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 [**?**].

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 [**?**]. 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) [**?** **?**]. 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.

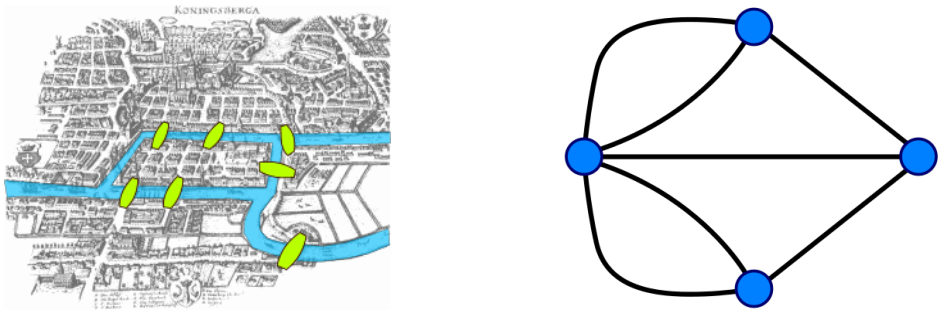

Figure 1.1: The Kronigsber problem of seven bridges.

1.2 Types of networks

The graph or network G is defined as $G = (V, E)$, where $V = \{v_1, v_2, ...v_N\}$ is a set of N nodes (vertices), and $E = \{e_1, ..e_L\}$ is a set of L edges (links). The edge is pair of nodes $e = (v_i, v_j)$, such that $\{v_i, v_j\} \in V$. 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** $|A| = N \times N$ [**?**]. 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 adjacancy 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.

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 [**?**]. 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 [**?**]. 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.

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

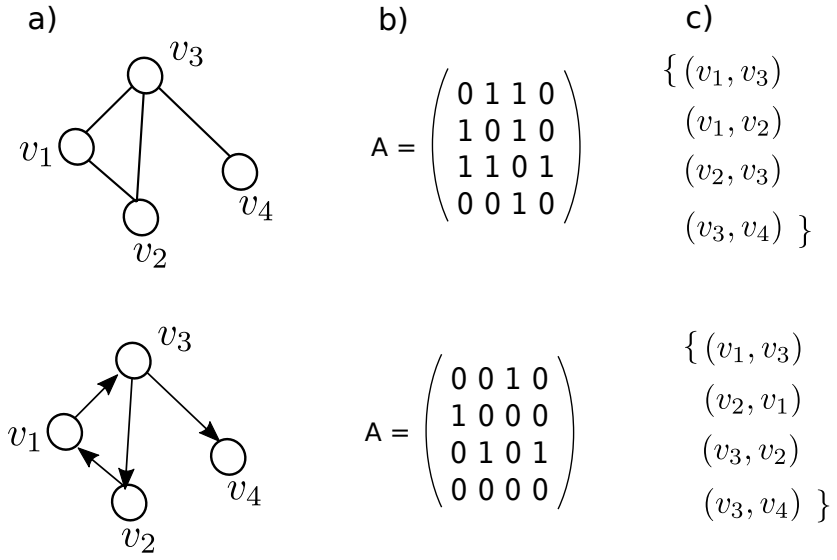

Figure 1.2: a) Graph representation of undirected (top panel) and directed (bottom panel) network. The same networks are represented with adjacency matrices column b), and edge list representation in column c).

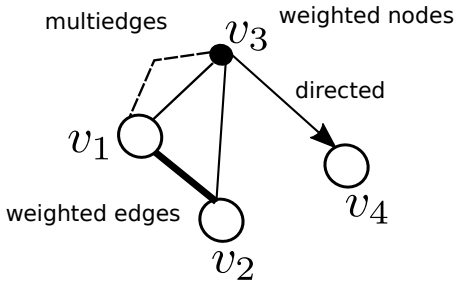

Figure 1.3: The complex networks may represent different characteristics of the system. The edges can be directed, weighted or multiply. Also nodes can be assigned with different weights or any relevant feature.

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[**? ?**]. 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 [**?** **?**].

