p_{22}$, and between groups $p_{12}$, instead of traditionaly assortative or dissasortative structure we can define core-periphery structure $p_{11} > p_{12} > p_{22}$. In the principle core-periphery structure does not have to be limited to only two groups, and we can define layered, onion, structure. The network can have more cores, that are not directly connected to each other.

The simple method for finding core-periphery structure is to assume that nodes in core have higher degree in the core than in the periphery. Another simple method is to construct k-cores. K core is group of nodes that each has connection to at least k other members of the group. K-cores form a nested set, and become denser with higher k. The core-periphery structure can be detected optimizing the measure similar to modularity, as defined by Borgatti and Everett. Their goal is to find the division that minimizes the number of edges in the periphery. So they define the score function that is equal to number of edges in the periphery minus the expected number of such edges placed at random. $\rho = \frac{1}{2}\sum_{ij}(A_{ij} - p)g_i g_j$.

The another way to detect core-periphery structure is to use the inference method based on fits to a stochastic block model. In this method we fit observed network to a block model with two groups, such that edge-probabilities have form $p_{11} > p_{12} > p_{22}$.

**Stochastic Block Model** is model where each node, in given network G, belongs to one of $B$ blocks. Vector $\theta_i = r$ indicates that node $i$ is in block $r$, while SBM matrix $\{p\}_{BxB}$, specify the probability $p_{rs}$ that nodes from group $r$ are connected to nodes in group $s$. The SBM model is looking for the most probable model that can reproduce a given network G. Probability of having model parameters $\theta, p$ given network $G$ is proportional to likelihood of generating network $G$, prior of SBM matrix and prior on block assignments:

$$P(\theta, p|G) = P(G|\theta, p)P(p)P(\theta) \tag{1.12}$$

$$P(G|\theta, p) = \prod_{i<j} p_{r_i s_j}^{A_{ij}} (1 - p_{r_i s_j})^{1-A_{ij}} \tag{1.13}$$

where $A_{ij}$ is number of edges between nodes $i$ and $j$.

Prior on p is modified for core-periphery model such that $P(p) = 3! I_{0<p_{22}<p_{12}<p_{11}<1}$, while prior on $\theta$ consists of three parts: probability of having $l$ blocks; given the number of layers probability $P(n|l)$ of having groups of sizes $n_1..n_l$ and the probability $P(\theta|n)$ of having particular assignments of nodes to blocks.

For fitting model in the work [19] authors use Metropolis-within-Gibbs algorithm. The likelihood of SBM model increase with number of blocks and model itself does not define optimal number of communities. Inferring minimum description length (MDL) of the model is one approach to decide which model is more likely.

## 1.4 Network models

### 1.4.1 Random network model

The random graph model was introduced by mathematicians Paul Erdős and Alfred Rényi in 1959. In this model, connections between nodes are chosen randomly, and every link has the same probability of existing. The graph is characterized only by a number of the nodes $N$ and the linking probability $p$, so Erdős-Rényi graph is written as $G(n, p)$.

The creation of ER random network consists of the following steps:

- we start with $N$ isolated nodes
- between each $N(N-1)/2$ pair of nodes we create link with probability $p$; sampling random number $r \in (0, 1)$, we create link if $r \leq p$

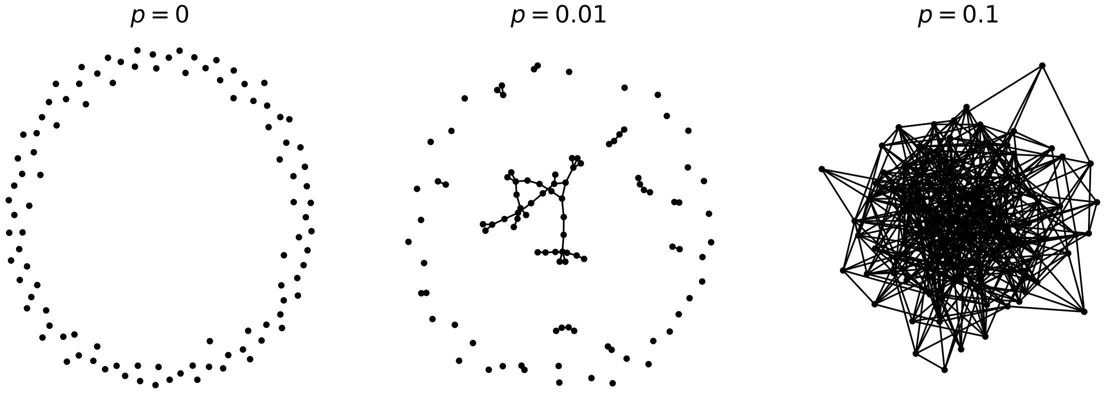

Figure 1.6: ER graph with $N = 100$ nodes and different linking probabilities $p$.

We should note that this process is stochastic. The networks $G(N, p)$ with the same parameters do not need to have the same structure; i.e. they differ in the number of links. Therefore, the single random graph is only one graph from all the possible realizations in the statistical ensemble.

Two simple quantities that could be estimated are the average number of links and the average degree. For complete graph with $N$ nodes, number of edges is $N(N-1)/2$. As the probability of drawing every edge is $p$, the **average number of links** is simply given as

$$\langle L \rangle = \frac{N(N-1)}{2} p \tag{1.14}$$

From there, we conclude that the network's density is equal to probability $p$. The **average degree** is approximated as: $\langle k \rangle = 2\langle L \rangle / N$, leading to:

$$\langle k \rangle = (N-1)p \tag{1.15}$$

The **degree distribution** of ER random graph follows the binomial distribution.

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k} \tag{1.16}$$

The probability that the node has degree $k$ is given with the second term $p^k$, while the probability that other N-1-k links are not created is given with the third part of the equation. Finally, there are $\binom{N-1}{k}$ combinations for one node, to have $k$ links from $N-1$ possible links.

The binomial distribution describes very well small networks. For larger networks, we find that they are sparse and that the average degree is much smaller than a number of nodes $\langle k \rangle << N$. In this limit, binomial distribution becomes the Poisson, which now depends only on one parameter $\langle k \rangle$

$$p(k) = \frac{1}{k!} e^{-\langle k \rangle} \langle k \rangle^k \tag{1.17}$$

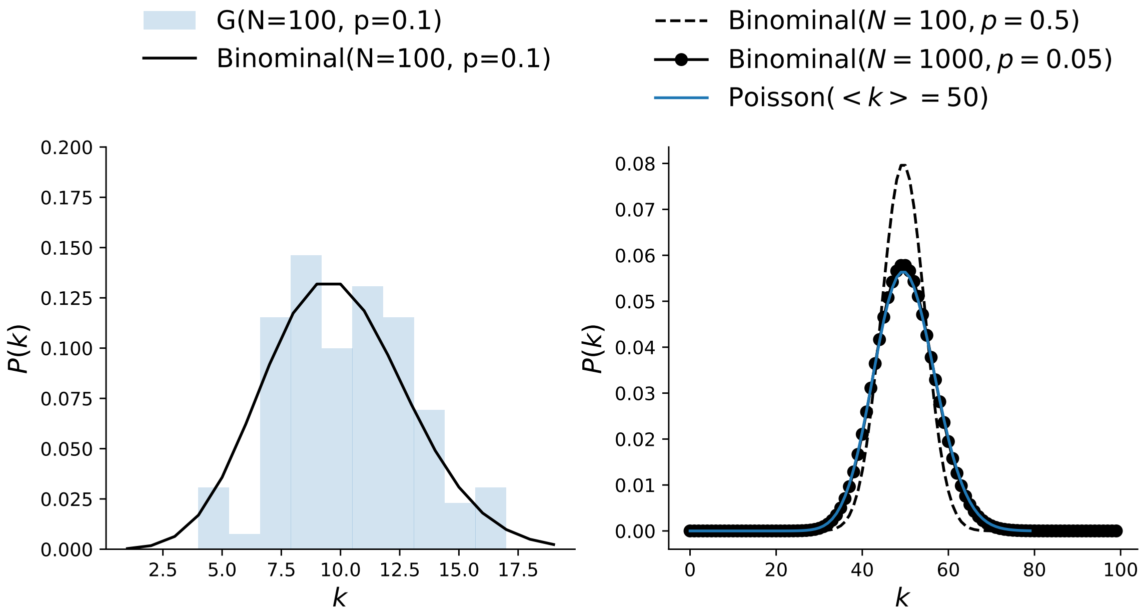

Figure 1.7: Degree distribution of ER graph. Degree distribution of small networks follow binominal. Larger networks are better approximated with Poison distribution, and degree distribution for fixed average degree $< k >$ becomes independent of the network size.

