NETWORKS GOES TO SCHOOL

#1

for secondary schools

NET WORKS

NET
WORKS

NETWORKS is a 10-year research programme hosted by the University of Amsterdam (UvA), Eindhoven University of Technology (TU/e), Leiden University (UL), and the Center for Mathematics and Computer Science (CWI) in Amsterdam.

The work in this book is supported by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research (NWO) through Gravitation-grant NETWORKS-024.002.003.

TU/e CWI

CONTENTS

INTRODUCTION

Transportation, traffic, communication and energy networks form the backbone of our modern society. To deal with the uncertainty, variation, unpredictability, size and complexity inherent in these networks, we need to develop radically new ways of thinking. The ultimate goal is to build self-organizing and intelligent networks.

The NWO-funded Gravitation programme NETWORKS started in the Summer of 2014 and covers a broad range of topics dealing with stochastic and algorithmic aspects of networks. The aim of the programme is to address the pressing challenges posed by large-scale networks with the help of stochastics and algorithmics. The focus is on modelling, understanding, controlling and optimizing networks that are complex and highly volatile.

In April 2018 the first event “NETWORKS goes to school” was organised by Marta Maggioni (Leiden University) and Nicos Starreveld (University of Amsterdam), by an initiative of Bart Groeneveld (University of Amsterdam). This book collects the material prepared for that event.

The book is intended for both secondary school students and teachers. It can be used both for self study and as a textbook for preparing lectures on networks. Chapter 1 is an introduction to the mathematics behind networks. It is a short course into networks theory, with several examples and exercises to practise the material. Chapter 2 presents three research articles on networks which can be read independently of each other; it is an introduction to some applications of networks theory. Chapter 3 exhibits some of the topics studied within NETWORKS; each section is based on a talk given at the “NETWORKS goes to school” event. Chapters 2 and 3 can be read independently of each other. The required background is given in Chapter 1.

The book has been written with the help of Khadija Lachhab.

The NETWORKS programme has an online platform that can be found at www.networkpages.nl, with various articles on networks. The Networks Pages aims to provide content suitable for people with all kinds of backgrounds and educational levels/experiences.

CHAPTER 1

Mathematical background

In this chapter we provide some background knowledge on the mathematical theory of graphs and networks. In Section 1.1 we formally define a graph, and discuss some basic concepts on graph theory, together with some examples. In Section 1.2 some basic structural properties of graphs are introduced and applied to real-world networks. Section 1.3 contains an overview of the material that will be needed in the sequel, such as trigonometric functions, complex numbers and probability distributions.

1.1. Graph theory

An intuitive definition of a network would be a 'collection of objects that are interconnected in some way'. For example a collection of people, who can be interconnected by friendships; or a collection of scientists, who can be interconnected by collaborations. To make this notion precise, we turn to graph theory.

DEFINITION 1.1.1. A **graph** is a pair $G = (V, E)$, where

- V is the set of nodes or vertices;
- E is the set of edges or branches, connecting the nodes.

Typically, we number the nodes from $\{1, 2, 3, \ldots\}$. We denote an edge between two nodes i and j by $\{i, j\}$. To define a graph, we can write down the sets V and E.

EXAMPLE 1.1.1. Consider

$$V = \{1, 2, 3, 4, 5, 6\}, \quad E = \{\{1, 2\}, \{1, 5\}, \{2, 3\}, \{2, 5\}, \{3, 4\}, \{4, 5\}, \{4, 6\}\}.$$

Then $G = (V, E)$ is a graph with six nodes and seven edges.

Sometimes it is useful to have a graphical representation of a graph. Typically we do this by drawing nodes as a circle with a label in it, and edges as a line between nodes. However, you are free to choose any representation you may like! In fact, the location of the nodes is also arbitrary, it only matters the way in which the edges connect the nodes together.

EXAMPLE 1.1.1 (Continued). In Figure 1.1.1 we see two ways in which the graph G can be drawn.