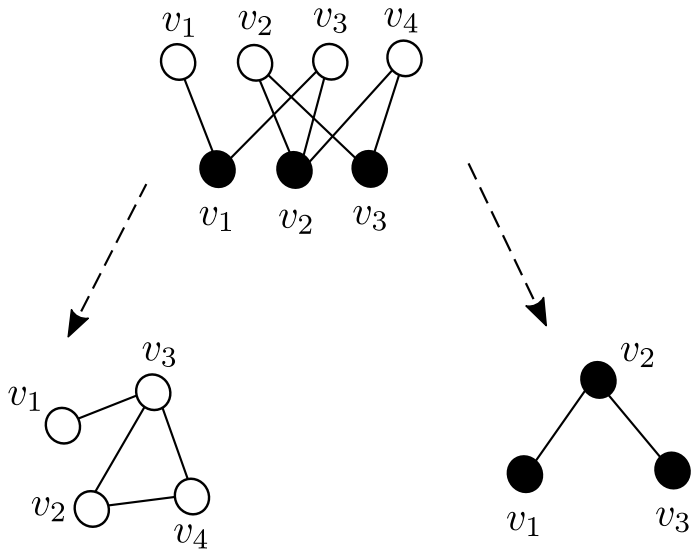

Figure 1.4: Bipartite graph and two partition projections.

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 [**?**]. 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 Δt is event duration [**?**]. 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.

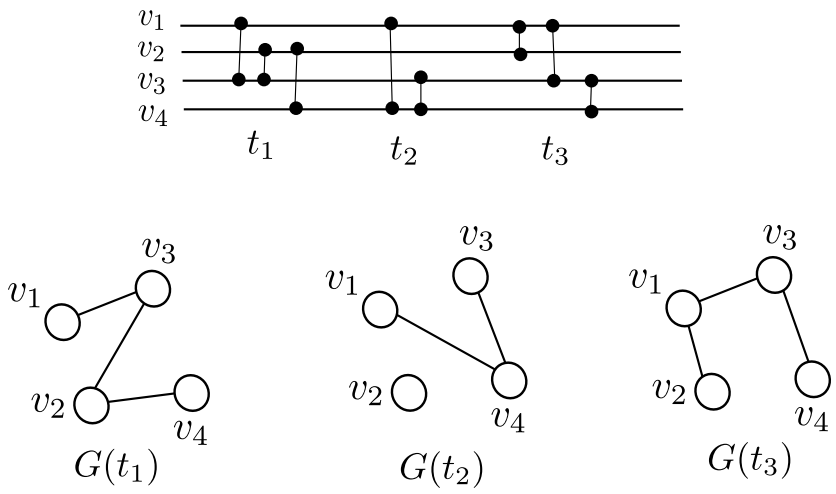

Figure 1.5: Temporal network.

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 [**?**]. 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 [**? ?**].

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 [**?**]. 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 [**?**]. Or the airline transportation network, where each layer represents the flights of different airline companies [**?**].

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 γ 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 γ exponent should be $\gamma > 2$. Therefore, real networks have a scale-free structure with the emergence of the hubs [**?**].

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 $V(G) = \{v_1, ..v_n\}$, $A[i, j] = 1$ if there is a link between nodes v_i and v_j. If G is a simple graph with adjacency matrix A and degree sequence $d = [d_1, ..., d_n]$

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

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 [**?**], $kP(k'|k)P(k) = k'P(k|k')P(k')$ [**?**]. 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 σ_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 [**?**]

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

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 [**?**], [**?**].

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.

Louvain algorithm

Infomap algorithm

Block Stochastic Model

Core-periphery structure

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

$$p_{ij} = \frac{k_i k_j}{2L}$$

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

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 $p_{11} > p_{12} > p_{22}$.

SBM 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 θ, p given network G is proportional to likelihood of generating network G, prior of SBM matrix and prior on block assignments:

$$P(\theta, p|G) = P(G|\theta, p)P(p)P(\theta) \tag{1.10}$$

$$P(G|\theta,p) = \prod_{i<j} p_{r_i s_j}^{A_{ij}} (1 - p_{r_i s_j})^{1-A_{ij}} \tag{1.11}$$

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 θ 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 [**?**] 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 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 $r \leq p$

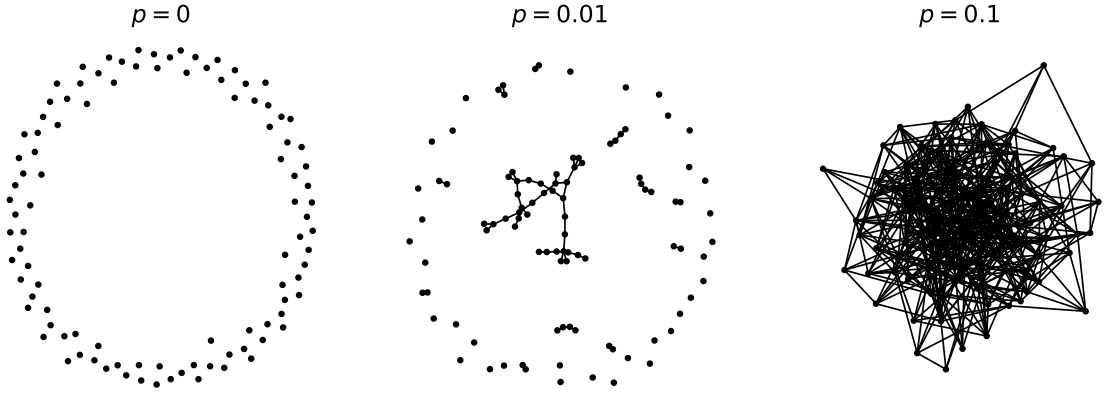

Figure 1.6: ER graph with $N = 100$ nodes and different linking probabilities p.