The random graph has a very small **average path length**, it is given as $\langle l \rangle = \frac{lnN}{ln(pN)}$ that is characteristic of many large networks. The clustering coefficient is proportional to linking probability, $\langle C \rangle = p$, so in large random networks, we find a small clustering coefficient, contrary to real-world networks.

The figure 1.6 shows how the network becomes more connected by increasing the linking probability $p$. When $p = 0$, all nodes are disconnected. In the other limit, $p = 1$, the network is fully connected. Between those two probabilities exists critical probability, where the giant component appears. The giant component is a sub-graph, which size is proportional to the network size. In other words, the network does not have disconnected components. Such change in the network is a phase transition in network connectivity and is related to percolation theory.

The phase transition occurs when average degree is $\langle k \rangle = 1$, which gives us: $p_c = \frac{1}{N-1}$, meaning that all nodes have degree larger than 1. When the $\langle k \rangle < 1$, the network is in the sub-critical regime where all components are small. In the critical regime, the size of the giant component is proportional to the $N^{2/3}$. In the supercritical regime, $\langle k \rangle > 1$, the probability of a giant component appearing is 1.

### 1.4.2 Small-world networks

Inspired by the idea that real-world networks are highly clustered and the average distance is small, Watts and Strogatz proposed the "small-world" model. The model starts from the regular lattice, and with rewiring links, the network starts to resemble small-world property. The procedure is the following:

- At the beginning, nodes are placed on the ring lattice, and each node is connected to $k/2$ first neighbours on the left and the right side. Initially, the clustering coefficient is high, $c = 3/4$.
- For each link in the network, with probability $p$, we choose a random node to rewire the link. This makes long-distance nodes connect, decreasing the network's average path length.

The model interpolates between the regular graph when the probability is $p = 0$ and the random graph with $p = 1$ when all links are randomly rewired. Short distances and high clustering are present in the network for the critical probabilities.

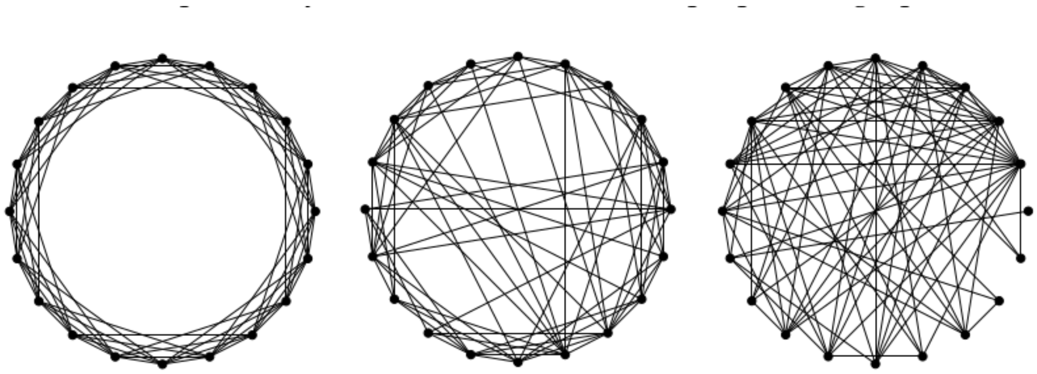

Figure 1.8: Watts and Strogatz graph model creation

Even though the small-world network model lacks the power-law degree distribution found in the real-world networks, it is an important model that motivated the research on random graphs.

### 1.4.3 Barabási-Albert model

The ER random graph model and WS small-world model are static models, where the number of nodes is fixed. It is one of the reasons why they can not fully explain the properties of real systems. The size of real systems does not remain constant; real networks grow. For the network, the growth means that at each time step, new nodes are added to the network. The simplest model that produces the scale-free networks is Barabasi-Albert model.

- The model starts from the small number, $n_0$ randomly connected nodes, with $m_0$ links.
- At each time step, new node with $m$ links joins to the network. New node creates links with the nodes already present in the network, following the linking rules; i this case rules of preferential attachment.

The preferential attachment is important ingredient for generating system with scale-free properties. In the real-system the linking between nodes is not random process, there exists the preference toward specific types of nodes. For example the popular web-pages can easily get more visits or it is common that already popular papers will get more citations. This effect is also called rich-get-richer or preferential attachment.

The simplest formulation of the preferential attachment model is that new nodes tend to connect with high degree nodes. The linking probability $\Pi$ is then proportional to node degree $k$:

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j} \tag{1.18}$$

As at each time step one node arrive, we can estimate the number of nodes at the time step t, $N(t) = n_0 + t$, with links $L(t) = m_0 + mt$.

First we can calculate the evolution of network degree in time.

$$\frac{dk_i}{dt} = m\Pi(k_i) = m\frac{k_i}{\sum_j k_j} = m\frac{k_i}{m_0 + 2mt} \tag{1.19}$$

Note that new node, that arrived at time point $t_i$ has degree $m$, as it links to $m$ old nodes. Solving the equation we get that at $t > t_i$, has degree that grows as square root of time, also it shows that younger nodes easily acquire larger degree.

$$k_i(t) = m\left(\frac{t}{t_i}\right)^{\frac{1}{2}} \tag{1.20}$$

Degree distribution follows power-law, and for large k is aproximated with $P(k) = k^{-\gamma}$, such that $\gamma = 3$. More precisely, the degree distribution has form:

$$P(k) = \frac{2m(m+1)}{k(k+1)(k+2)} \tag{1.21}$$

For large $k$ it is exactly power-law. It is also independent of the time and size of the system, meaning the emergence of stationary scale-free state. Distributions do not depend on the N. If we vary $m$ the slope of distributions is the same, but they are parallel. After re-scaling $p(k)/m^2$, they fall on the same line.

As network grows nodes with larger degree becomes bigger, so we end up with few nodes with many links, called hubs. The **network diameter**, represents the maximum distance in network, $d \sim \frac{lnN}{lnlnN}$. The diameter grows slower than $lnN$, making the distances in BA model smaller than in random graph. The difference is found for large N. Knowing that BA network has hubs, that shorten the path between less connected nodes. Also, if hubs are removed from the network, network easily partition in several components, loosing its properties. The **clustering coefficient** of the BA model follows $C \sim \frac{lnN^2}{N}$. It is different from clustering found in random networks, and BA networks are in general more clustered.

The combination of the growth and preferential attachment linking is crucial for getting scale free networks. For example, eliminating the preferential attachment; in growing network with random linking, degree distribution is stationary, but it follows exponential. In contrast, the absence of growth leads to the non-stationary degree distribution. When number of nodes is fixed, while the network grows only in number of links, such that randomly chosen node $i$ connects to node $j$ according to probability $\Pi$. At the beginning, the degree distribution follows the power-law, same as in BA model. As more links are added to the network, the distribution changes it's shape, first the peak appears, while at the end network becomes complete graph, where all nodes have the same degree.

### 1.4.4 Nonlinear preferential attachment model

In the nonlinear preferential attachment model linking probability also depends on the node degree. The dependence is not linear and has the following a form:

$$\Pi(k_i) = k_i^{\beta} \tag{1.22}$$

The probability that newly added node attaches to node $i$ depends on the existing i-th node degree $k_i$, and the parameter $\beta$. When $\beta = 1$, the model is BA model, where degree distribution follows the power-law. When $\beta = 0$, linking probability becomes uniform; i.e. it corresponds to random network model, and degree distribution is Poisson; there is exponential decay.