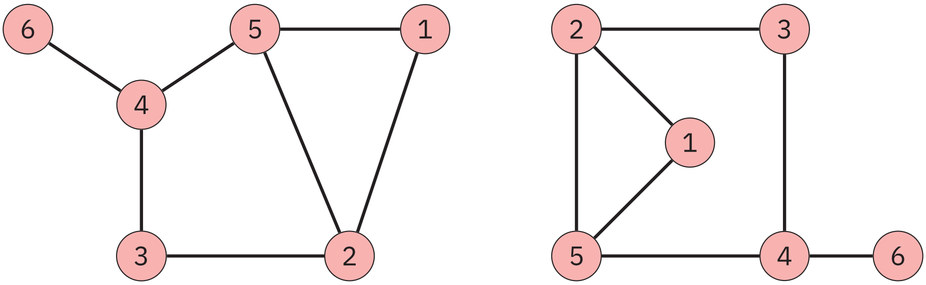


Figure 1.1.1. Two different representations of the graph in Example 1.1.1.

The degree of a node is a property that can quantify how 'important' the node is in the graph.

DEFINITION 1.1.2. The **degree** of a node is the number of edges that are connected to the node.

EXAMPLE 1.1.1 (Continued). The degrees of all nodes can be easily calculated, for example by using Figure 1.1.1. The degrees are shown in Table 1.1.1.

Node	1	2	3	4	5	6
Degree	2	3	2	3	3	1

Table 1.1.1. Degree of the nodes of the graph in Example 1.1.1.

Two special types of graphs are **trees** and **bipartite graphs**. To define a tree, we first have to define a cycle.

DEFINITION 1.1.3. A **cycle** C is a path over the edges in a graph, starting and ending in the same node. We denote cycles by the sequence of nodes travelled, i.e., $C = \{v_1, v_2, \dots, v_k, v_1\}$ starts and ends in node v_1 and travels over nodes $v_2, \dots, v_k$.

EXAMPLE 1.1.1 (Continued). The graph contains multiple cycles, for example

$$C = \{2, 3, 4, 5, 2\}.$$

DEFINITION 1.1.4. A **tree** is a graph without cycles.

EXAMPLE 1.1.2. Since the graph in Example 1.1.1 contains a cycle, it is not a tree. An example of a tree is given in Figure 1.1.2.

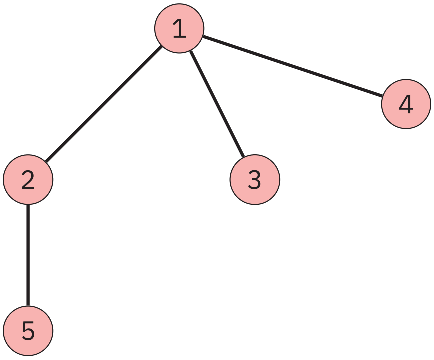


Figure 1.1.2. An example of a tree, a graph without any cycles.

DEFINITION 1.1.5. A **bipartite graph** is a graph in which the nodes can be divided into two sets V_1 and V_2, such that every edge connects a node in V_1 to a node in V_2. There are thus no edges between two nodes from V_1, or between two nodes from V_2.

We usually draw the bipartite graph by putting the nodes from V_1 on one side, and the nodes from V_2 on the other side.

EXAMPLE 1.1.3. Consider the bipartite graph in Figure 1.1.3. Then

$$V_1 = \{1, 2, 3\}, \quad V_2 = \{4, 5\}.$$

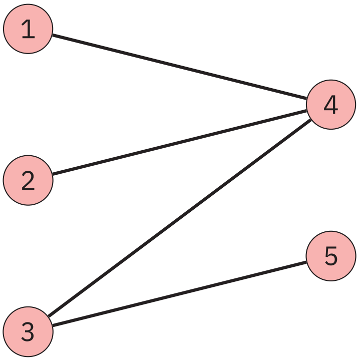


Figure 1.1.3. An example of a bipartite graph.