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

$$\langle L \rangle = \frac{N(N-1)}{2} p \tag{1.12}$$

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:

$$\langle k \rangle = (N-1)p \tag{1.13}$$

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

$$P(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k} \tag{1.14}$$

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 $\langle k \rangle$

$$p(k) = \frac{1}{k!} e^{-\langle k \rangle} \langle k \rangle^k \tag{1.15}$$

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

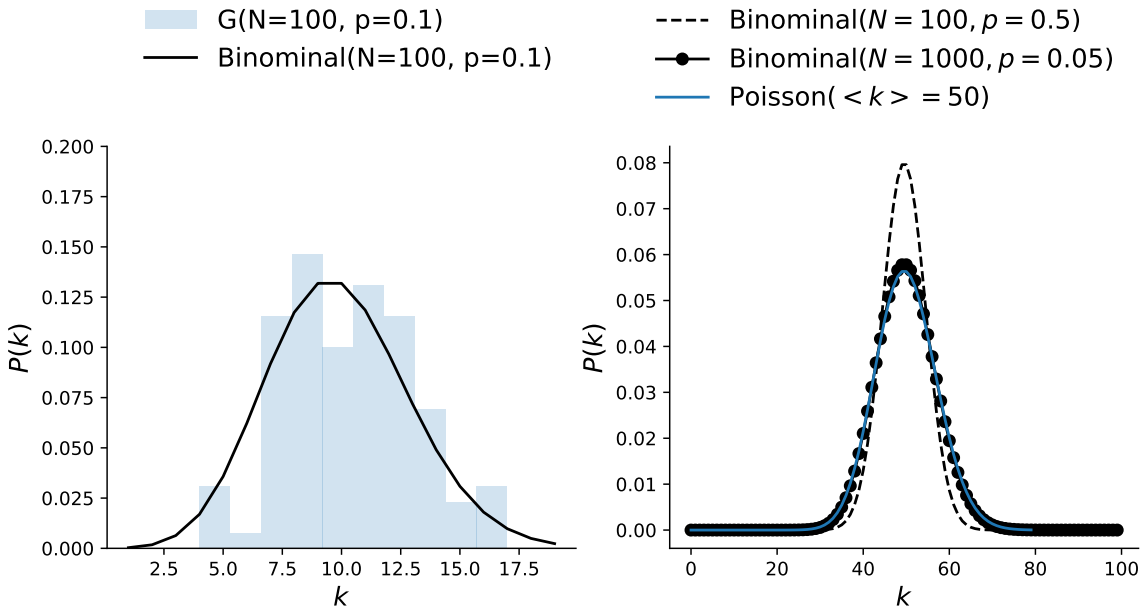

Figure 1.7: Degree distribution of ER graph. Degree distribution of small networks follow binominal. Larger networks are better approximated with Poison distribution, and degree distribution for fixed average degree $< k >$ becomes independent of the network size.

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.

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.

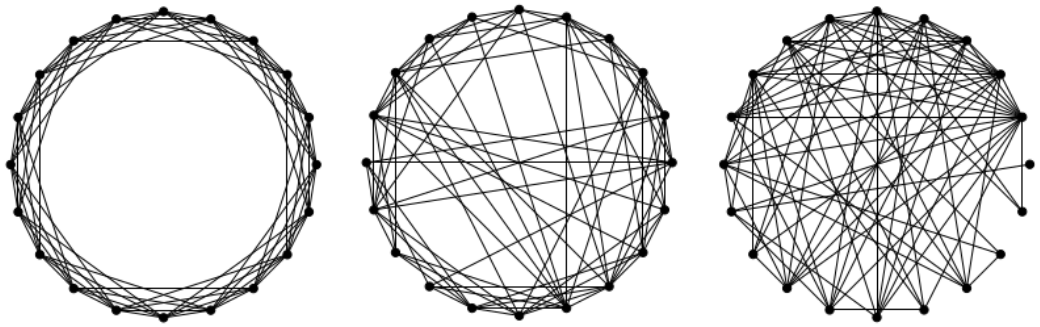

Figure 1.8: Watts and Strogatz graph model creation

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 Π is then proportional to node degree k:

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j} \tag{1.16}$$

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.

$$\frac{dk_i}{dt} = m\Pi(k_i) = m\frac{k_i}{\sum_j k_j} = m\frac{k_i}{m_0 + 2mt} \tag{1.17}$$

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.

$$k_i(t) = m\left(\frac{t}{t_i}\right)^{\frac{1}{2}} \tag{1.18}$$

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:

$$P(k) = \frac{2m(m+1)}{k(k+1)(k+2)} \tag{1.19}$$

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 Π. 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:

$$\Pi(k_i) = {k_i}^{\beta} \tag{1.20}$$

The probability that newly added node attaches to node i depends on the existing i-th node degree k_i, and the parameter β. 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.

$$\Pi_i(t) \sim k_i \tau_i^{\alpha} \tag{1.21}$$

The parameter α 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 α, each node is connected to the oldest node in the network. The scale-free properties are present; the power-law exponent γ deviates from $\gamma = 3$. It is found that γ ranges between 2 and 3.

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

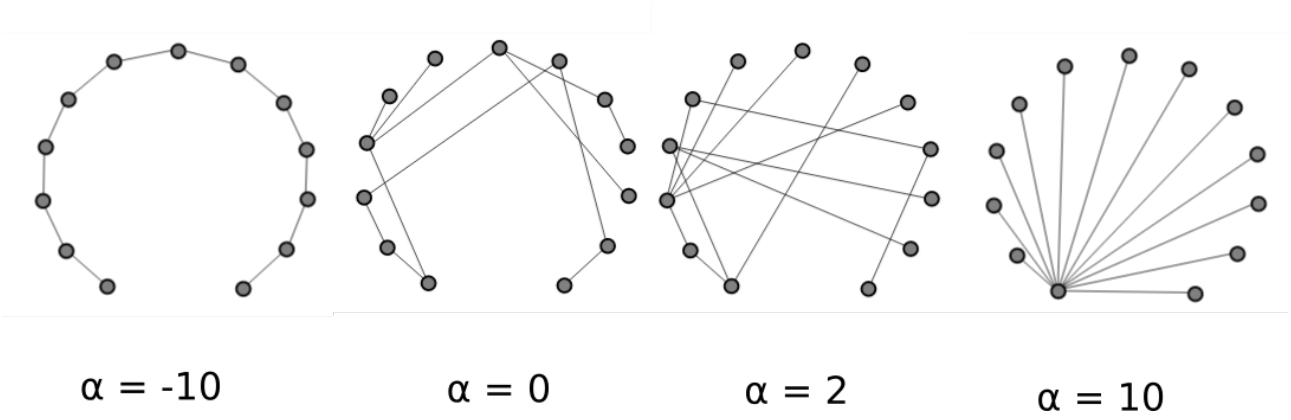

Figure 1.9: Aging model

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

$$\Pi_i(t) \sim k_i(t)^{\beta} \tau_i^{\alpha} \tag{1.22}$$

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

- 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

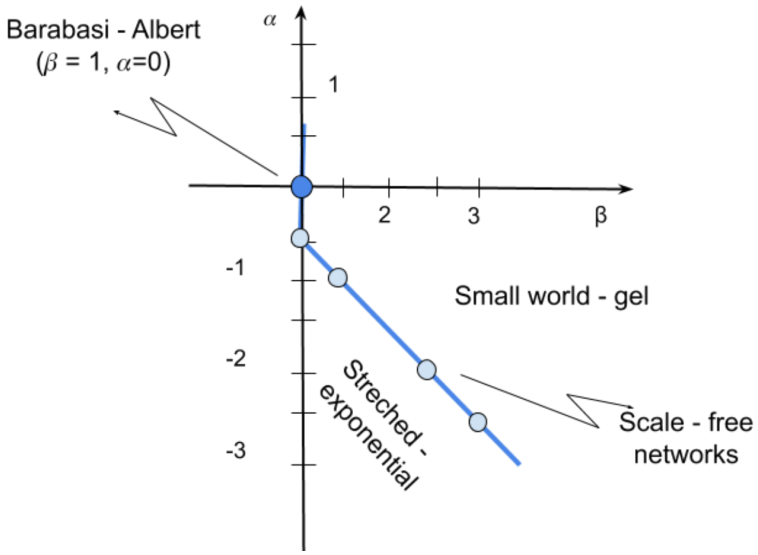

Figure 1.10: Phase diagram of aging network model

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 [**?**] 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.