For $\beta > 1$, the effects of preferential attachment are increased, leading to emergence of super-hubs. The hub-and-spoke network appear in this regime, where almost all nodes are connected to few high-degree nodes.

On the other hand, if $\beta < 1$, the model is in so called sub-linear preferential attachment regime. The linking probability is not random so degree distribution does not follow Poisson; but also the preference toward high degree nodes is too week for having the pure power-law. Instead degree distribution converge to stretched exponential.

### 1.4.5 Ageing model

To understand how aging can impact the network structure we look into probability dependent on two parameters, nodes degree $k$ and age of node $i$ at the time point $t$ $\tau_i = (t - t_i)$, where $t_i$ is the time when node $i$ is added to the network.

$$\Pi_i(t) \sim k_i \tau_i^{\alpha} \tag{1.23}$$

The parameter $\alpha$ controls the linking probability dependence on the nodes' age if $\alpha = 0$, the ageing of nodes is disregarded.

If $\alpha > 0$ is positive, the older nodes are more likely to create connections. In this regime, the preferential attachment stays present, and the high-degree and older nodes are preferred. For very high $\alpha$, each node is connected to the oldest node in the network. The scale-free properties are present; the power-law exponent $\gamma$ deviates from $\gamma = 3$. It is found that $\gamma$ ranges between 2 and 3.

When $\alpha$ is negative, ageing overcomes the role of preferential attachment, and scale-free properties are lost. For significant negative $\alpha$ network becomes a chain; the youngest nodes are those who get connected.

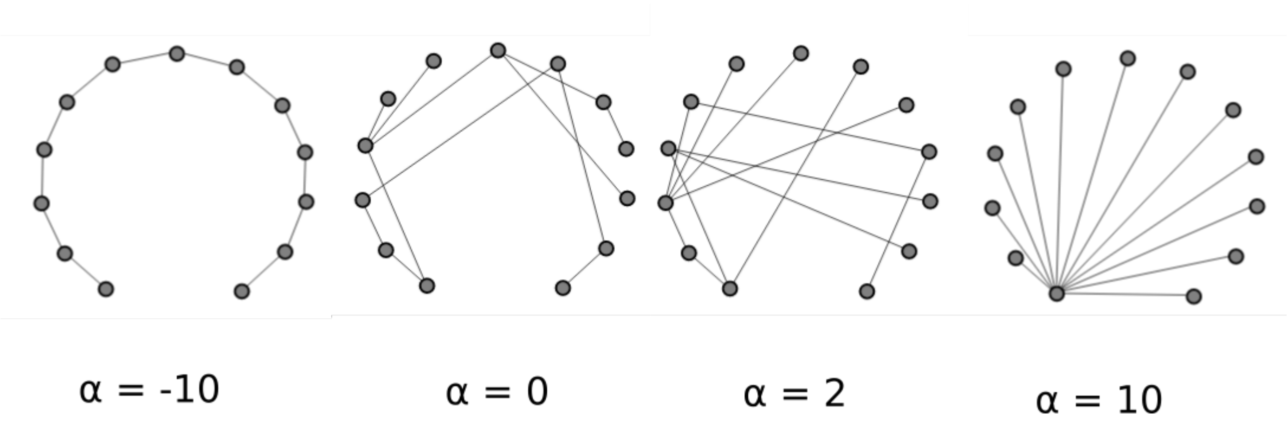

Figure 1.9: Aging model

In the general ageing model, the non-linearity on the node degree is introduced, so this model has two tunable parameters $\alpha$ and $\beta$. The probability that a link is created between the new node and the existing node is defined as

$$\Pi_i(t) \sim k_i(t)^{\beta} \tau_i^{\alpha} \tag{1.24}$$

As before, depending on model parameters network evolves to different structures [20].

- For example if we fix $\beta = 1$ and $\alpha = 0$ generated networks are scale-free; degree distribution is $P(k) \sim k^{-\gamma}$ with $\gamma = 3$.
- In the case of nonlinear preferential attachment $\beta \neq 1$ and $\alpha = 0$ scale-free properties disappear.
- Scale-free property can be produced along the critical line $\beta(\alpha^*)$ in the $\alpha - \beta$ phase diagram, see Figure 1.10.
- For $\alpha > \alpha^*$ networks have **gel-like small world** behavior.
- For $\alpha < \alpha^*$ and near critical line $\beta(\alpha^*)$ degree distribution has **stretched exponential** shape

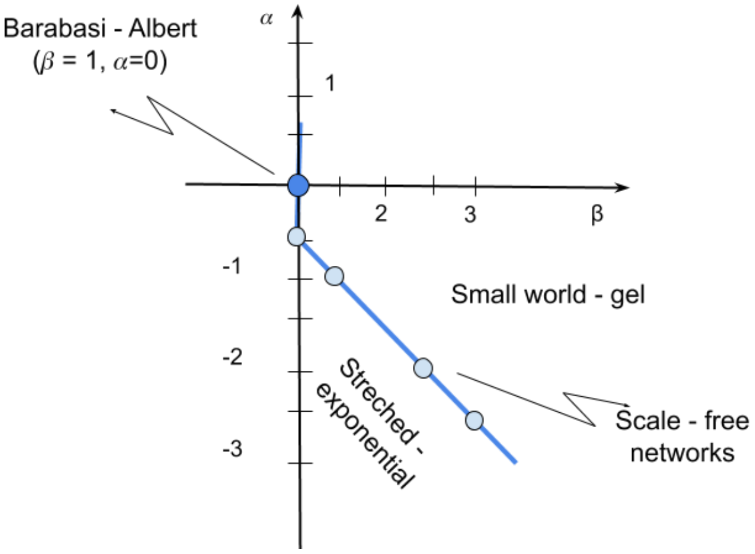

Figure 1.10: Phase diagram of aging network model

### 1.4.6 Stochastic block model

Stochastic block model (SBM) is based on connection probabilities between nodes. It is a generative model which includes existence of communities. Parameters that describe SBM for network G with N nodes are:

- k: number of groups
- group assignment vector, g: $g_i \in \{1, 2..k\}$, gives the group index of node $i$.
- SBM matrix, $p_{k \times k}$, whose elements $p_{ij}$ are the probabilities that edges between groups $g_i$ and $g_j$ exist.

Note that nodes within one group have the same connection probabilities.

SBM can generate and describe different types of network structures. Figure 1.11 [17] shows how the model matrix corresponds to resulting networks with two communities. First, for the assortative network (1.11 a), diagonal elements of the matrix have higher probabilities. This indicates dense connections inside the group, just like in classic community structures. In disassortative structure, (1.11 b), more connections exist between two partitions than inside them, i.e. off-diagonal elements have higher probabilities. Bipartite networks can be represented like this.

Figure (1.11 c) shows how the model represents core-periphery networks. Nodes of one block (core) are well connected with itself and with other partition (periphery). From the last case, we can note that SBM with one group is the Erdos Renyi random graph (1.11 d) because all probabilities inside and between groups are equal.

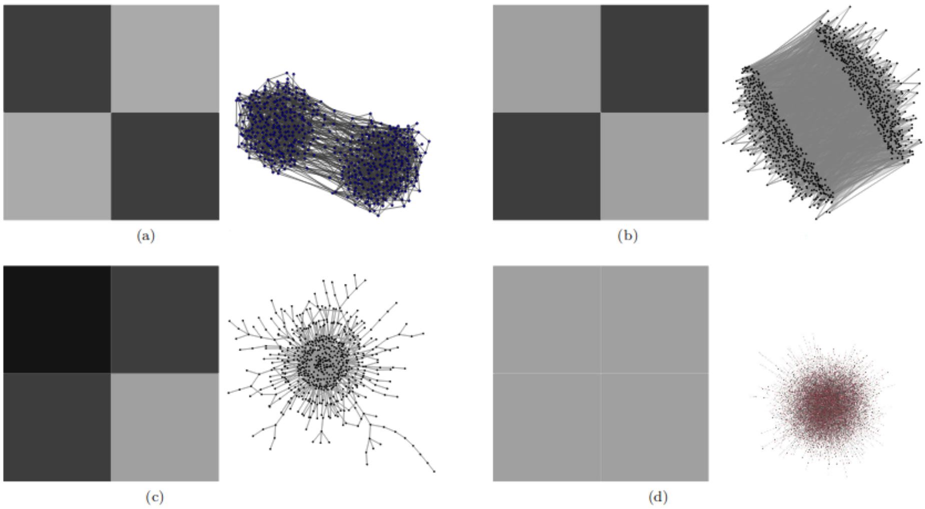

Figure 1.11: Stochastic Block model for different networks structures. (a) assortative. (b) dissortative. (c) core-periphery. (d) Erdos Renyi random graph.