To represent a graph, we could also write down an **adjacency matrix** (see Section 1.3). In the adjacency matrix, we keep track of which connections are in the graph. If there is an edge between two nodes, the corresponding element in the matrix is 1. If there is no edge between two nodes, the corresponding element is 0.

DEFINITION 1.1.6. Let G be a graph with m nodes. The **adjacency matrix** A of G is an $m \times m$-matrix with elements

$$A_{ij} = \begin{cases} 1, & \{i,j\} \text{ is an edge,} \\ 0, & \{i,j\} \text{ is not an edge,} \end{cases} \qquad 1 \leq i, j \leq m.$$

EXAMPLE 1.1.4. Consider the graph in Figure 1.1.1. Then its adjacency matrix is

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

The degree of each node can be easily calculated using the adjacency matrix. Do you see how?

We can also look at the adjacency matrix in a different way, as the matrix of the number of paths of length 1 between nodes. Indeed, if there is an edge between two nodes, this is a path of length 1 between the nodes and the element in the matrix is equal to 1. But how do we know how many paths exist of length 2, or length 10? By multiplying adjacency matrices! More formally, to generate the matrix of paths of length k, denoted by A^k, we multiply the matrix of paths of length $k-1$ with the adjacency matrix.

EXAMPLE 1.1.4 (Continued). Consider again the graph in Figure 1.1.1. Then

$$A^2 = A \cdot A = \begin{bmatrix} 2 & 1 & 1 & 1 & 1 & 0 \\ 1 & 3 & 0 & 2 & 1 & 0 \\ 1 & 0 & 2 & 0 & 2 & 1 \\ 1 & 2 & 0 & 3 & 0 & 0 \\ 1 & 1 & 2 & 0 & 3 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \quad A^3 = A^2 \cdot A = \begin{bmatrix} 2 & 4 & 2 & 2 & 4 & 1 \\ 4 & 2 & 5 & 1 & 6 & 2 \\ 2 & 5 & 0 & 5 & 1 & 0 \\ 2 & 1 & 5 & 0 & 6 & 3 \\ 4 & 6 & 1 & 6 & 2 & 0 \\ 1 & 0 & 2 & 0 & 2 & 1 \end{bmatrix}.$$

For example, there are 2 paths of length 2 from node 1 to itself. Namely, the first path is traveling from node 1 to node 5 and back, the second path is traveling from node 1 to node 2 and back. In the same way, we see that there are 6 paths of length 3 from node 5 to node 2. Can you find them all?

Though not explicitly mentioned, so far we have considered *undirected* graphs. Undirected refers to the fact that an edge between two nodes can be traversed in both directions. We may also consider *directed* graphs, where we may only travel along edges in one direction. We then have directed edges, or *arcs*.

DEFINITION 1.1.7. A **directed graph** is a pair $G = (V, A)$, where

- V is the set of nodes or vertices;
- A is the set of directed edges or arcs, connecting the nodes. Each directed edge can only be traversed in one direction.

We now denote a directed edge between two vertices i and j by (i, j). Note that (j, i) does not have to exist! We typically draw directed edges as arrows, indicating the direction in which we can travel.

EXAMPLE 1.1.5. Consider

$$V = \{1, 2, 3, 4\}, \quad A = \{(1, 2), (2, 3), (3, 1), (4, 1)\}.$$

Then $G = (V, A)$ is a directed graph with four vertices and four edges. A representation of the graph is given in Figure 1.1.4. Note that we can travel from node 1 to node 2 in one step, but we need two steps to travel from node 2 to node 1.

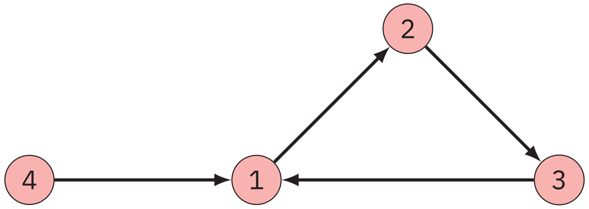


Figure 1.1.4. A representation of the graph in Example 1.1.5.

