tightcenter

panikzettel.htwr-aachen.de

Social Networks^{2020} Panikzettel

Daniel Sous

Version 17 — 03.08.2021

Contents

This Panikzettel is about the lecture Social Networks by Prof. Dr. Markus Strohmaier held in the summer semester 2020.

> We have two Panikzettel on Social Networks! This one is from the summer semester of 2020. There is also one written mainly in the summer of 2019: Social Networks 2019 Panikzettel.

This Panikzettel is Open Source. We appreciate comments and suggestions at https://git.rwth-aachen.de/philipp.schroer/panikzettel.

1 Introduction and Concepts

1.1 Graph Theory

A graph consists of components called nodes or vertices. In general, the set of node or vertices is called N or V. Interactions between two components are indicated with links or edges. The set of links or edges is generally called L or E. The complete system is called network or graph and is defined as $G = (N, L)$.

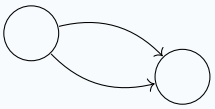
The links of an undirected graph are symmetrical i.e. they have no direction. The links of a directed graph can be asymmetrical i.e. they have an explicit direction.

Definition: Self-Loop

A link whose source and destination are the same node.

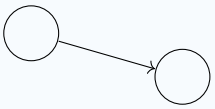
Definition: Multigraph

A graph that have loops or multiple edges between two nodes.



Definition: Simple Graph

A simple graph contains no loops and no multiple links.



In *binary networks* links just have two states: present or not. In *weighted networks* each links is labeled with a value indicating the connection's strength. A strength of 0 indicates that there is no link.

1.2 Measuring Network Structure

Definition: Adjacency Matrix

In a binary network (N, L) an *adjacency matrix* $A^{n \times n}$ with $n = |N|$ can be used to describe the links $L \subseteq N \times N$:

$$A_{ij} = \begin{cases} 1 & \text{if there is a link from } i \text{ to } j, \\ 0 & \text{otherwise.} \end{cases}$$

If the network is a weighted network, the elements of the matrix represent the weights w_{ij}:

$$A_{ij} = w_{ij}.$$

The maximum number of links in a graph with $n = |N|$ nodes is given with:

$$L_{\max} = \binom{n}{2} = \frac{n(n-1)}{2}$$

A graph with $|L| = L_{\max}$ is called *complete graph*.

Definition: Density

Given a graph $G = (N, L)$. The relationship between the number of links and the possible number of links is called *density*:

$$\text{density}(G) = \frac{|L|}{L_{\max}} = \frac{|L|}{\binom{n}{2}} = \frac{2|L|}{n(n-1)}$$

where $n = |N|$.

A network is called *sparse* if the number of nodes is in the same order as the number of links ($|N| \approx |L|$). Networks with $|L| \gg |N|$ are said to be *dense*.

Definition: Degree, Average Degree

Undirected graphs: k_n defines the number of links node $n \in N$ has.

Directed graphs: k_n^{out} defines the number of outgoing and k_n^{in} defines the number of incoming links of node $n \in N$.

$\langle k \rangle$ defines the average degree:

$$\langle k \rangle := \frac{1}{|N|} \sum_{i \in N} k_i = \frac{2|L|}{|N|}$$

1.3 Distance

Some definitions about distances in a graph $G = (N, L)$.

Definition: Walk

A *walk* is a sequence of nodes in which each node is adjacent to the next one.

Definition: Path

A *path* is a walk without node repeats and different ends (no loop).

Definition: Distance

The *distance* (shortest path, geodesic path) between two nodes is defined as the number of edges along the shortest path connecting them.

Definition: Diameter

The *diameter* $d_{\max}$ defines the maximum shortest path (or maximum distance) between any pair of nodes in a given graph.

1.4 Clustering Coefficient

Let a graph $G = (N, L)$.

Definition: Clustering Coefficient

The *clustering coefficient* of node $n \in N$ is

$$C_n = \frac{2e_n}{k_n(k_n - 1)}$$

where k_n is the degree of node n and e_n is the number of links between the neighbors of node n.

1.5 Small-World Problem

A given network shows the *small-world property* if the number of vertices reachable from a central vertex grows exponentially with the distance. So, a network shows the small-world effect if the average distance between two vertices in the network d_{avg} scales logarithmically or slower with the networks size:

$$d_{\text{avg}} \sim \log|N|$$

1.6 Components

Given a graph $G = (N, L)$.

Definition: Connected Component

A *connected component*[a] is a subset of nodes $C \subseteq N$ where there is a path between each pair of node $x, y \in C$.

- A graph $G = (N, L)$ is called *connected graph* if there is one connected component with $N = C$.
- The largest component is called *giant component*.
- Components that are not the giant component are called *isolates*.
- A link $l \in L$ is called *bridge* if the number of connected components increases by erasing it.

[a] If the same property holds in a directed graph it is called *strongly connected directed graph*.

1.7 Centrality Measures

Centrality measurements try to identify **nodes** that are more important than others. It depends on the network's context which measurement the highest amount of information gives.

1.7.1 Degree Centrality

The *degree centrality* just measures the **number of links** each node has. This centrality measurement is easy to calculate but does not differentiate between links of different importance. Thus, it is a **local measure**, i.e. it does not depend on the rest of the network.

1.7.2 Closeness Centrality

The *closeness centrality* measures the mean distance to all other nodes. The centrality measure states that nodes with a smaller average distance are more important than others. Given the distance d_{ij} between two nodes i and j, the closeness centrality C_i is calculated using the mean distance d_i to all other nodes:

$$C_i = \frac{1}{d_i} \quad \text{where} \quad d_i = \frac{1}{|N| - 1} \sum_{j \in N \setminus \{i\}} d_{ij}.$$

This centrality measurement is weak in small-diameter networks as the range of variation is too narrow. Additionally, it is not even defined for networks with two or more isolated components.

1.7.3 Betweenness Centrality

The *betweenness centrality* of a node v measures how often node v is visited when going along the shortest path of each pair of nodes. So, the betweenness centrality of a node v is given with

$$g(v) = \sum_{\substack{v,s,t \in N, \\ s \neq v \neq t}} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the total number of shortest paths between s and t and $\sigma_{st}(v)$ is the total number of shortest path between s and t that pass through v.