The benefit of this model is that we can generate many networks with similar group structure. The model can fit real data, which results in finding network communities. For the given network $G$ and number of groups $k$, the best nodes partition $g$ is found by maximizing the likelihood function. Beside inferring communities, SBM has application in prediction of missing links. This simply formulated model has many variants, motivated by specific properties of real data. For example, for networks which are degree heterogeneous, there is degree corrected SBM. In some social networks, users can belong to more than one group, and this can be modelled with mixed membership SBM. Other extensions include application to bipartite, weighted network, hierarchical model, etc. Also, several algorithms for optimization of likelihood function are proposed. The overview of these versions and methods are given in [21].

## 1.5 The probability distributions

The shape of degree distribution is important for getting the first insight into the characteristics of the complex network. When nodes are generated at random and any two nodes are linked with the same probability $p$, we expect the binomial distribution, or for larger networks it is Poisson distribution $P(k) = \frac{1}{k!}e^{-\langle k\rangle}\langle k\rangle^k$, where $\langle k\rangle = Np$. A different approach is to add one node and connect it randomly to the network at each time step. The obtained network then has the exponential degree distribution $P(k) = e^{-\lambda k}$. These are exponentially bounded distributions, meaning they decay exponentially or faster for the large values.

On the other hand, heavy-tailed distributions decay slower than exponential, and the events for large values are rare but still possible. For example, in the preferential attachment model, degree distribution emerges to the power law. Also, many empirical data exhibit the heavy-tailed distribution. Even if they look like a power law, after statistical analysis, it may be concluded that the data deviate from the power law and could be equally good or even better fitted with some other distribution. Commonly used alternative distributions are log-normal distribution, stretched-exponential or power-law with an exponential cutoff.

This section gives an overview of relevant distributions and methods for fitting data and testing the quality of the performed fit.

### 1.5.1 The properties of distributions

**Power-law** The power-law distribution is defined as

$$p(k) = Ck^{-\gamma} \tag{1.25}$$

where parameter $\gamma$ is an exponent of the power-law distribution while the C is the normalizing constant.

The distribution can take discrete and continuous values, and it is defined for positive values $k > 0$, so there is a lower bound to the power-law function $k_{min}$. For the discrete case $C = 1/\zeta(\gamma, k_{min})$, while in the continuous case $C = (\gamma - 1)k_{min}^{\gamma-1}$.

The power-law distribution is called scale-free distribution. If we scale the value $k$ for the factor 2 the ratio of $p(x)/p(2x)$ is constant and does not depend on the $k$. We'll find that these criteria are not satisfied by any other distribution.

$$\frac{p(k)}{p(2k)} = \frac{Ak^{-\gamma}}{A(2k)^{-\gamma}} = 2^{\gamma} \tag{1.26}$$

The scale-free function is defined as $p(bx) = g(b)p(x)$. The solution of this equation is $p(x) = p(1)x^{-\gamma}$, where $\gamma = -p(1)/p^{'}(1)$ lead us to conclusion that if function is self-similar it has to be power-law.

**Lognormal distribution**. The variable $x$ has the lognormal distribution if the random variable $y = ln(x)$ is distributed as normal distribution.

$$f(y) = \frac{1}{2\pi\sigma}e^{-(y-\mu)^2/2\sigma^2} \tag{1.27}$$

where $\mu$ is the mean, and $\sigma$ is the standard deviation. The density distribution of the log-normal distribution is defined as

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}}e^{-(log(x)-\mu)^2/2\sigma^2} \tag{1.28}$$

The lognormal distribution has finite mean $e^{\mu+1/2\sigma^2}$, and the variance $e^{2\mu+\sigma^2}(e^{\sigma^2-1})$. [22]. Despite the finite moments, the log-normal distribution can be similar to the power-law distribution. If the variance is large, then the probability function on the log-log plot appears linear for a large range of values.

Using the **multiplicative processes**, we can generate the log-normal distribution. The log-normal distribution is generated by processes that economist Gibrat called the law of proportionate effect. If we start from the organism of size $S_0$. At each time step, the organism may grow or shrink according to the random variable $\epsilon$,

$$S_t = \epsilon_t S_{t-1} \tag{1.29}$$

When the system's state at time t is proportional to the state at the previous time step, we have the multiplicative process. The $\epsilon$ is a proportionality constant that can change over time. The current state depends only on the initial size $S_0$ and the $\epsilon$ variables.:

$$S_t = \epsilon_t S_{t-1} = \epsilon_t \epsilon_{t-1} ... \epsilon_2 \epsilon_1 S_0 \tag{1.30}$$

If $\epsilon_t$ is drawn from the log-normal distribution, then $S_t$ also follows log-normal, as the product of log-normal distributions is again log-normal. Still, the $\epsilon$ distribution does not determine the distribution of the $S_t$. Taking the logarithm of the equation:

$$ln(S_t) = ln(S_0) + \sum_{i=0}^{t} ln(\epsilon_i) \tag{1.31}$$

The sum of the logarithms of the $\epsilon_t$, according to the Central Limit Theorem (CLT), follows the normal distribution. The CLT states that the sum of identically distributed random variables with finite variance converges to the normal distribution. If $ln(S_t)$ is normally distributed, then $S_t$ follows the log-normal distribution.

The multiplicative processes generate the log-normal distribution. Introducing threshold in the multiplicative process leads to the power law. For example, in the Champernowne model, individuals are divided into classes according to their income. The minimum income is m. People between incomes m and $\gamma m$ are in the first class, in the second class are people with incomes between $\gamma m$ and $\gamma^2 m$. The individuals can change their class, so it is described as a multiplicative process, but with a threshold, as income can not be lower than m. If we fix $\gamma = 2$, and consider that with probability $p_{i,i-1} = 2/3$ the change is from higher to lower class. In contrast, with probability, $p_{i,i+1} = 1/3$ individual goes to higher class. In this process, the distribution of incomes emerges to the power-law distribution.

**Power law with exponential cut-off**. The density function has following form

$$p(k) = Ck^{-\gamma}e^{-\lambda k} \tag{1.32}$$

where $k > 0$ and $\gamma > 0$. This function combines the power-law and exponential terms responsible for an exponentially bounded tail. Taking the logarithm $ln(p(k)) = lnC - \gamma lnk - \lambda k$, when $k << 1/\lambda$ the second term dominates, so distribution follows te power-law, with exponent $\gamma$. Otherwise, the $\lambda x$ term dominates, resulting in an exponential cutoff for high values.

**Streched exponential** The stretched exponential distribution is defined as:

$$p(k) = ck^{\beta-1}e^{-(\lambda k)^{\beta}} \tag{1.33}$$

the parameter $\beta$ is stretching exponent determining the properties of the function $p(k)$. For $\beta = 1$, the function is exponential. For $\beta < 1$ it is hard to distinguish the distribution from the power law. We have a compressed exponential function for $\beta > 1$, so $k$ varies in the narrow range.