As we now have incoming and outgoing edges, we must slightly modify our notion of the degree of a node.

DEFINITION 1.1.8. The **indegree** of a node is the number of edges that enter the node. The **outdegree** of a node is the number of edges that leave the node.

EXAMPLE 1.1.5 (Continued). Using Figure 1.1.5, the indegrees and outdegrees of all nodes are easily determined and shown in Table 1.1.2.

Node	1	2	3	4
Indegree	2	1	1	0
Outdegree	1	1	1	1

Table 1.1.2. Indegree and outdegree of the nodes of the graph in Example 1.1.5.

DEFINITION 1.1.9. Let G be a directed graph with m nodes. The **adjacency matrix** A of G is a $m \times m$-matrix with elements

$$A_{ij} = \begin{cases} 1, & (i,j) \text{ is a directed edge,} \\ 0, & (i,j) \text{ is not a directed edge,} \end{cases} \qquad 1 \leq i,j \leq m.$$

Though this definition may seem equal to that of an undirected graph, the difference is evident in the resulting adjacency matrix. In an undirected graph, an edge travels in both directions. This means that the adjacency matrix of an undirected graph is *symmetric*, i.e., the element on row i and column j is equal to the element on row j and column i. In a directed graph, an edge only travels one way so the resulting matrix is not (necessarily) symmetric.

EXAMPLE 1.1.6. Consider the directed graph in Figure 1.1.4. Then its adjacency matrix is given by

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Note that this matrix is not symmetric, compare this with the adjacency matrix in Example 1.1.4.

As in the undirected case, we can determine paths of length k by multiplying the matrix of paths of length $k-1$ with the adjacency matrix. The only difference is that we now obtain the number of *directed* paths, which again can be travelled in only one direction.

EXAMPLE 1.1.6 (Continued). Consider again the directed graph in Figure 1.1.4. Then

$$A^2 = A \cdot A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad A^3 = A^2 \cdot A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Then for example, we can travel from node 1 to node 3 in two steps. We can also travel from node 4 to node 3 in three steps.

1.2. Networks

When we consider real-world networks, we usually talk about ‘complex’ networks. Complexity here means that, in general, real-world networks consist of a very large amount of nodes and edges, which are highly interconnected. Some examples of real-world networks are:

- **acquaintance networks**: in 1967, the psychologist Stanley Milgram performed an experiment in which he sent 60 letters to various people. They were asked to forward the letter to a specific person. The catch was that they could only pass the letter to one of their personal acquaintances, who then also had to pass it on to one of their acquaintances. At that time, only roughly 5% of the letters reached their destination. However, in later experiments, the success rate was increased to 35%, and even to 95% by using a high value package instead of a letter, or by providing more clues about the recipient. The main conclusion drawn was that most people are connected by a path of length at most 6. This number 6 occurs in other networks as well, for example in *Six Degrees of Kevin Bacon*.
- **social networks**: an example of a large and complex social network is the network of Facebook friendships. Each node is a Facebook user, and an edge connects two nodes if the corresponding users are friends on Facebook. Note that this is an undirected network, as Facebook friendships ‘travel’ in both directions. An example of a directed social network is the Twitter network. If we consider nodes to be Twitter users, then a directed edge from node i to node j would indicate that user i follows user j on Twitter. Since you do no automatically follow back, each edge can be travelled only in one direction.

- **transportation networks**: networks that represent roads, railway lines, or other forms of transportation quickly become increasingly complex. Consider for instance the Dutch railway network of NS, on which the main problems are efficiently determining the train time schedule and the allocation of material. We will study this in detail in Chapter 2, Section 2.3.

As networks are usually large and complex, it is difficult to obtain a global description of their structure. The examples of graphs we encountered in Section 1.1 were small and easily drawn for instance. Try this for a network with a billion nodes, and nobody will be happy with the result. In such complex networks it is important to develop methods that can tell us something about the structure of a network. In this booklet we will consider small-world, scale-free, modularity and treewidth. Research has shown that many real-world networks share the first two properties.