Normalization:

$$g(v) = \frac{g(v)}{0.5 \cdot (|N| - 1) \cdot (|N| - 2)}$$

1.7.4 Eigenvector Centrality

The *eigenvector centrality* introduces the idea that nodes are more important if they have connections to nodes that are themselves important. Step t of the eigenvector centrality is calculated with

$$x(t) = A^t x(0)$$

where $A \in \{0,1\}^{|N| \times |N|}$ is the adjacency matrix and $x(t) \in \mathbb{R}^{|N|}$ is the vector of centrality values.

$$x(t) = A^t \sum_i c_i v_i = \sum_i c_i k_i^t v_i = k_1^t \sum_i c_i \left[\frac{k_i}{k_1}\right]^t v_i$$

k_i eigenvalues of A, k_1 largest eigenvalue.

1.8 Importance of Edges

Generally, ties (links or edges) are classified in two categories. *Strong ties* represent friendships and *weak ties* represent the connection between acquaintances. Weak ties connect different parts of a network. Thus, weak ties are *bridges*.

A node A fulfills the **strong triadic closure property** if A has strong ties to some nodes B and C, then B and C share a strong or weak tie.

Definition: Local Bridge

An edge is called *local bridge* if its node have no friends in common.

Formally: An edge $e = \{i, j\}$ is called *bridge* if the distance between its nodes i and j increases to a value greater than 2 when deleting e.

2 Models of Networks

Models of networks allow us to generate and grow networks according to the given rules and compare different instances with each other.

2.1 Erdős-Rényi Model

A Erdős-Rényi model is an undirected network model which is given with $G(n, p)$ where $n \in \mathbb{N}$ is the number of nodes and $p \in [0, 1]$ is the probability that an edge exists.

Algorithm: Generation of Erdős-Rényi Model

1. Start with n isolated nodes.
2. For each pair of nodes generate a random number r between 0 and 1. If $r < p$ connect the nodes with an edge.

To generate a network of the Erdős-Rényi model with n nodes, $\frac{n \cdot (n-1)}{2} = L_{\max}$ random numbers need to be generated.

The probability of having $m \in \mathbb{N}$ links in an Erdős-Rényi model is given with:

$$P(m) = \underbrace{\binom{\binom{n}{2}}{m}}_{\substack{\text{Number of ways} \\ \text{you can place} \\ m \text{ links}}} \cdot \underbrace{p^m}_{\substack{\text{Probability of} \\ m \text{ successful} \\ \text{links}}} \cdot \underbrace{(1-p)^{\binom{n}{2}-m}}_{\substack{\text{Probability that} \\ \text{remaining links} \\ \text{are not successful}}}$$

As $P(m)$ is a binomial distribution its *mean number of edges* can be derived as

$$\begin{aligned} \langle m \rangle &= \sum_{m=0}^{\binom{n}{2}} m \cdot P(m) \\ &= \binom{n}{2} p \end{aligned}$$

The average degree of $G(n, p)$ is given with $c = (n-1)p$.

2.1.1 Poisson Distribution

The degree distribution of an Erdős-Rényi model forms a Poisson probability distribution $P(k)$ given with

$$P(k) \simeq \frac{c^k}{k!} e^{-c}$$

where $k \in \mathbb{N}_0$ is the degree and $c \in \mathbb{R}$ is the average degree. In **??** three example Poisson distributions are shown.

The global clustering coefficient C of an Erdős-Rényi model $G(n, p)$ is given with probability p. Consequently, in networks with a small p there is very little clustering. These networks behave tree-like locally.

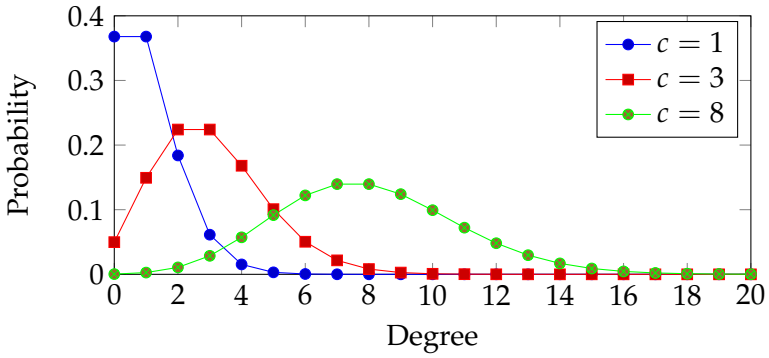


Figure 1: Example Poisson Distributions with mean degree $c = 1, 3, 8$

2.1.2 Phase Transition

The phase transition describes the relation between the mean degree c of an Erdős-Rényi network $G(n, p)$ and the giant component. The mean degree is changed by letting p run from 0 to 1.

$c < 1$	**subcritical regime** The network contains only small tree-like components. The giant component scales with $\log(n)$.
$c = 1$	**critical point** The giant component scales with $n^{\frac{2}{3}}$.
$c > 1$	**supercritical regime** The giant component scales with n. The second largest component scales with $\log(n)$.
$c > \log(n)$	**connected regime** There exists only a single giant component.

Table 1: The four phases of the Erdős-Rényi model.

Let $u \in [0, 1]$ be the fraction of nodes **not** contained in the giant component. Then it is:

$$u = e^{-c(1-u)}$$

2.2 Watts-Strogatz Model (Small-World Model)

Reminder: The small-world phenomenon combines short paths with high clustering.

2.2.1 Circle Model

In the *circle model* all vertices are arranged in the circle and connected to its x nearest neighbors. Thus, the clustering coefficient C can be varied depending on x:

$$C = \frac{3(x-2)}{4(x-1)}$$

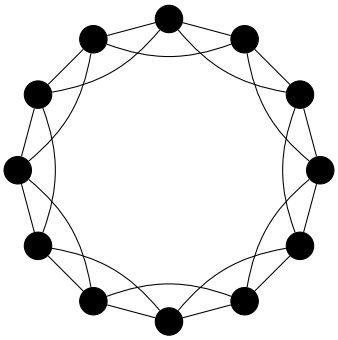


Figure 2: Example Circle model with 12 nodes and $x = 2$

For the **original small-world model** we start with a regular circle model. For all edges we exchange one end for another randomly chosen node with the probability p. The added randomness may create bridges that can decrease average distances drastically.