### 1.5.2 Plotting the distributions

The first step in analysing the empirical data is to create the frequency plot or histogram. Data are binned in equal intervals, and the number of data points within the interval are plotted. When plotting heavy-tailed distributions, it is hard to determine whether the distribution is exponential or power law. If data are from power law distribution on the double logarithmic scale, they will look linear:

$$logp(k) = \gamma log(k) + c \tag{1.34}$$

On the log-log scale, we can notice that in the tail of the distribution data are noise. As the size of the bins is constant, the bins' density for large values also becomes large. To avoid the fluctuations in the tail, we can use logarithmic binning. The noise is reduced by dividing the $x$ axis into $n$ bins $b_n = c^n$, so the following bin is wider than the previous one. For the base $c$ we can choose any value $c > 1$. Similarly, the binning can take the following form $b_n = k_0 \exp{(cn)}$, where $k_0$ is the minimum data point, while the $c$ is the arbitrary base. All data points between values $[b_n, b_{n+1})$ are represented with one point $p(k_n) = N_n/b_n$, where $N_n$ is number of nodes found in the bin $b_n$ and $k_n = \sum_i k_i/N_n$ is average degree of the nodes in the bin $b_n$. By averaging over bins in the tail, noise in the tail of the distribution is reduced. Still, no matter how bin size is chosen, the information about original data points is lost, especially in the distribution tail where bins are larger and include more samples. Figure 1.12 shows how different distributions look like on linear (first column) and log-log scale (second column).

Instead of plotting the probability distribution it is possible to calculate the cumulative distribution, defined as $P(k) = \int_k^\infty p(k^{'})dk^{'}$ for continuous function or as $P(k) = \sum_{k^{'}=1}^{k} p(k^{'})$ for the discrete function. For example the CDF function for power law is also power-law function but with exponent $\gamma - 1$: $P(k) = k^{-(\gamma-1)}$. Note that for cumulative distribution, it is not necessary to use log-binning.

### 1.5.3 Estimating the distribution parameters

The maximum likelihood estimation(MLE) is a method where we consider that data comes from a particular distribution, so we want to maximize the likelihood of the data to find the distribution parameters. For given set of i.i.d. observations $x_1, x_2, ...x_n$, sampled from the distribution $p(x)$ we can define the likelihood function [23]. The likelihood function tells us how likely it is to have the given data if the distribution parameters are $\theta$.

$$L(\theta|x_1, ...x_n) = \prod_{i=1}^{i=n} p(x_i|\theta) \tag{1.35}$$

The parameter that maximize the likelihood function is $\theta_{max} \in argmaxL(\theta|x_1, ...x_n)$.

We can solve the equation and derive the expression for maximum likelihood parameters. The parameters can be obtained with numerical optimization for distributions where an analytical solution is unavailable. In practice is much easier to work with logarithm of the likelihood, $log(L) = \sum_{i=1}^{i=N} p(\theta|x_i)$, as the product changes to summation. For the power-law distribution, the exponent is calculated as $\gamma = 1 + n[\sum ln\frac{k_i}{k_{min}}]^{-1}$. For discrete distribution solution may be obtained optimizing the log-likelihood function $log(L) = log\prod_{i=1}^{n} \frac{k_i^{-\gamma}}{\zeta(\gamma, k_{min})}$.

We can use the MLE method to fit any distribution to the data. Even if obtained distribution looks like a power law, and some parameters are estimated, it does not have to be that data are truly from the power-law distribution. With the MLE method alone, it is impossible to distinguish between different distributions, and we do not know how accurate the obtained results are. To determine the quality of the fit, we need to use another statistical method called the **goodness-of-the-fit** test. The main idea is based on calculating the distance between distributions of empirical data and the model using Kolmogorov-Smirnov statistics. The Kolmogorov Smirnov statistics is the maximum distance between the CDF of the data and the fitted model.

$$D = max|S(x) - P(x)| \tag{1.36}$$

First, we fit empirical data to get model parameters and calculate the KS statistics of this fit. Then, many synthetic data sets are generated with model optimized model parameters. Then each synthetic data set is fitted, and KS statistics are obtained relative to its model. From there, we can calculate **p-value**, the fraction of times that KS-statistics in synthetic distributions is larger than in empirical data.

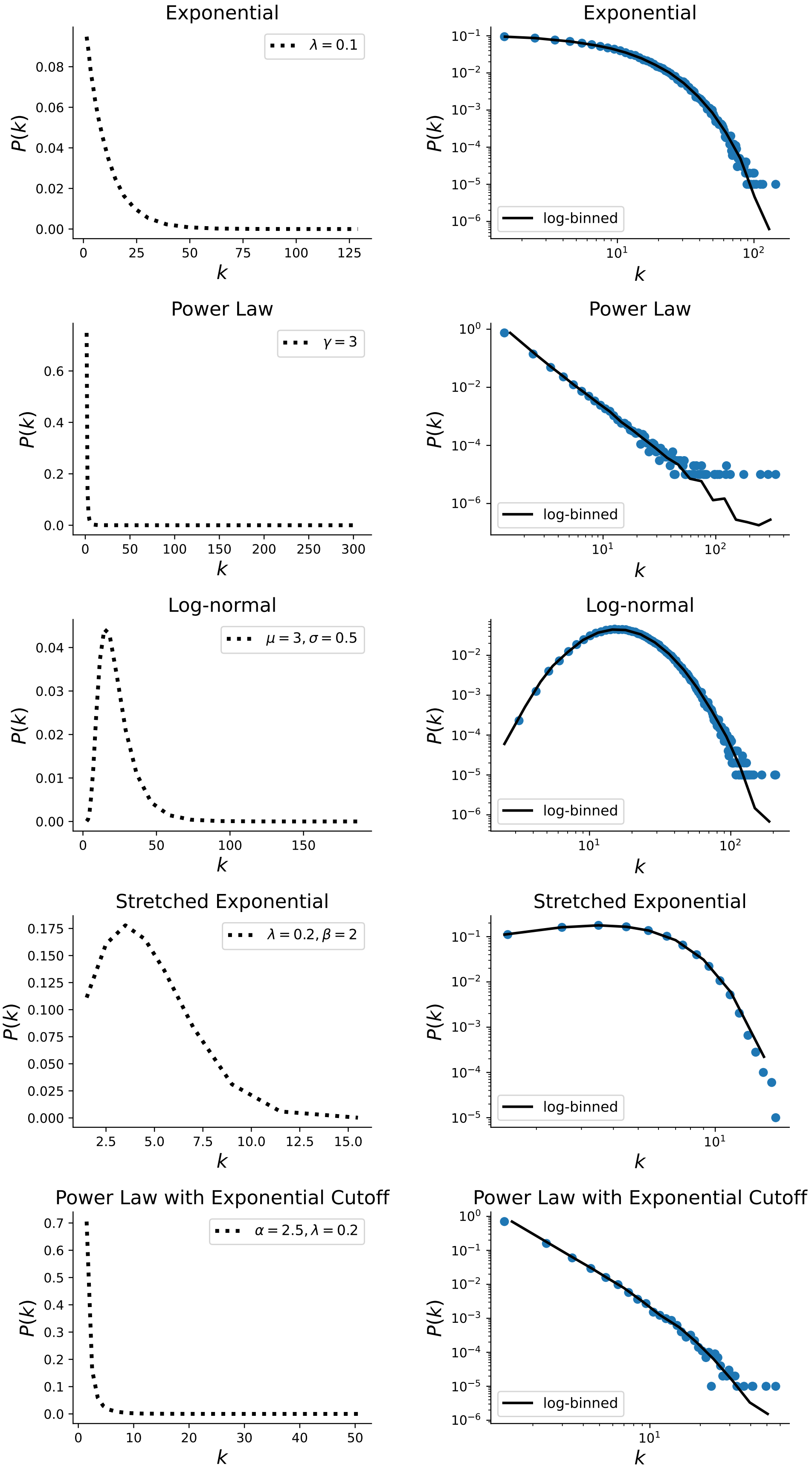

Figure 1.12: Probability distributions on linear and double logarithmic scale.

If $p - value < 0.1$, we reject the hypothesis that this distribution describes the empirical data.

Otherwise, the model can not be rejected. Failing to reject the hypothesis does not mean the model is a correct distribution for the data. Other distributions might fit the data equally good or even better. To have an accurate p-value, we need a large sample. For a small number of synthetic distributions, it is possible to have a high p-value, even if the distribution is the wrong model for the data. Finally, we need to be confident in obtained results. The same procedure can be repeated for different distributions. If the p-value for the power law is high, while for alternative distribution, it is low, we can conclude that the power law is a more probable fit.