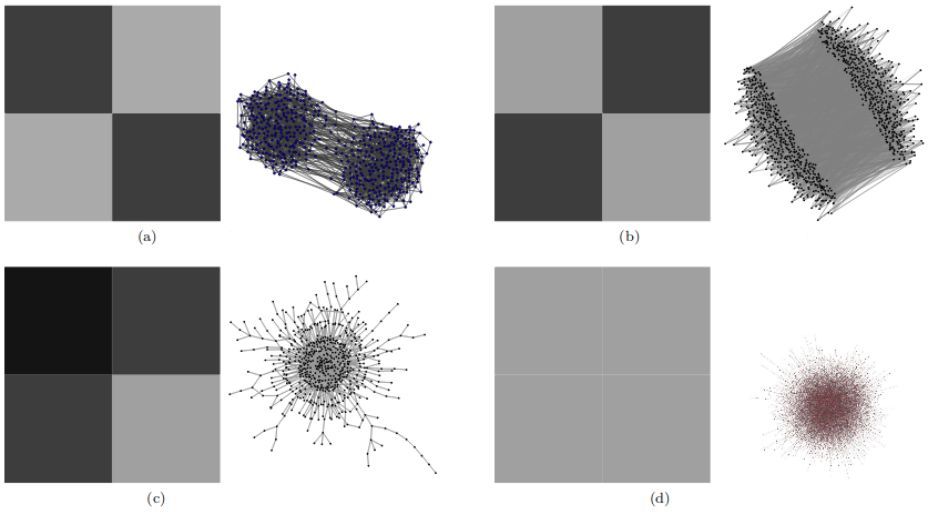

Figure 1.11: Stochastic Block model for different networks structures. (a) assortative. (b) dissortative. (c) core-periphery. (d) Erdos Renyi random graph.

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 [**?**].

1.5 The properties of probability distributions

1.5.1 Poisson distribution

Poisson distribution is exponentially bounded distribution. The property of these distributions is that they decay exponentialy or faster for high x. The largest expected value grows as $logx$, meaning that outliers representing the high x values are rate. In the network science the most common distribution is Poisson distribution, while outside the network science the most common distribution is Gaussian distribution.

On the other hand, fat tailed distributions, long tailed or heavy tailed distributions refer to distributions whose decay for large x is slower than exponential. In this distributions events with large x value are rare but possible.

1.5.2 Power law

The power-law distribution is defined as

$$p(k) = Ck^{-\gamma} \tag{1.23}$$

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

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

The likelihood function for continuous data set is defined as

$$l(k) = \prod p(k) = \prod \frac{\gamma-1}{k_{min}}(\frac{k_i}{k_{min}})^{-\gamma} \tag{1.24}$$

Minimizing the loglikelihood function, exponent is calculated as $\gamma = 1 + n[\sum ln\frac{k_i}{k_{min}}]^{-1}$.

For discrete distribution the log likelihood is $log(l) = ln\prod \frac{k_i^{-\gamma}}{\zeta(\gamma,k_{min})}$. For this equation does not exist the analytical solution, but the equation can be numericaly optimized.

The power-law distribution is called scale-free distribution. This is distribution that is the same on all scales.

For example if we take power law distribution:

$$\frac{p(x)}{p(2x)} = \frac{Ax^{-\alpha}}{A(2x)^{-\alpha}} = 2^{\alpha}$$

This ratio is constant and does not depend on the x. If we take any other distribution, we'll find that this criteria is not satisfied.

In the general form, scale free function is defined as:

$$p(bx) = g(b)p(x)$$

Solving this equation we get $p(x) = p(1)x^{-\alpha}$, where $\alpha = -p(1)/p^{'}(1)$, concluding that if function is self-similar, it has to be a power-law.

1.5.3 Lognormal distribution

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

$$f(y) = \frac{1}{2\pi\sigma} e^{-(y-\mu)^2/2\sigma^2} \tag{1.25}$$

where μ is mean, and σ is standard deviation. The density distribution of the lognormal distribution is defined as

$$f(x) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-(log(x)-\mu)^2/2\sigma^2} \tag{1.26}$$

The lognormal distribution has finite mean $e^{\mu+1/2\sigma^2}$, and the variance $e^{2\mu+\sigma^2}(e^{\sigma^2-1})$. [**?**].

Despite the finite moments, the lognormal distribution can be similar to power-law distribution. If the variance of lognormal distribution is large then the probability function on log-log plot appears linear for large range of values.

The normal distribution has property that the sum of two independent normal random variables is normal random variable with mean $\mu_1 + \mu_2$, and variance $\sigma_1^2 + \sigma_2^2$. It follows that two log-normally distributed random variables also have a lognormal distribution. [**?**].