In a **second version of the small-world model** edges are only added randomly. Thus, no edges are removed.

In the second version we add for each non-shortcut edge (edges that were present in the original circle model) a shortcut edge with probability p at a random location. So, we have $\frac{1}{2}nxp$ shortcuts on average with nxp ends of shortcuts. So, the degree distribution is Poisson distributed

$$p_s = e^{-xp}\frac{(xp)^s}{s!}$$

with mean xp and s number of shortcuts added to a vertex. The total degree k of a vertex is given with $k = s + x$. So $s = k - x$ can be replace in the distribution.

2.3 Configuration Model

The basic idea of *configuration models* is creating a network with a given degree distribution. A Configuration Model is a random graph G defined by $G(n, \vec{k})$ where n is the number of nodes and $\vec{k}$ is a n-dimensional vector that gives a degree k_i for each node i in the graph.

The **first approach** of generating configuration models is calculating the probability that an edge between two nodes exists.

Definition: Probabilistic Links

Let $G(N, E)$ be a graph with $n = |N|$ nodes and $m = |E|$ edges. Given two nodes $i \in N$ and $j \in N$ with degrees k_i and k_j set the probability that an edge $\{i, j\}$ exists to

$$p_{ij} = \frac{k_i k_j}{2m} = \frac{k_i k_j}{\sum_{l=1}^{n} k_l}$$

For larger graph the deviation of the desired degrees k_i are small. But the calculation for this approach are expensive: $\mathcal{O}(n^2)$.

The **second approach** adds k_i stub edges to each node i, where k_i is the degree of node i. Next, pick two stubs at random and join them until no more stubs remain.

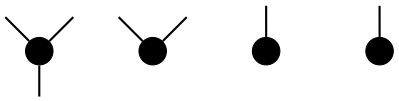


Figure 3: Example nodes with stubs

The calculations for this approach cost $\mathcal{O}(m)$, where m is the number of edges in the resulting graph and $2m$ the number of stubs.

In the previously presented approaches for generating a configuration model it is not guaranteed that there are no self-loops and multiedges. But the probability of a self-loop or multiedge

$$p_{\text{self/multi}} = \frac{1}{2}\left[\frac{\langle k^2\rangle - \langle k\rangle}{\langle k\rangle}\right]^2$$

depending on the average degree $\langle k\rangle$ is low for a larger number of nodes. If a self-loop/multiegde was generated, it is simply dropped/combined.

The clustering coefficient of the Configuration Model graph is given with

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

2.4 Barabasi-Albert Model

The Barabasi-Albert model belongs to the class of **generative network models**. These types of models explore hypothesized generative mechanisms to see what graph structures they produce. If these structures are similar to real networks, it suggests that the hypothesized mechanisms may also be at work in these networks.

2.4.1 Power-Law Distribution

The *power-law distribution* is a degree distribution that often occurs in empirical networks. The statement "small is common" is represented by this distribution. A power-law distribution is described by the function

$$f(k) = ak^{-\gamma}$$

where k is the degree, $f(k)$ is the occurrence of degree k and $a, \gamma \in \mathbb{R}_+$ are variables. **??** shows a set of example power-law distributions.

The following algorithm describes how to generate a scale-free network (Barabasi-Albert model) with $m_0 + t$ nodes.

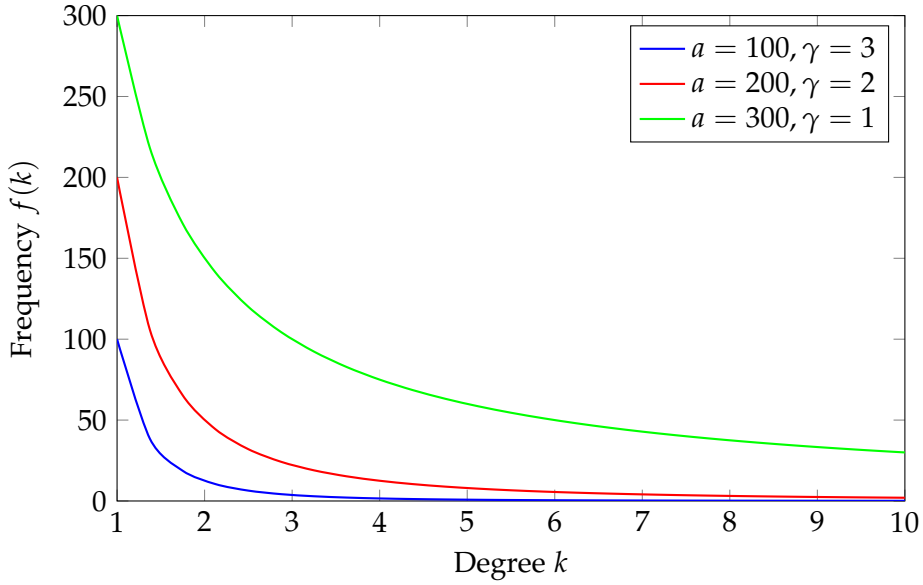


Figure 4: Example Power-Law Distributions

Algorithm: Generating Scale-Free Networks

1. Start with a graph $G = (V, E)$ where $m_0 = |V|$ and a given set of edges E.
2. Define a number $m \in \mathbb{N}$ that will be the degree for each additional node.
3. Calculate for each node $i \in V$ the probability $\pi(k_i)$ that a new node connects to i:

$$\pi(k_i) = \frac{k_i}{\sum_{j \in V} k_j}$$

   where k_i is the degree of node i.
4. Add a new node to the graph and add m edges from the new node to other nodes corresponding to the calculated probability.
5. Repeat step 3 and 4 until t nodes were added to the initial graph.