Another method, the **likelihood ratio test**, allows us to compare two distributions directly. The distribution with a higher likelihood under empirical data is a better fit. We can calculate the likelihood ratio, or it is easier to obtain the likelihood ratio's logarithm because its sign determines which distribution is a better fit. For given two distributions $p_1(x)$and $p_2(x)$.

The likelihoods are defined as $L_1 = \prod_{i=1}^{n} p_1(x)$ and $L_2 = \prod_{i=1}^{n} p_2(x)$, or the ratio of likelihoods as $R = \frac{L_1}{L_2} = \prod_{i=1}^{n} \frac{p_1(x)}{p_2(x)}$. Taking the logarithm, we obtain the log-likelihood ratio

$$\mathcal{R} = \sum_{i=1}^{n} [logp_1(x_i) - logp_2(x_i)] \tag{1.37}$$

As data $x_i$ are independent, by central limit theorem, their sum $\mathcal{R}$ becomes normally distributed, with expected variance $\sigma^2$. We can approximate the variance as

$$\sigma^2 = \frac{1}{n}\sum_{1}^{n}[(l_i - l_i) - (< l >^{(1)} - < l >^{(2)})]$$

When $R > 0$ the first distribution is a better fit to data, and when $R < 0$, the other one should be chosen. When $R = 0$, it is not possible to distinguish between two distributions. The sign of $R$ is not enough criteria to conclude which distribution is a better fit, and it is a random variable subject to statistical fluctuations. We need a log-likelihood ratio that is sufficiently positive or negative to ensure that its sign does not result from fluctuations.

If we are suspected that the expectation value of the log-likelihood ratio is zero, the observed sign of  is simply the product of fluctuations and can not be trusted. The probability that the measured log-likelihood ratio has a magnitude as large or larger than the observed value R is given as

$$p = \frac{1}{\sqrt{2\pi n\sigma^2}} \int_{-\infty}^{-|\mathcal{R}|} e^{-x^2/2n\sigma^2} dx + \int_{|\mathcal{R}|}^{\infty} e^{-x^2/2n\sigma^2} dx \tag{1.38}$$

Here we use the standard two-tail hypothesis test, assuming that the null hypothesis is $R = 0$. If the p-value is larger than a threshold, the R sign is unreliable, and the test does not favour any distribution. If p is small, $p < 0.1$ then it is unlikely that the observed sign is obtained by chance, so we reject the null hypothesis that $R = 0$.

## 1.6 Fractal analysis

Approach to study complex systems is detecting time-series of selected variables. Some systems are characterised by periodic ot nearly periodic behaviour. In complex systems this periodic behaviour is not limited to one or two characteristic frequencies. They extend over wide spectrum and fluctuations on many time scales as well as well as broad distributions. In these cases dynamics of the system is characterized by scaling laws, which are valid over a wide range of time scales or frequencies. If dynamic of the system can be described with one scaling exponent system is monofractal, otherwise we deal with multifractal time-series.Rescaling of time $t$ by a factor $a$ may require rescaling of the time-series values $x(t)$ by factor $a^H$, then we have the self-similarity. The Hurst exponent, $H$ characterize the type of self-affinity.

$$x(t) = a^H x(at)$$

### 1.6.1 Long and Short-term correlations

The time-series are presistent such that a large value is usually followed by large values and small values. Considering the increments $\delta x_i = x_i - x_{i-1}$, of self-affine series $i = 1, .., N$, with N values measured equidistant in time, so $\delta x_i$ can be either persistent, independent or anti-persistent. For the random walk with $H = 0.5$ the increments are independent of each other. Persistent and anti-persistent increments, where a positive increment is likely to be followed by another positive or negative increment. For stationary data with constant mean and standard deviation the auto-covariance function can determine the degree of persistence.

$$C(s) = \langle \Delta x_i \Delta x_{i+s} \rangle = \frac{1}{N-s} \sum_{i=1}^{N-s} \Delta x_i \Delta x_{i+s} \tag{1.39}$$

If the data are uncorrelated the $C(s) = 0$. Short range correlations are described by $C(s)$ declining exponentially

$$C(s) = exp(-s/t_c)$$

such behaviour is typical for increments generated by an auto-regresive process

$$\Delta x_i = c\Delta x_{i-1} + \epsilon_i$$

with random uncorrelated offsets $\epsilon_i$ and $c = exp(-1/t_c)$.

For long-range correlations $\int C(s)$ diverges in the limit for long series. In practive this means that caracteristic time can not be defined because it increaseas with N. Contrary to short-range correlations, the correlation function decline as power-law

$$C(s) = s^{-\gamma}$$

This type of behavior can be modeled by Fourier filtering techniques. Long-term correlated, behavior of $\Delta x_i$ leads to self-affine scalling beahviout characterized by Hurst exponent $H = 1 - \gamma/2$.

A direct calculation of the $C(s)$ is difficult due to present noise in the data and non-stationarity. Non-stationarities make the definition of $C(s)$ probalematic, because the its average is not well defined, also $C(s)$ fluctuates around zero on large scales s, so it is not possible to obtain the correct correlation exponent $\gamma$. Instead calculating $C(s)$ we can calculate the Hurst exponent $H$.

### 1.6.2 Rescaled range analysis

Hurst proposed method called the **rescaled range analysis** $R/S$. It begins with splitting the time series $x_i$ into non overlaping segments $\nu$ of the size s, having $N_s = int(N/s)$ segments. Then is calculated the profile in each segment.

$$Y_\nu(j) = \sum_{i=1}^{j} (x_{\nu s+i} - \langle x_{\nu s+i} \rangle_s)$$

Substracting the averages, constant trends in the data are eliminated. The differences between minimum and maximum value and the standardan deviation in each segment are calculated as: $R_\nu(s) = maxY_\nu(j) - minY_\nu(j)$, $S_\nu(s) = \sqrt{\frac{1}{s}\sum Y_\nu^2(j)}$

Finally, the rescaled range is averged over all segments to obtain the fluctuation function F(s).

$$F_{RS}(s) = \frac{1}{N_s}\sum\frac{R_\nu(s)}{S_\nu(s)} \sim s^H$$

where the H is Hurst exponent. The values of H that can be obtained by Hurst rescaled analysis are $0 < H < 2$. Values $H < 1/2$ indicate long-term anti-correlated data while $H > 1/2$ long-term positively correlated data.

### 1.6.3 Fluctuation analysis

The **fluctuation analysis** is based on the random walk theory. For given time series $\{x_i\}$ with length N, we first define global profile in the form of cumulative sum, equation 1.40, where $\langle x\rangle$ represents average of the time series. Subtracting the mean of the time series is supposed to eliminate global trends.

$$Y(j) = \sum_{i=0}^{j}(x_i - \langle x\rangle), \quad j = 1, ..., N \tag{1.40}$$

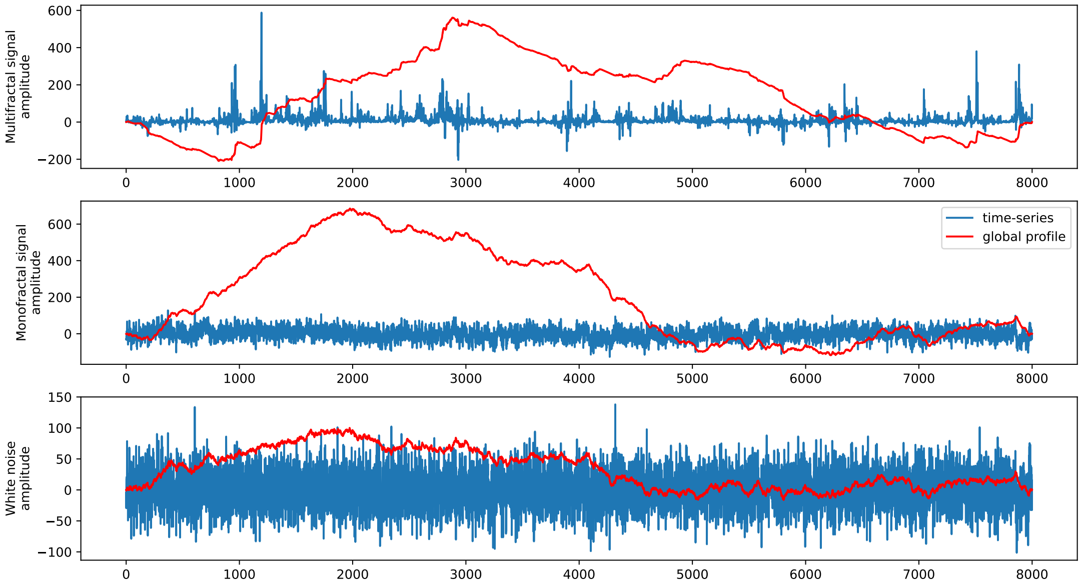

Figure 1.13: Multifractal, monofractal and whitenoise signals.