1.2.1. Small-world

The small-world phenomenon roughly states that distances in real-world networks are quite small. Consider for example the acquaintance network, in which it seemed that most people were connected by a path of length at most 6. Networks that have the small-world property are *highly connected*. This means that if we consider the largest component in the graph that is connected (so there exists a path between each pair of nodes in the component), then a large proportion of all nodes in the graph will be in that component. Many networks even consist of one connected component, for example the Internet.

Do you find it surprising that in many real-world networks, such as social networks, the path between two nodes is so short? It could be, as a social network often contains a lot of *closed triads*. In short, this means that since the friends of your friends are often also your friends, we encounter a lot of short circuits in the graph. But how is it then possible to reach all other people within just a few steps?

A mathematical model that exhibits both many closed triads and very short paths is the **Watts-Strogatz model**. The main idea is that everyone lives on an $n \times n$-grid, this grid may symbolize geographic proximity. For some constant value of r, assume that every node knows all other nodes within r grid steps. These are people you know because you live close to them (your neighbours, your classmates). For some other constant value k, assume that every node knows k other nodes uniformly at random across the grid. These are people you know who live far away, for a reason that has nothing to do with geographical proximity (your distant relatives, your friends you met on holiday).

Using the connections thus made, we can construct a *random graph*, i.e., a graph in which edges are generated according to some probability distribution. It is clear that this network has many closed triads, since most of your friends are close to you on the grid. Then they are also close to each other, and thus know one another according to the assumption. However, adding the k random connections helps us in establishing short paths to other nodes.

Since these k nodes are chosen at random, it is likely that they have very little overlap with your other friends. Travelling via a node k then gives you access to a whole other part of the grid. An example of the Watts-Strogatz model for $n = 4$, $r = 1$ and $k = 1$ is given in Figure 1.2.1. For a detailed exploration of the small-world phenomenon, see [4, Chapter 20].

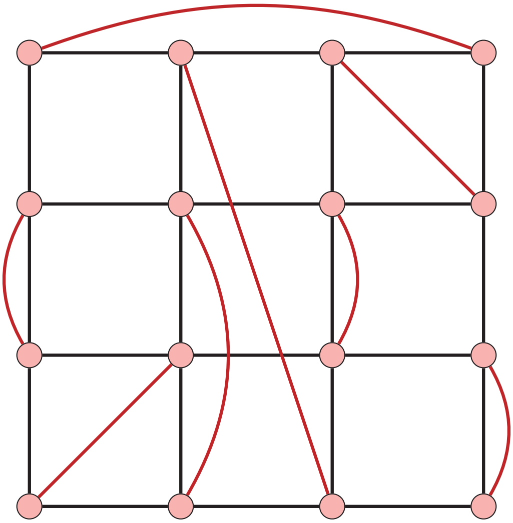


Figure 1.2.1. An example of the Watts-Strogatz model, with $n = 4$, $r = 1$ and $k = 1$. The red edges denote the connections made uniformly across the grid.

1.2.2. Scale-free

The scale-free phenomenon roughly states that the degrees in real-world networks show a lot of variability. Most nodes have just a few connections, but there are some nodes with a huge amount of connections. These popular nodes are called *hubs* and will be connected to most parts of the network.

In a scale-free network, the distribution of the degree of the nodes follows a *power law*. This means that, for some constant c and some exponent $\tau > 0$, the number of nodes of degree k is proportional to

$$f(k) = ck^{-\tau}.$$

Hence the larger k is, the fewer nodes we encounter with degree k. An example of a power law is given in Figure 1.2.2. Note the difference with the bell curve of the normal distribution, as in Figure 1.3.8 of the next section. In the bell curve, most nodes have a degree around the average with very few outliers. In the power law, most nodes have a very small degree but there are some nodes with very large degree.