1.5.4 Power law with exponential cutoff

The density function has the form

$$p(x) = Cx^{-\gamma} e^{-\lambda x} \tag{1.27}$$

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

1.5.5 Streched exponential

the stretched exponential

$$p(x) = cx^{\beta-1} e^{-(\lambda x)^\beta} \tag{1.28}$$

the parameter β is streching exponent determining the properties of the function $p(x)$. For $\beta = 1$, the function is exponential. For $\beta < 1$ it is hard to distinguish the distribution from power-law. For $\beta > 1$ we have a compressed exponential function, so x vary in the narrow range.

Whenever we encounter fat-tailed distributions, there is discussions asking which distribution offers the best fit to the data. In many systems empirical data is not sufficient to distinguish between distributions.

1.5.6 Plotting the fat-tailed distributions

With plotting the distribution of the power-law data on the double logarithmic scale we should obtain the straight line, as $ln(p(k)) = \gamma ln(k) + c$. The tail of the distribution is noisy. As the size of the bins is constant, the density of the bins for large distribution values 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 before. 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 this, averaging over bins in the tail, which now have more sampled points, reduces the statistical errors.

Instead of plotting the probability distribution (i.e. creating histogram), it is possible to calculate the cumulative distribution. $P(k) = \int_x^\infty p(x^{'})dx^{'} = x^{-(\gamma-1)}$. It is also the power-law distribution with exponent $\gamma - 1$. Note that for cumulative distribution it is not necessary to use log-binning.

1.5.7 The goodness-of-fit

Minimizing the loglikelihood of function for given data, allow us to analitically or even numerically fit distributions and estimate fit parameters. Still, it does not tell us how good the fit is.

The figure shows the distributions of three small datasets, drawn from power-law with $\gamma = 2.5$, lognormal $\mu = 0.3, \sigma = 2$ and exponential with $\lambda = 0.125$. The distributions look as straight line on the log-log plot, we could try to fit them to the power law distribution and obviosly, some model parameters could be estimated. It is not straightforward to say weather particular data really follows given distribution. Even if data follow powe-law, their observed distributions are not likely to exactly follow power-lawl there are some deviations because of the random nature of sampling procedure.

The basic approach is to sample many synthetic data sets from a tru power-law and measure how they fluctuate from power-law form and compare results on empirical data. If empirical data are much further from power-law than syntetic one, the power-law is not plausible fit to the data.

A standard aproach to answering this kind of questions is to use goodness-of-fit test which generate the p-value that quantifies the plausibility of the hypothesis. Measuring the distance between distributions of empirical data and the model. p-value is defined to be fraction of synthetic distances that are larger than the empirical distance.

For measuring the distance between distributions we can use the Kolmogorov-Smirnov statistics, but in general other goodness of fit measures could be used. The Kolmogorov Smirnov statistics is the maximum distance between the CDF of the data and the fitted model.

$$D = max|S(x) - P(x)| \tag{1.29}$$

First we fit empirical data to get model parameters, and calculate the KS statistics of this fit. Then, large number of syntetic data sets, are generated with model optimized model parameters. Then each syntetic dataset is fitted, and KS statistics is obtained relative to its own model. Then simply we count the fraction when the KS statistics of syntetic distributions is larger than in empirical data, that represents the p-value.

If $p < 0.1$ then 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 correct distribution for the data. There might be other distributions that fit the data equaly good, or even better. For having accurate p-value we need large sample. For small number of syntetic distributions it is possible to have high p-value, even if the distribution is wrong model to the data.

Using goodness of fit it is possible to calculate p-value for any distribution. For example comparing p-value of power-law fit to the p-value of trunctated power-law fit, we can conclude which one is better fit to data. If p-value for powe law is high, while for alterantive distribution it is low, we can conclude that power-law is more probable fit.

The another method called the likelihood ratio test allows us to directly compare two distributions. The distribution with higher likelihood under epmpirical data is better fit. We can calculate the likelihood ratio, or it is easier to obtain the logarithm of likelihood ratio, because its sign determine which distribution is better fit. For given two distributions $p_1(x)$and $p_2(x)$.

The likelihoods are defiend 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 loglikelihood ratio

$$\mathscr{R} = \sum_{i=1}^{n} [lnp_1(x) - lnp_2(x)] \tag{1.30}$$

As data x_i are independent, by central limit theorem their sum $\mathscr{R}$ becomes normally distributed, with expected variance σ^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 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 better fit. It is a random variable subject to statistical fluctuations. We need loglikelihood ratio that is sufficently positive or negative, and to be sure that its sign is not result of fluctuations.