In the Barabasi-Albert model, nodes with higher degree are more likely to get new links. Therefore, nodes with a high degree will get an even higher degree and nodes with a low degree will stick at their low degree. This behavior causes a power-law degree distribution.

2.5 Comparison

In this section the presented network models are compared against each other and against the empirical network. See **??** and **??** for the comparison.

2.6 Robustness

Robustness measure the network's reliability of certain aspects when there are nodes (node attacks) or edges (edge attacks) deleted. E.g. when (after how many removals) does the giant component

Network Property	Erdős-Rényi	Configuration	Barabasi-Albert
degree distribution	Poisson($\langle k \rangle$)	specified	Power Law with $\gamma = 3$
diameter	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	
clustering coefficient	$\mathcal{O}(\frac{1}{n})$	$\mathcal{O}(\frac{1}{n})$	$\frac{(\ln \lvert N \rvert)^2}{\lvert N \rvert}$
reciprocity	$\mathcal{O}(\frac{1}{n})$	$\mathcal{O}(\frac{1}{n})$	
giant component	$\langle k \rangle > 1$	$\langle k^2 \rangle - 2\langle k \rangle > 0$	
average distance			$\frac{\log \lvert N \rvert}{\log \log \lvert N \rvert}$

Table 2: Comparison between network models

	Degree Distribution	Clustering	Diameter
Empirical Networks	Heavy tailed, heterogeneous	high	low
Erdős-Rényi	Poisson (homogeneous)	low	low
Watts-Strogatz	Nearly regular	high	low
Configuration	Specified	low	low

Table 3: Comparison of empirical networks and network models

split up into isolated components? Additionally, how does the selection of the nodes or edges influence reliability of a network?

Definition: Molloy-Reed Criterion

The Molloy-Reed criterion states that for any degree distribution a giant component exists if

$$\frac{\langle k^2 \rangle}{\langle k \rangle} \geq 2$$

where $\langle k \rangle$ is the average degree and $\langle k^2 \rangle$ is the average of the squared degrees.

Thereby, the critical threshold where the network breaks apart is given with

$$f_c = 1 - \frac{1}{\frac{\langle k^2 \rangle}{\langle k \rangle} - 1}.$$

Real networks are robust to failures or random attacks, but are vulnerable to targeted attacks.

3 Mesoscopic Structures

This section deals with structures at mesoscopic scale (which is somewhere between microscopic and macroscopic scale).

3.1 Communities

In a network a group of nodes is called community if they are more likely to connect to each other than to nodes of other communities. Furthermore, communities are described by the following three hypotheses:

H1 *Fundamental Hypothesis*
A network's community structure is uniquely encoded in its wiring diagram.
H2 *Connectedness Hypothesis*
A community corresponds to a connected subgraph.
H3 *Density Hypthesis*
Communities are locally dense subgraphs.

There are different ways of defining communities. The **first approach** would be defining communities by *cliques*. The problem with this approach is that small cliques are frequent and large cliques are rare. The **second approach** splits up the degree of each node i into k_i^{int} (internal degree), number of connections to the same community and k_i^{ext} (external degree), number of connections to other communities.

- A community $\mathcal{C}$ is a *strong community* if for each node i in $\mathcal{C}$

$$k_i^{\text{int}} > k_i^{\text{ext}}.$$

- A community $\mathcal{C}$ is a *weak community* if its total internal degree exceeds its total external degree:

$$\sum_{i \in \mathcal{C}} k_i^{\text{int}} > \sum_{i \in \mathcal{C}} k_i^{\text{ext}}.$$

3.1.1 Hierarchical Clustering

Exact community detection by considering all possibilities is computationally infeasible for larger networks. Therefore, we use algorithms that predict the communities using heuristics.

Algorithm: Ravasz - Agglomerative hierarchical clustering

Given an undirected network $G = (V, E)$.

1. Define the similarity matrix $S \in \mathbb{R}_{\geq 0}^{|V| \times |V|}$ that gives the similarity between each node $i, j \in V$ with entries:

$$s_{ij} = \frac{J(i,j) + \Theta(A_{ij})}{\min(k_i, k_j)}$$

where $J(i,j)$ gives the number of common neighbors of node i and j, $\Theta(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{otherwise} \end{cases}$ is the heaviside step function, A is the adjacency matrix of graph G and k_i is the degree of node i.
2. Create a community for each node itself.
3. Merge communities with highest average similarity.
4. Repeat step 1 for the new communities and merge them again (step 2 & 3) until all nodes form a single community.
5. Calculating a dendrogram visualizes the order in which the nodes are assigned to specific communities. To identify the communities we must cut the dendrogram. Hierarchical clustering does not tell us where that cut should be.

The time complexity of the algorithm is in $\mathcal{O}(|V|^2)$.

The Girvan-Newman algorithm is a divisive hierarchical clustering algorithm. It starts deleting edges that connect different communities until all edges are deleted. For choosing these edges the betweenness centrality can be used. Thereby, it is possible to cut the network into different communities. The time complexity of this algorithm for a graph $G = (V, E)$ is in $\mathcal{O}(|E|^2 \cdot |V|)$ (or for sparse networks in $\mathcal{O}(|V|^3)$).

3.1.2 Modularity

Modularity is an approach of rating a given partition that is based on the assumption that there are no community structures in randomly wired networks.

Definition: Modularity

Given a graph $G = (N, L)$ and community subsets $\mathcal{C}_1, \ldots, \mathcal{C}_n \subseteq N$ such that $\bigcup_{i=1}^{n} \mathcal{C}_i = N$ and $\bigcap_{i=1}^{n} \mathcal{C}_i = \emptyset$. The modularity M of the given partition is given with

$$M = \sum_{i=1}^{n} \left[\frac{|L_i|}{|L|} - \left(\frac{k_i}{2|L|} \right)^2 \right]$$

where $|L_i|$ is the number of links of the subgraph of C_i, $|L|$ is the number of links in the original graph and k_i is the summed degree of all nodes in C_i.

Higher modularity values indicate better partitions. A modularity value of 0 indicates that there is a single community. Negative modularity values are also possible.