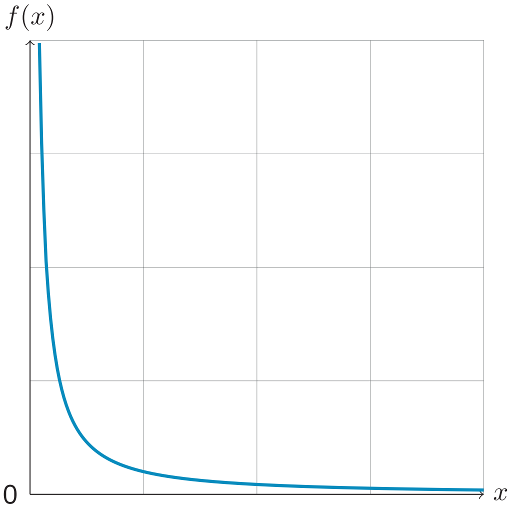


Figure 1.2.2. Partial plot of the power law $f(x) = \frac{1}{5}x^{-\frac{6}{5}}$.

How do we explain that many real-world networks exhibit the same scale-free behaviour? A possible explanation may be found by considering a mathematical graph model using the **preferential attachment** mechanism. The basic idea of such a model is as follows. We start with one node and at each point in time, we add another node to the network. The new node will choose an already existing node with which to connect, with probability *proportional to the degree of the existing node*.

There are two important things to note about this model. First, it is a *growing* network model. Compare this for instance with the Watss-Strogatz model, where we start with all nodes given and we determine which edges will connect them. In the preferential attachment model, we do not start with all the nodes given. Rather, we add nodes and their edges one at a time. Second, since the connecting probabilities depend on the degree of the existing node, nodes with a high degree in the network are more likely to gain new connections. This effect will eventually produce a few hubs in the network, with very large degree.

If we continue in this way, the distribution of the degree of the nodes will eventually follow a power law distribution. Is the preferential attachment model a good representation of real-world networks? One important thing to note is that the growing nature of the network is representative for most real-world networks. Consider for example the Hollywood network, where nodes are actors and edges are films that actors have collaborated on. As there are always more people getting into acting, the network is constantly growing. Another thing is that 'popular' nodes often are more likely to gain new connections. In the Hollywood network, new actors are more likely to work together with well-established actors, if not simply for the fact that they are in a large amount of movies.

How do the connections in the network change when certain nodes are removed? This is the main question behind the concept of *robustness*. The robustness of a network is the ability of the network to withstand failures and attacks, hence it measures the stability of a network. One way of measuring robustness is by considering the shortest paths, and whether or not the shortest paths change when we remove nodes. We can either remove nodes at random (which could happen due to random failures) or target specific nodes (which could happen in targeted attacks). Networks that are scale-free turn out to be quite resistant to removal of random nodes, as these are usually nodes with a low degree. However, the removal of a few key hubs could mean that a great many of the connections in the network fail. The shortest paths then also change significantly, as these usually travel via a key hub. Of course to do this, one must still determine which nodes are key hubs. For a detailed exploration of the notion of scale-free, see [2].

1.2.3. Modularity

Modularity is a measure of the structure of a network. It indicates whether there exist communities or groups in a network. The main idea behind it is that we compare the given network to a network with the same weighted degrees, but in which all edges are rewired at random. If there are more connections between sets of nodes in the original network than in the random network, this indicates some community structure.

Let G be a graph with m edges, where the number of edges between i and j is w_{ij}. We cut each edge into two halves which we call stubs, and each stub is randomly matched with another stub. We consider the quantity

$$Q = \frac{1}{2m}\sum_{k=1}^{l}\sum_{i,j\in C_k}\left(w_{ij} - \frac{k_i k_j}{2m}\right). \tag{1.2.1}$$

We divide the nodes into a total of l sets, which we call $C_1, \ldots, C_l$. These represent the (suspected) communities. Per set C_k, we then consider all vertices that are in the set. In the original network, the number of edges between i and j is w_{ij}. Let k_i be the degree of node i. Since the rewired network is *random*, we cannot speak about the absolute number of edges between i and j. Instead, we consider the expected number of edges between i and j which is equal to

$$\frac{k_i k_j}{2m}.$$

We thus see that if the difference between w_{ij} and this expected number is positive, then the connection in the original network is stronger than in the random network. To actually identify the communities in a network, we maximize Q over all possible sets C_k. The higher this number is, the stronger the communities are, i.e., the more connections they have within their community.