The profile of the signal Y is divided into $N_s = int(N/s)$ non overlapping segments of length $s$. If $N$ is not divisible with s the last segment will be shorter. This is handled by doing the same division from the opposite side of time series which gives us $2N_s$ segments. Then we calculate the fluctuations in the each segment $F^2(\nu, s)$ and finally averaging over all subsequences we obtain the mean fluctuation. From the scaling of function we can determine the Hurst exponent.

$$F_2(s) = [\frac{1}{2N_s} \sum F^2(\nu, s)]^{1/2} \sim s^H \tag{1.41}$$

The several methods are proposed for calculating the fluctuating function $F^2(\nu, s)$:

- The simplest way to calculate the fluctuations, is to consider difference of the values at end-points of each segment. This is same as eliminating the linear trend from each segment.

$$F^2(\nu, s) = [Y(\nu s) - Y((\nu+1)s)]^2$$

- The trends present in the time-series does not have to be linear. To deal with non-stationary time-series it is necessary to remove the polynomial trend, within each segment by least-square fitting. The method is called detrended fluctuation analysis (DFA). From each segment $\nu$, local trend $p^m_{\nu,s}$ - polynomial of order m - should be eliminated, and the variance $F^2(\nu, s)$ of detrended signal is calculated as in equation

$$F^2(\nu, s) = \frac{1}{s} \sum_{j=1}^{s} \left[Y(j) - p^m_{\nu,s}(j)\right]^2 \tag{1.42}$$

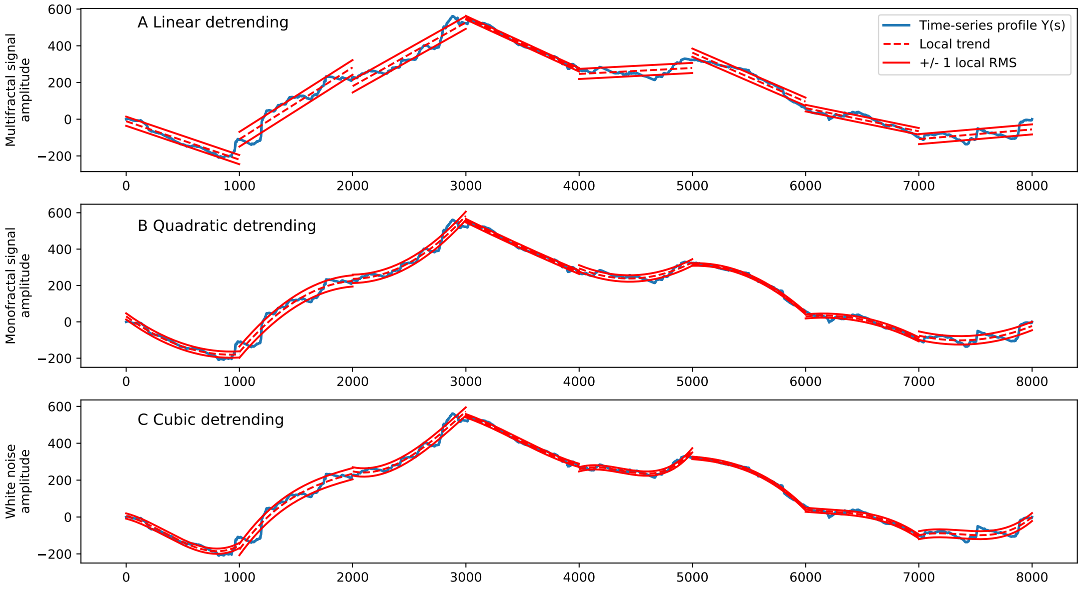

Figure 1.14: Detrending the signal, for the segments of length $s = 1000$.

### 1.6.4 Multifractality of the signals

Many records do not exhibit a simple mono fractal scaling behaviour. The scaling behaviour may be more complicated , and different scaling exponents can be found for many interwoven fractal subsets of the time series. In this case the multifractal analysis must be applied. Two general types of multifractality exist. The multifractality due to a broad probability distribution for values of the time series, the multifractality can not be destroyed. Multifractality due to different long-term correlations of the small and large fluctuations. In this case the probability density function of the values can be regular distribution with finite moments, and the corresponding shuffled series will exhibit non-multifractal scaling as correlations are destroyed with shuffling procedure. If both kinds of multifractality are present, the shuffled series will show weaker multifractality than the original series. Multifractal analysis will reveal higher order correlations, multifractal scaling can be observed if the scaling behaviour of small and large fluctuations is different. Extreme events might be more or less correlated than typical events. Multifractal detrended fluctuation analysis (MFDFA) is used [24, 25] to estimate multifractal Hurst exponent H(q).

$$F_q(s) = \left\{ \frac{1}{2N_s} \sum_{\nu}^{2N_s} \left[ F^2(\nu, s) \right]^{\frac{q}{2}} \right\}^{\frac{1}{q}}, q \neq 0$$

The MFDFA for $q = 2$ is equivalent DFA method. The value of $H(0)$, which corresponds to the limit $H(q), q - 0$, cannot be determined directly because of the exponent diverge. Instead logarithmic averaging procedure has to be considered.

$$F_0(s) = \exp \left\{ \frac{1}{4N_s} \sum_{\nu}^{2N_s} ln \left[ F^2(\nu, s) \right] \right\}, q = 0 \tag{1.43}$$

The fluctuating function scales as power-law $F_q(s) \sim s^{H(q)}$ and the analysis of log-log plots $F_q(s)$ gives us an estimate of multifractal Hurst exponent $H(q)$.

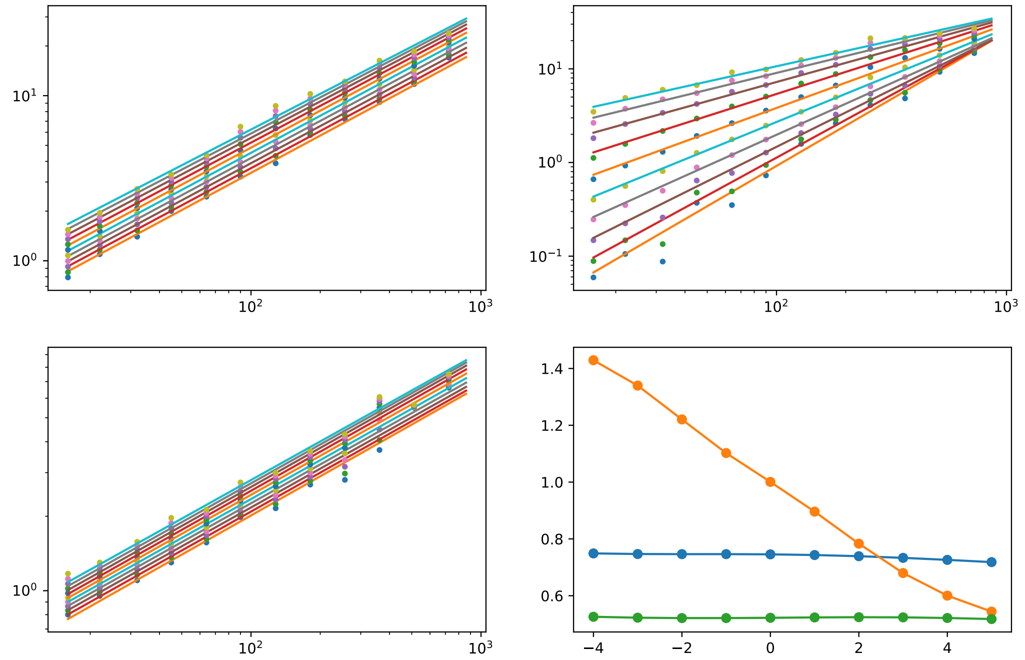

Figure 1.15: Multifractal, monofractal and whitenoise signals.