Algorithm: Greedy Modularity Maximization

1. Assign each node to a community of its own.
2. Inspect each pair of communities connected by at least one link and compute the modularity variation ΔM obtained if we merge these two communities.
3. Identify the community pairs for which ΔM is the largest and merge them. Note that modularity of a particular partition is always calculated from the full topology of the network.
4. Repeat step 2 until all nodes are merged into a single community.
5. Record for each step and select the partition for which the modularity is maximal.

The problem with modularity is if there are two communities A and B with total degree k_A and k_B the resolution limit of modularity is given with $k_A \sim k_B = k \leq \sqrt{2L}$. Modularity cannot detect communities smaller than this size.

A faster way to compute communities using modularity is the Louvain algorithm that detects communities in two steps that are iteratively repeated.

Algorithm: Louvain Method

Given a (weighted) network $G = (N, L)$.

1. Assign each node to a different community. For each node $i \in N$ calculate the gain in modularity $\Delta M_{i,C}$ if we place node i in one of its neighboring communities C. We move node i to the community with the largest positive gain. This process is applied to all nodes until no improvement is achieved. The gain in modularity $\Delta M_{i,C}$ is given with

$$\Delta M_{i,C} = \left[\frac{\sum_{in} + 2k_{i,in}}{2W} - \left(\frac{\sum_{tot} + k_i}{2W}\right)^2\right] - \left[\frac{\sum_{in}}{2W} - \left(\frac{\sum_{tot}}{2W}\right)^2 - \left(\frac{k_i}{2W}\right)^2\right]$$

   where $\sum_{in}$ is the sum of (weighted) links inside of C, $\sum_{tot}$ is the sum of all (weighted) links of nodes in C, k_i is the sum of (weighted) links of node i, $k_{i,in}$ is the sum of (weighted) links from i to nodes in C and W is the sum of all (weighted) links in the network.
2. Construct a new network whose nodes are communities identified in step 1. The weight of a link between two nodes is the sum of (weighted) links between the corresponding communities. Links within a community lead to weighted self-loops.

Repeat both steps until no more changes are applied.

3.1.3 Infomap

The idea of *infomaps* is to have a walker that walks randomly through the network. Since edges within a community are more probable than edges between communities the walker should get stuck within communities (at least for a while). Thereby, it is possible to draw conclusions of the community structure. To make this method even more efficient it is possible to make use of *Huffman codes*.

3.1.4 Label Propagation

Label propagation is a very fast and local method to detect communities within a network.

Algorithm: Label Propagation

1. Initialize each node to have its own label.
2. For each node (randomly ordered) adopt the most popular label among its neighbors. Ties are settled randomly.
3. Repeat step 2 until no more changes are applied.

3.1.5 Rand Index

The Rand Index $\mathcal{R}$ is able to measure the performance of a community detection if the actual (ground truth) communities are known.

Definition: Rand Index

$$\mathcal{R} = \frac{n_{00} + n_{11}}{n_{00} + n_{11} + n_{10} + n_{01}}$$

- n_{11} Number of pairs of elements in the same community under both D and T
- n_{00} Number of pairs of elements not in the same community under both D and T
- n_{01} Number of pairs of elements not in the same community under D but in the same community under T
- n_{10} Number of pairs of elements in the same community under D but not in the same community under T

where T is the ground truth and D is the detected solution.

3.2 Homophily and Assortativity

A network is *homophilic* (or *heterophilic*) if its nodes are more (less) likely to connect to a node with a similar value of a given property. The homophily of two nodes i and j with a given property x_i and x_j can be measured using the covariance.

Definition: Covariance

The *covariance* will be high if x_i and x_j at either end of an edge tend to be both high or both low. Given a graph $G = (N, L)$, the covariance is given with:

$$\text{cov}(x_i, x_j) = \frac{1}{2|L|} \sum_{i,j \in N} \left(A_{ij} - \frac{k_i k_j}{2|L|} \right) x_i x_j$$

After normalization of the covariance we obtain the *Pearson correlation*.

Definition: Pearson Correlation

The *Pearson correlation* considers scalar observations over all edges of a network $G = (N, L)$:

$$r = \frac{\sum_{ij}(A_{ij} - \frac{k_i k_j}{2|L|}) x_i x_j}{\sum_{ij}(k_i \delta_{ij} - \frac{k_i k_j}{2|L|}) x_i x_j}$$

3.2.1 Degree Assortativity

When considering homophily with respect to the scalar degree value it is called degree assortativity. So, the Pearson correlation for degree assortativity is given with

$$r = \frac{\sum_{ij}(A_{ij} - \frac{k_i k_j}{2|L|}) k_i k_j}{\sum_{ij}(k_i \delta_{ij} - \frac{k_i k_j}{2|L|}) k_i k_j}$$

where $x_i = k_i$.

This can be simplified by iterating over the edges instead of iterating over the nodes twice. Given an undirected graph $G = (N, L)$ the Pearson correlation can be calculated using

$$r = \frac{S_1 S_e - S_2^2}{S_1 S_3 - S_2^2}$$

where

$$S_e = 2 \sum_{\{i,j\} \in L} k_i k_j \qquad S_1 = \sum_{i \in N} k_i \qquad S_2 = \sum_{i \in N} k_i^2 \qquad S_3 = \sum_{i \in N} k_i^3.$$

Assortativity has a direct impact on the network's structure. In high assortativity networks hubs tend to connect to hubs and low degree nodes tend to connect to low degree nodes. A natural *core-periphery* structure emerges. In networks with low degree assortativity ($r < 0$) hubs tend to connect to low degree nodes. Erdős-Rényi and Barabasi-Albert networks are neutral and have degree assortativity close to 0.

The assortativity of a given network $G = (N, L)$ can be adjusted by specific rewiring of the edges.

Algorithm: Xalvi-Brunet and Sokolov (XBS)

1. Choose two random links and name their adjacent nodes a, b, c, d such that $k_a \geq k_b \geq k_c \geq k_d$ where k_i is the degree of node i.
2. Break the links and rewire depending on the goal:
   - *Increase assortativity*
     Join a with b and c with d.
   - *Increase disassortativity*
     Join a with d and b with c.