If we are suspcted thaat the true expectation value of the loglikelihood ratio is zero, meaning that observed sign of $\mathscr{R}$ is simply product of fluctuations and an not be trusted. The probability that measured log likelihood ratio has magnitude as large or larger than observed value R is given as

$$p = \frac{1}{\sqrt{2\pi n\sigma^2}} \int_{-\infty}^{-|\mathscr{R}|} e^{-x^2/2n\sigma^2} dx + \int_{|\mathscr{R}|}^{\infty} e^{-x^2/2n\sigma^2} dx \tag{1.31}$$

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

1.6 Multiplicative processes

Using the multiplicative processes we can generate the power-law but also the log-normal distribution.

The lognormal 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 random variable ε,

$$S_t = \varepsilon_t S_{t-1}$$

When the state of the system at time t is proportional to the state of the system at previous time step, we have the multiplicative process. The ε is proportionality constant that can change over time. The state of the system at time step t is determined by the product of the various ε_t and the initial size S_0

$$S_t = \varepsilon_t S_{t-1} = \varepsilon_t \varepsilon_{t-1} ... \varepsilon_2 \varepsilon_1 S_0$$

The S_t is drawn from the log-normal distribution. If ε_t is drawn from the lognormal distribution, then S_t also follows lognormal, as the product of lognormal distributions is again lognormal. Still, the distribution of the ε does not determine the distribution of the $S_j t$.

Taking the logarithm of the equation:

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

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

Using the multiplicative processes it is possible to generate the lognormal distribution. Also similar process may lead to the power-law distribution.

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

1.7 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 multifraactal time-series.

Rescaling of time t by a factor a may require rescaling of the time-series values $x(t)$ by factor a^H, to get the self-similaraity. In this case it is:

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

The Hurst exponent characterize the type of self-affinity.

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.

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 δx_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.

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

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

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

with random uncorrelated offsets ε_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 Δx_i leads to self-affine scalling beahviout characterized by $H = 1 - \gamma/2$.

A direct calculation of the $C(s)$ is difficult due to present noise in the data and nonstationarity. Non-stationarities make the definition of $C(s)$ probalematic, because the 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 γ.

Hurst's rescaled-range analysis The method called rescaled range analysis R/S was proposed by the Hurst. It begins with spliting the time series x_i into non overlaping segments ν 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. Finally 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).

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

the H is Hurst exponent introduced in the first equation. The values of H that can be obtained by Hurst rescaled analasys are $< 0 H < 2$. Values $H < 1/2$ indicate long-term anticorrelated data, $H > 1/2$ indicated long-term positively correlated data. For power-lae correlations decaying faster than $1/s$, $H = 1/2$, like for uncorrelated data.

On the other hand the standard fluctuation analysis is based on the random walk theory. For time series with zero mean, we consider the global profile, the cumulative sum:

$$Y(j) = \sum x_i$$

, and then study how fluctuations of the profile, in a given time window of size s increase with s.

We first divide each record of N elements into Ns non-overlaping segments of the size s, and another Ns non-overlaping segments starting from the end. Then we calculate the fluctuations in the each segment. In the standard FA we get the fluctuations just from the values of the profile at both endpoints of each segment.

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

Then we can average F^2 over all subsequences to obtain the mean fluctuation

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

For the relevant case of long-term correlations where $C(s)$ follows the power-law behaviour, $F_2(s)$ also increases by power-law.

The fluctuation exponent is indentical with Hurst exponent for monofractal data.

1.7.1 Detrended fluctuation analysis

DFA was introduced by Peng et al. and represents an important method for describing the non-stationary time series. As before methods it is also base on random walk theory, and the FA analysis is special case with linear detrending.

Like in FA method first one calculates the global profile of the time series. DFA deaks with monotonous trends in a detrending procedure. This is done by establishing a polynomial trend within each segment by least-square fitting and subtracting this trend from the original profile, detrending.

$$Y_s(j) = Y(j) - y^m_{\nu,s}(j)$$

The degree of the polinomial can be varied in order to eliminate constant m=0, linear m=1, quadratic m-2, or higher order trends of the profile function. The variance of the detrended profile in each segment gives us mean-square fluctuations

$$F^2(\nu s) = \frac{1}{s}\sum Y^2_s(j)$$

Finaly fluctioations over all segments are averaged, to obtain the mean fluctuations $F_2(s)$ as in eq. F2s, and as before, from the scaling of the fluctuating function we can determine the Hurst exponent.