For the monofractal time series, H(q) is independent of q, meaning that scaling is indentical for all segments, and averaging fluctuations gives indentical scaling for all values of q. If small and large fluctuations scale differently, there will be dependence of h(q) on q. Positive values of q, segments with large variance are dominant in the Fq(s), so positive q describes segments with large fluctuations. The negative values of q, H(q) describes the scaling of the segments with small fluctuations.

## 1.7 Dynamical reputation model

With Dynamic Interaction Based Reputation Model (DIBRM) we can quantify the user reputation $T_n$ using equation 1.44.

$$T_n = T_{n-1}\beta^{\Delta_n} + I_n \tag{1.44}$$

The first part of the equation takes into account the previous reputation value $T_{n-1}$ that is weighted with coefficient $\beta_n^{\Delta}$. Depending on the frequency of the interaction controls the reputation decay. Parameter $\beta$ is forgeting factor, parameter

$$\Delta_n = \frac{t_n - t_{n-1}}{t_a} \tag{1.45}$$

measures time between interactions $t_n$ and $t_{n-1}$, while $t_a$ is characteristic time window of interaction. While $\Delta_n < 1$, user is frequently active and the number of sequential activities $A_n$ increases by 1. When the $\Delta_n$ is large the first part of equation gives more contribution to the first part, so the reputation decreases.

The second part of the equation, $I_n$ is reputation gained with each interaction. The basic value of each interaction is given as $I_{bn}$ and the parameter $\alpha$ is weight of cumulative part.

$$I_n = I_{b_n}(1 + \alpha(1 - \frac{1}{A_n + 1})) \tag{1.46}$$

In our implementation of the model, the trust is updated each day for every user irrespective of their activity status. Therefore, the decay itself is a combination of $\beta$ and $\Delta_n$: the more days pass without recorded interaction from a specific user, the more their reputation decays. Lower values of beta lead to faster decay of trust as shown on figure.

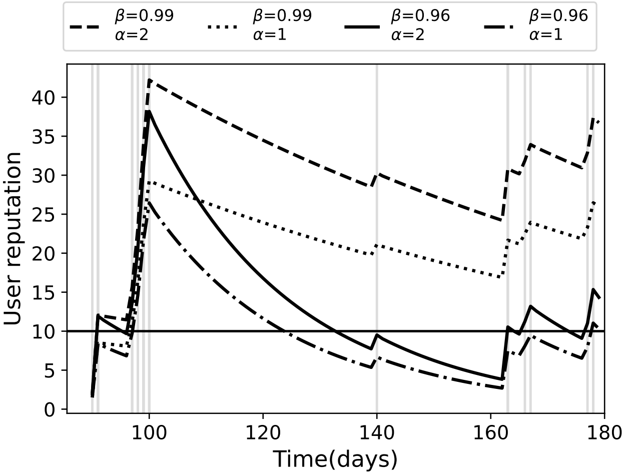

Figure 1.16: Multifractal, monofractal and whitenoise signals.

# Bibliography

[1] Stefano Boccaletti, Vito Latora, Yamir Moreno, Martin Chavez, and D-U Hwang. Complex networks: Structure and dynamics. *Physics reports*, 424(4-5):175–308, 2006.

[2] V. Latora, V. Nicosia, and G. Russo. Complex networks: Principles, methods and applications. 2017.

[3] L da F Costa, Francisco A Rodrigues, Gonzalo Travieso, and Paulino Ribeiro Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in physics*, 56(1):167–242, 2007.

[4] Luciano da Fontoura Costa, Osvaldo N Oliveira Jr, Gonzalo Travieso, Francisco Aparecido Rodrigues, Paulino Ribeiro Villas Boas, Lucas Antiqueira, Matheus Palhares Viana, and Luis Enrique Correa Rocha. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. *Advances in Physics*, 60(3):329–412, 2011.

[5] Guido Caldarelli. *Scale-Free Networks: Complex Webs in Nature and Technology*. 05 2007.

[6] Albert-László Barabási and Márton Pósfai. *Network science*. Cambridge University Press, Cambridge, 2016.

[7] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[8] Petter Holme and Jari äki. Temporal networks. *Physics reports*, 519(3):97–125, 2012.

[9] Naoki Masuda and Renaud Lambiotte. *A Guide to Temporal Networks*. 10 2016.

[10] Petter Holme. Modern temporal network theory: a colloquium. *The European Physical Journal B*, 88(9):1–30, 2015.

[11] Gautier Krings, Márton Karsai, Sebastian Bernhardsson, Vincent D Blondel, and Jari Saramäki. Effects of time window size and placement on the structure of an aggregated communication network. *EPJ Data Science*, 1(1):1–16, 2012.

[12] Naomi A Arnold, Benjamin Steer, Imane Hafnaoui, Hugo A Parada G, Raul J Mondragon, Félix Cuadrado, and Richard G Clegg. Moving with the times: Investigating the alt-right network gab with temporal interaction graphs. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–17, 2021.

[13] Mason A Porter. What is... a multilayer network. *Notices of the AMS*, 65(11), 2018.

[14] Alberto Aleta and Yamir Moreno. Multilayer networks in a nutshell. *Annual Review of Condensed Matter Physics*, 10(1):45–62, 2019.

[15] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P. Gleeson, Yamir Moreno, and Mason A. Porter. Multilayer networks. *Journal of Complex Networks*, 2(3):203–271, 07 2014.

[16] Tiago A Schieber, Laura Carpi, Albert Díaz-Guilera, Panos M Pardalos, Cristina Masoller, and Martín G Ravetti. Quantification of network structural dissimilarities. *Nature communications*, 8(1):1–10, 2017.

[17] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1 – 44, 2016. Community detection in networks: A user guide.

[18] Martin Rosvall, Jean-Charles Delvenne, Michael T. Schaub, and Renaud Lambiotte. Different approaches to community detection. *CoRR*, abs/1712.06468, 2017.

[19] Ryan J Gallagher, Jean-Gabriel Young, and Brooke Foucault Welles. A clarified typology of core-periphery structure in networks. *arXiv preprint arXiv:2005.10191*, 2020.

[20] Kamalika Basu Hajra and Parongama Sen. Phase transitions in an aging network. *Physical Review E*, 70(5):056103, 2004.

[21] Thorben Funke and Till Becker. Stochastic block models: A comparison of variants and inference methods. *PLOS ONE*, 14(4):1–40, 04 2019.

[22] Michael Mitzenmacher. A brief history of generative models for power law and lognormal distributions. *Internet mathematics*, 1(2):226–251, 2004.

[23] J. Nair, A. Wierman, and B. Zwart. *The Fundamentals of Heavy Tails*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2022.

[24] Jan W Kantelhardt, Stephan A Zschiegner, Eva Koscielny-Bunde, Shlomo Havlin, Armin Bunde, and H Eugene Stanley. Multifractal detrended fluctuation analysis of nonstationary time series. *Physica A: Statistical Mechanics and its Applications*, 316(1-4):87–114, 2002.

[25] Espen Alexander Fürst EAFI Ihlen. Introduction to multifractal detrended fluctuation analysis in matlab. *Frontiers in physiology*, 3:141, 2012.

[26] Claudiu Duma, Nahid Shahmehri, and Germano Caronni. Dynamic trust metrics for peer-to-peer systems. In *16th International Workshop on Database and Expert Systems Applications (DEXA'05)*, pages 776–781. IEEE, 2005.

[27] A. Melnikov, J. Lee, V. Rivera, M. Mazzara, and L. Longo. Towards dynamic interaction-based reputation models. In *2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA)*, pages 422–428, 2018.