Repeat step 1 & 2 and stop after a certain number of non changing attempts.

To moderate the effect the **tuned XBS** model only does an assortative (disassortative) rewiring with probability p. Otherwise the rewiring is random.

3.2.2 Degree Correlation

Given a graph $G = (N, L)$. For each node $i \in N$ the average degree of its neighbors is calculated:

$$k_{nn}(i) = \frac{1}{k_i} \sum_{j \in N} A_{ij} k_j$$

We define $N_k = \{n \in N | k_n = k\} \subseteq N$ the subset of nodes with degree k. The *degree correlation function* is the average k_{nn} for all nodes of degree k. Thus, it is given with

$$k_{nn}(k) = \frac{1}{|N_k|} \sum_{i \in N_k} k_{nn}(i).$$

3.2.3 Stochastic Block Models

Block models allow to generate networks with communities.

Definition: Stochastic Block Model (SBM)

A stochastic block model generates a network $G = (N, L)$ and is defined as tuple (k, z, M) where

- $k \in \mathbb{N}$ is the numbers of communities in the resulting network
- $z \in \{1, \ldots, k\}^{|N|}$ indexing the nodes into communities
- $M \in [0,1]^{k \times k}$ where M_{uv} denotes the probability that a vertex of group u connects to a vertex in group v.

A stochastic block model can now be generated by adding edges between i and j corresponding to the probability $M_{z_i z_j}$ where $z_i \in \{1, \ldots, k\}$ is the community number of node i.

The probability of a graph $G = (V, E)$ given a node labeling $z \in \{1, \ldots, k\}^{|V|}$ and a block matrix M is

$$P(G|z, M) = \underbrace{\prod_{\{i,j\} \in E} M_{z_i z_j}}_{\text{edge}} \underbrace{\prod_{\{i,j\} \notin E} (1 - M_{z_i z_j})}_{\text{non edge}}$$

4 Advanced Network Structure

4.1 Signed Networks

In signed networks each edge has either a positive sign or a negative sign. To evaluate these networks we can take a look at the triangles. For triangles there are four different configurations shown in **??**. **Structural balance theory** splits these configuration up into balanced and imbalanced/stressed

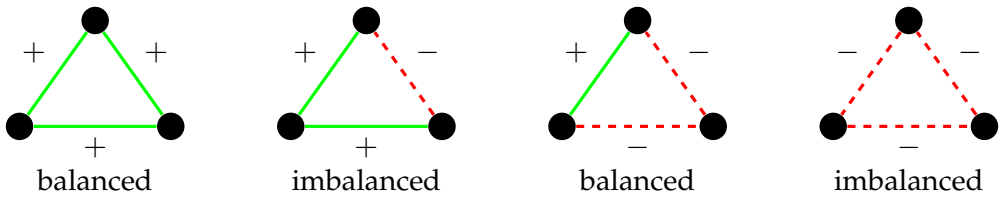


Figure 5: Four possible configurations of signed triangles (strong formulation)

configurations. A network is balanced if all of its triangles are balanced.

Theorem: Harary's Balance Theorem

A **complete** signed network is balanced only if it has two groups of nodes, where all nodes in each group have positive ties only, and all the ties between the groups are negative. One of the groups can be empty.

In the weak formulation of balance, triangles with negative edges only are also declared as *balanced*.

Theorem: Weak Structural Balance

A **complete** signed network is balanced if and only if it has an arbitrary number of groups of nodes where all edges within groups are labeled with + and all edges between groups are negative.

If there is an imbalanced triangle in a signed network, it can be an indicator that the network will **tend to change** since it is currently out of equilibrium. In **??** an example for this behavior is shown. Here is shown that if a married couple is divorced it is likely that a common friend will only keep

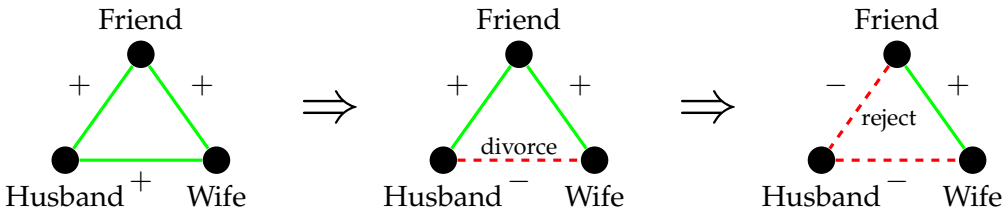


Figure 6: Example for Balance Theorem

friendship with either the husband or the wife. This concept can also be applied to larger scale. For example, the alliances formed before and during World War I followed this concept.

Comparing empirical signed networks with random signed networks reveals that there are much more balanced triangles and much less imbalanced triangles in the empirical network than in the random network. This fact suggests that *structural balance theory* does indeed describe signed networks.

4.2 Directed Networks

Directed networks contain more information than their undirected counterpoints which is why they are more complex. At dyadic (pairs of nodes) level directed networks allow four different configuration instead of two. There is one configuration for a null dyad (no connection between two nodes). There are two configurations for asymmetric dyads ($n_i \to n_j$ or $n_i \leftarrow n_j$). The last configuration is the mutual dyad ($n_i \leftrightarrow n_j$).

Definition: Dyadic Census

Measuring the number of mutual, asymmetric and null dyads is called *dyadic census*. Given a directed graph $G = (N, L)$ with the adjacency matrix X the dyads are defined as

$$\begin{aligned} \text{Mutual:} \quad M &= \sum_{\substack{i,j \in N \\ i<j}} X_{ij} X_{ji} \\ \text{Asymmetric:} \quad A &= |L| - 2M \\ \text{Null:} \quad &\binom{n}{2} - A - M \end{aligned}$$

Thereby, a simple reciprocity value r can be calculated with $r = \frac{2M}{|L|}$ where $r = 0$ implies no reciprocated edges and $r = 1$ implies that every edge is bidirectional.