But where does the expected number of edges come from? To see this, consider the binomial distribution and the corresponding expected value (see Section 1.3). Let X_{ij} be the number of edges between i and j in the rewired network. Define success as the existence of an edge between i and j. Then X_{ij} has a binomial distribution with parameters:

- probability of success: $\frac{1}{2m}$;
- number of trials: $k_i k_j$.

The probability of success is the probability of connecting a stub of i to a stub of j. Since the stubs are randomly chosen, and there are a total of $2m$ stubs, the probability of choosing one particular stub is $1/(2m)$. The number of trials is the amount of opportunities we have to connect stubs from i to stubs from j. As the degree of node i is k_i, node i has k_i stubs. Similarly, node j has k_j stubs. The amount of opportunities are then the number of combinations of a stub from i and a stub from j, so a total of $k_i k_j$.

To determine the expected number of edges between i and j in the rewired network, we have to determine the expected value of X_{ij}. Since X_{ij} has a binomial distribution, it follows immediately that

$$\mathbb{E}[X_{ij}] = \frac{k_i k_j}{2m}.$$

1.2.4. Treewidth

In general, we consider tree graphs as 'simple' graphs. But how do we measure how simple a non-tree graph is? We do this by considering how tree-like a graph is, by using a concept called the *treewidth*. First we have to define a tree decomposition of a graph.

DEFINITION 1.2.1. A **tree decomposition** of a graph $G = (V, E)$ is a pair (T, X), where $X = \{X_t : t \in V(T)\}$ is a family of subsets of V and T is a tree whose nodes are the subsets X_i, such that

- for all edges $\{u, v\} \in E$ of the graph G, there exists a node t of the tree such that $u, v \in X_t$;
- for each pair y, z of nodes of T, if w is any node on the path between y and z then $X_y \cap X_z \subseteq X_w$.

This is quite a technical definition, so lets unpack step by step what is meant by a tree decomposition. Using the graph G, we will construct a tree T. Each node t of T is associated with a set X_t. This set contains nodes from G, in such a way that

- for every edge $\{u, v\}$ in G, there must be a set X_t in which both u and v are contained;
- if X_y and X_z both contain the node w, then for all nodes on the path between y and z, their respective sets X should contain w as well.

To make the concept more clear, we will look at an example.

EXAMPLE 1.2.1. Consider the graph in Figure 1.2.3. Several tree decompositions are shown in Figure 1.2.4. The left figure is a trivial decomposition and can be done for any graph. The other figures are two other decompositions for this graph. Can you check if the two requirements are true for these figures?

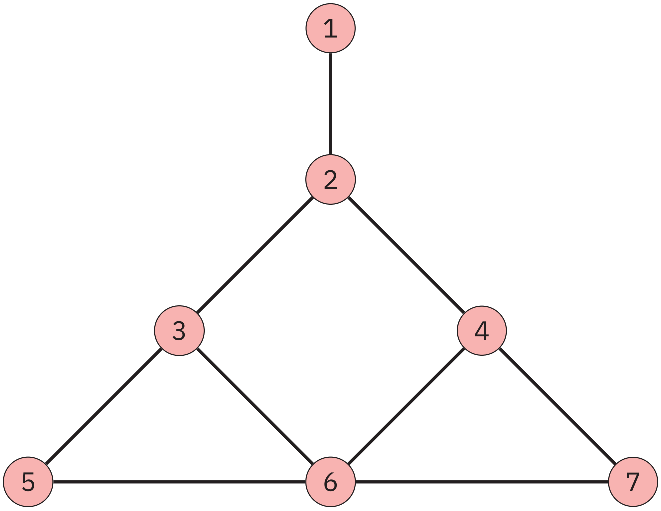


Figure 1.2.3. A graph without tree structure.