1.7.2 Multifractality of the signals

Multifractal detrended fluctuation analysis (MFDFA) [**?** **?**] to estimate multifractal Hurst exponent H(q). For given time series $\{x_i\}$ with length N, first we define global profile in the form of cumulative

sum, equation 1.32, where where $\langle x \rangle$ represents average of the time series:

$$Y(j) = \sum_{i=0}^{j} (x_i - \langle x \rangle), \quad j = 1, ..., N \tag{1.32}$$

Subtracting the mean of the time series is supposed to eliminate global trends. 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. From each segment ν, 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 1.33:

$$F^2(\nu, s) = \frac{1}{s} \sum_{j=1}^{s} \left[Y(j) - p^m_{\nu,s}(j) \right]^2 \tag{1.33}$$

Then the q-th order fluctuating function is:

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

The 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.

$$F_0(s) = \exp \left\{ \frac{1}{4N_s} \sum_{\nu}^{2N_s} ln \left[F^2(\nu, s) \right] \right\}, q = 0 \tag{1.34}$$

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 $H(q)$.

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.

Also, large fluctuations are characterized by smaller scaling exponent.

Multifractal signal has different scaling properties over scales while monofractal is independent of the scale, i.e., H(q) is constant.

1.8 Dynamical reputation model

Any dynamical trust or reputation model has to take into account distinct social and psychological attributes of these phenomena in order to estimate the value of any given trust metric [**?**]. First of all, the dynamics of trust is asymmetric, meaning that trust is easier to lose than to gain. As part of asymmetric dynamics, in order to make trust easier to loose the trust metric has to be sensitive to new experiences (recent activity or the absence of the activity of the agent), while still maintaining nontrivial influence of old behavior. The impact of new experiences has to be independent of the total

number of recorded or accumulated past interactions, making high levels of trust easy to lose. Finally, the trust metric has to detect and penalize both the sudden misbehavior and the possibly long term oscillatory behavior which deviates from community norms.

We estimate dynamic reputation of the Stack Exchange users using Dynamic Interaction Based Reputation Model (DIBRM) [**?**]. This model is based on the idea of dynamic reputation, which means that the reputation of users within the community changes continuously through time: it should rapidly decrease when there is no registered activity from the specific user in the community (reputation decay), and it should grow when frequent, constant interactions and contributions to the community are detected. The highest growth of user's reputation is found through bursts of activity followed by short period of inactivity.

In our implementation of the model, we do not distinguish between positive and negative interactions in the Stack Exchange communities. Therefore, we treat any interaction in the community (question, answer or comment) as potentially valuable contribution. In fact, evaluation criteria for Stack Exchange websites going through beta testing, described in SI, do not distinguish between positive and negative interactions. The percentage of negative interactions in the communities we investigated was below 5%, see Table 1 in SI. Filtering positive interactions would also require filtering out comments because they are not rated by the community, and that would eliminate a large portion of direct interactions between the users of a community, which is essential for estimating their reputation.

In DIBRM, reputation value for each user of the community is estimated combining three different factors: 1) *reputation growth* - the cumulative factor which represents the importance of users' activities; 2) *reputation decay* - the forgetting factor which represents the continuous decrease of reputation due to inactivity; *the activity period factor* - measuring the length of the period of time in which the change of reputation happened. In case of Stack Exchange communities, the forgetting factor has a literal meaning, as we can assume that past contributions provided by a user are being forgotten by active users as their attention is captured by more recent content.

In line with the the basic dichotomy of reputation dynamics, which revolves around the varying influence of past and recent behavior, DIBRM has two components: *cumulative factor* - estimating the contribution of the most recent activities to the overall reputation of the user; *forgetting factor* - estimating the weight of past behavior. Estimating the value of recent behavior starts with the definition of the parameter storing the basic value of a single interaction I_{b_n}. Cumulative factor I_{c_n} then captures the additive effect of recent successive interactions. The reputational contribution I_n of most recent interaction n of any given user is estimated in the following way:

$$I_n = I_{b_n} + I_{c_n} = I_{b_n}(1 + \alpha(1 - \frac{1}{A_n + 1})) \tag{1.35}$$

Here, α is the weight of the cumulative part and A_n is the number of sequential activities. If there is no interaction at t_n, this part of interactions has a value of 0. Important property of this component of dynamic reputation is the notion of sequential activities. Two successive interactions made by a user are considered sequential if the time between those two activities is less or equal to the time parameter t_a which represents the time window of interaction. This time window represents maximum time spent by the user to make a meaningful contribution (post a question or answer or leave a comment).

$$\Delta_n = \frac{t_n - t_{n-1}}{t_a} \tag{1.36}$$

If $\Delta_n < 1$ is less than one the number of sequential activities A_n will increase by one, which means that the user is continuing to communicate frequently. On the other hand, large values Δ_n greatly increase the effect of the forgetting factor. This factor plays a major role in updating the total dynamic reputation of a user in each time step (after every recorded interaction):

$$T_n = T_{n-1}\beta^{\Delta_n} + I_n \tag{1.37}$$

Here, β is the forgetting factor. 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 β and Δ_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 **??**.