Definition: Garlaschelli and Loffredo Reciprocity

Given a network $G = (N, L)$ with the adjacency matrix X the Garlaschelli and Loffredo reciprocity is defined as

$$\rho = \frac{\sum_{i,j \in N, i \neq j} (X_{ij} - X')(X_{ji} - X')}{\sum_{i,j \in N, i \neq j} (X_{ij} - X')^2}$$

where $X' = \frac{|L|}{|N|(|N|-1)}$ denotes the ratio of observed to possible links. It can be simplified to:

$$\rho = \frac{r - X'}{1 - X'}$$

It is $\rho \in [-1, 1]$ where 1 is perfect reciprocity, -1 is no reciprocity and 0 is the expected value of reciprocity of a random network.

Considering a network of three nodes. Between each pair of nodes we could place two different directed edges ending up with a total of 6 different edges in the network. So, there are $2^6 = 64$ different **realizations**. Some of these realizations are **isomorphic** with respect to swapping the node labels. The isomorphic realizations are structurally indistinguishable from each other.

There are 16 isomorphic classes of triangles in directed networks called *motifs*. A triangle is called closed if there exists at least one edge between each pair of nodes.

Definition: Triadic Census

To identify each of the isomorphic classes they can be labeled by four characters:

- Number of mutual dyads,
- Number of asymmetric dyads,
- Number of null dyads,
- Optional Character: D (for down), U (for up), T (for transitive), C (for cyclic).

The triadic census is now a tuple $T \in \mathbb{N}_0^{16}$ counting the occurrence of each motif.

The transition from an "open" triangle to a "closed" triangle is called **triadic closure**.

Definition: Triadic Census z-Score

The z-score indicates how much the occurrence of the motifs varies from randomized networks:

$$z = \frac{x - E(x)}{\sigma_x}$$

where $E(x)$ is the expected number of occurs of motif x in the null model[a], calculated as average of 1000 different realizations of the null model. σ_x is the standard derivation of the occurs of the motif x over the realizations.

[a] A null model of a network is a modification of the original network where arbitrary many edges were randomly swapped.

4.3 Temporal Networks

Temporal networks are used to describe a networks that may change over time. Edge in a temporal network have timestamps indicating at which point in time they exist.

Definition: Temporal Network

A temporal network T defined by $T = (V, S)$ where

- V is the set of nodes,
- S is a function mapping pairs of nodes to sets of timestamps:

$$S(u, v) = \{t_1, \ldots, t_n\}$$

indicating that there are edges at times $t_1, \ldots, t_n$ from node $u \in V$ to node $v \in V$.

In temporal network we can define a new kind of path that respects the given timing.

Definition: Time-Respecting Path

Given a temporal network $G = (V, S)$. A time-respecting path from $i_0 \in V$ to $i_n \in V$ is defined as sequence

$$(i_0, i_1, t_1), (i_1, i_2, t_2), \ldots, (i_{n-1}, i_n, t_n)$$

where $t_1 < t_2 < \cdots < t_n$ and $t_k \in S(i_{k-1}, i_k)$ for all $k \in \{1, \ldots, n\}$.

The definition of time-respecting paths creates a new type of metric next to shortest paths. **Fastest paths** consider the minimum time that is necessary to reach a certain node.

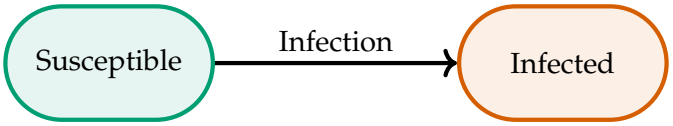
5 Dynamics and Spreading on Networks

5.1 Compartmental Models

There are models of different detail that help to understand the spreading of infectious diseases. Each model has different compartments which individuals can switch between. For the following section, remember that all models are wrong, but some are useful.

5.1.1 SI-Model

The SI-model is the simplest of the epidemic models. The model adds one of the two compartments *susceptible* and *infected* to each individual. Infected individuals spread the pathogens with probability β to susceptible neighbors. Once an individual is infected it stays infected.



Each individual has $\langle k \rangle$ contacts on average which get infected with probability β. It holds

$$\frac{\delta i}{\delta t} = \beta \langle k \rangle si \quad \text{and} \quad \frac{\delta s}{\delta t} = -\beta \langle k \rangle si$$

where s is the fraction of susceptible and i the fraction of susceptible population.

By integration we get the fraction of infected population i after t time steps:

$$i = \frac{i_0 e^{\beta \langle k \rangle t}}{1 - i_0 + i_0 e^{\beta \langle k \rangle t}}$$

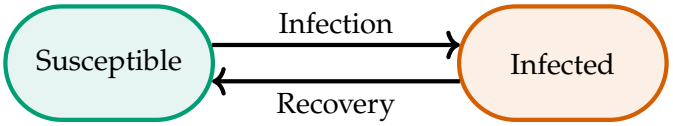
where i_0 is the fraction of infected population at time step 0.

The **characteristic time** τ is the time required to reach $\frac{1}{e} \approx 36.8\%$ of the susceptible individuals. For the SI-model it is

$$\tau = \frac{\langle k \rangle}{\beta(\langle k^2 \rangle - \langle k \rangle)}.$$

5.1.2 SIS-Model

Similar to the SI-model, the SIS-model has the two compartments susceptible and infected. In the SIS-model infected individuals can recover and thereby switch back to the susceptible state.



Additional to the SI-model, infected individuals recover at a fixed rate μ. It holds

$$\frac{\delta i}{\delta t} = \beta \langle k \rangle si - \mu i$$

where s is the fraction of susceptible and i the fraction of infected population and μi is term that captures the population that recovers from the disease.

The fraction of infected population i after t time steps is given with

$$i = \left(1 - \frac{\mu}{\beta \langle k \rangle}\right) \frac{Ce^{(\beta \langle k \rangle - \mu)t}}{1 + Ce^{(\beta \langle k \rangle - \mu)t}}$$

where the initial condition i_0 gives $C = \frac{i_0}{(1 - i_0 - \frac{\mu}{\beta \langle k \rangle})}$.

In the SIS-model the epidemic has two possible outcomes based on the μ and $\beta \langle k \rangle$. If $\mu < \beta \langle k \rangle$ the model results in a **endemic state** where $i(\infty) = 1 - \frac{\mu}{\beta \langle k \rangle}$ of the population will be infected in the long term. If $\mu > \beta \langle k \rangle$ the model results in a **disease-free state** where the initial infection dies out quickly since the number of recovering individuals exceeds the number of infected individuals.

The **basic reproduction number** R_0 is given with $\frac{\beta \langle k \rangle}{\mu}$. If $R_0 > 1$, the epidemic results in the endemic state.

The characteristic time for the SIS-model is given with

$$\tau = \frac{\langle k \rangle}{\beta(\langle k^2 \rangle - \mu \langle k \rangle)}.$$

5.1.3 SIR-Model

In the SIR-model infected individuals are removed after some time (because of immunity or death) instead returning to the susceptible state. Thereby, the whole population is removed in the long term.

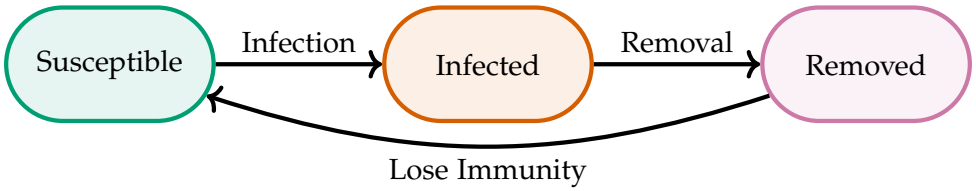
Infected individuals are removed at a fixed rate γ:

$$\frac{\delta i}{\delta t} = \beta\langle k\rangle si - \gamma i, \quad \frac{\delta s}{\delta t} = -\beta\langle k\rangle si, \quad \frac{\delta r}{\delta t} = \gamma i$$

where s is the fraction of susceptible, i the fraction of infected and r the fraction of removed population.

5.1.4 SIRS-Model

In the SIRS-model we also consider that the individuals lose their immunity after some time. So, removed individuals switch back to the susceptible state.



Removed individuals lose their immunity at a fixed rate d:

$$\frac{\delta i}{\delta t} = \beta\langle k\rangle si - \gamma i, \quad \frac{\delta s}{\delta t} = dr - \beta\langle k\rangle si, \quad \frac{\delta r}{\delta t} = \gamma i - dr$$

where s is the fraction of susceptible, i the fraction of infected and r the fraction of removed population.

5.2 Epidemic Threshold & Vaccinating Strategies

To predict whether a pathogen persists in the population, the *spreading rate* λ is defined:

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

which only depends on the pathogen's transmission probability β and recovery rate μ. A pathogen is only able to spread if the epidemic threshold λ_c is exceeded.

For the SIS-model and a *random network* the epidemic threshold is

$$\lambda_c = \frac{1}{\langle k\rangle + 1}.$$

For the SIS-model and a *scale-free network* the epidemic threshold is

$$\lambda_c = \frac{\langle k \rangle}{\langle k^2 \rangle}.$$

For large networks $\langle k^2 \rangle$ diverges whereby the epidemic threshold λ_c vanishes. This means that even viruses with a low transmission probability may spread successfully in a scale-free network.

To slow down an epidemic, the network structure can be exploited using a good vaccination strategy. This is comparable to measuring the robustness against different attack strategies. In general, vaccinating nodes with high degree (hubs) is a good vaccinating strategy. Unfortunately, in real networks it is hard to identify the hubs. The friendship paradox helps choosing nodes to vaccinate.

Theorem: Friendship Paradox

On average, most people have fewer friends than their friends do.

The paradox suggests that vaccinating random neighbors might lead to a slowdown of the epidemic. Empirical studies confirmed that the *random neighbor vaccinating strategy* is able slow down the spread of a pathogen.

5.3 Threshold Models & Cascades

Today, many goods offer two kinds of value to the customer. The *intrinsic value* measures the good's value on its own with respect to the customer. The *network value* measures the value a customer gets by other users using the product. Examples for these network goods are social media or programming languages. The total value of a good with respect to customer x is defined as

$$p = r(x) \cdot f(z)$$

where $x \in [0, 1]$ represents all possible consumers, $r(x)$ is a continuous, decreasing function measuring x's value of the good in isolation, z is the fraction of other individuals using the good and $f(z)$ measures the benefit of others using the good.

The equilibrium price for consumer z is now given with

$$p_{\text{equ}} = r(z) \cdot f(z).$$

The previous analysis implicitly assumes that the network is homogeneous since it depends on the fraction of all users of a good. Actually, the neighbors of an individual are relevant.

There are two alternatives A and B and each individual can adopt either A or B. If two nodes linked by an edge both adopt A they get a payoff $a > 0$ and if both adopt B they get a payoff $b > 0$. If their choice differs they get no payoff. A node should adopt A if

$$pda \geq (1-p)db \quad \Leftrightarrow \quad p \geq \frac{b}{a+b}$$

where d is the number of neighbors, p is the share of neighbors adopting A and a, b are payoff for A and B.

Consider a network in which everyone is using B. Some initial adopters are attracted by A for some intrinsic reasons. A node will adopt A if the fraction of A neighbors exceeds the threshold $q = \frac{b}{a+b}$. If every node in the network adopts A the chain reaction is called *complete cascade*.

Definition: Densely Connected Group

A group of nodes is connected with density p if each node in the group has at least a fraction p of its neighbors in the group.

Consider a set of initial adopters of A with a threshold q:

(i) If the remaining network contains a densely connected group of density greater than $1 - q$ then the initial adopters will not cause a complete cascade.
(ii) When a set of initial adopters does not cause a complete cascade, the remaining network must contain a group of density greater than $1 - q$.

In the previous model of cascades everyone has the same payoffs and thereby the same threshold. The model can be extended to heterogeneous payoffs whereby every node has its own threshold. Heterogeneous threshold increase the complexity of densely connected groups blocking a complete cascade since there are nodes that are easy to ”convince” and nodes that are hard to ”convince”.