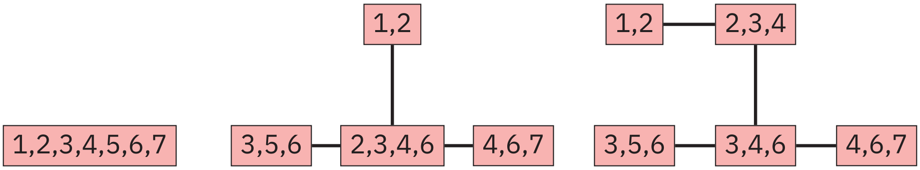


Figure 1.2.4. Three different tree decompositions of the graph in Figure 1.2.3.

We thus see that a tree decomposition is not unique, in fact, the trivial decomposition is valid for any graph. Then how will we use this notion to say something about how tree-like the original graph is, if we have multiple decompositions to choose from?

DEFINITION 1.2.2. The **width** of a tree decomposition is

$$\max\{|X_t| - 1 : t \in V(t)\},$$

i.e, the size of its largest set X_i minus one. The **treewidth** of a graph G is the minimum width of a tree decomposition of G, among all possible tree decompositions of G. We denote the treewidth of G by $tw(G)$.

EXAMPLE 1.2.1 (Continued). Consider the decompositions in Figure 1.2.4. Then its widths are 6, 3 and 2, respectively. For the treewidth $tw(G)$ we have to consider the minimum width among **all** possible tree decompositions, so it is possible to find a decomposition with width 1?

Finding the treewidth of a given graph has been proven to be equivalent to the game of **'Policemen catching robbers on a graph'**. We present an assignment derived from a presentation Bart Jansen gave in 2017, called *Micro-course on Structural Graph Parameters, part 1: Treewidth* [5].

Consider the following setup: a robber moves between nodes of a graph. The police chases him by using helicopters and tries to catch him by landing on a node. A police officer can catch the robber only if he lands on the node the robber is located. The robber moves to a neighbouring node, whenever there is no police officer there and he sees the helicopter coming to land on the node he is on. It can be shown that a graph G has treewidth at most $k-1$ if and only if k cops are sufficient to catch any robber on G.

Let's play this game on some specific graphs. Consider a tree, as in Figure 1.2.5. We see that robbers cannot hide on trees and two police officers are enough to catch any robber on such a graph.

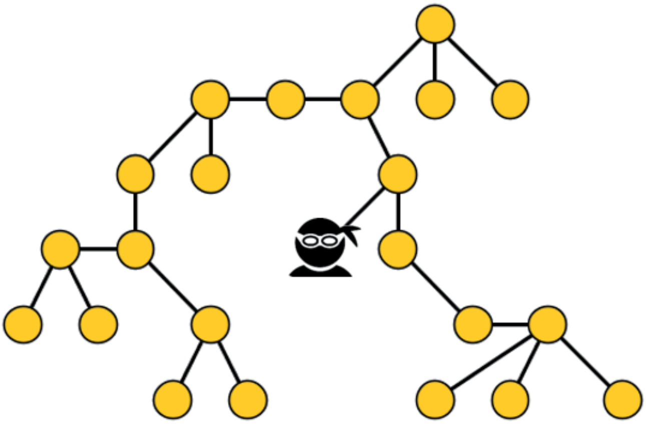


Figure 1.2.5. A robber moving on a tree.

EXERCISE 1. Consider the graphs given in Figure 1.2.6. How many police officers are needed in order to catch a robber moving on each one of these graphs?

We can also consider a real-world 'policemen catching robbers on a graph'-game. In the local police station in Leiden, two emergencies occur. There are two robberies, one in a jewellery store on the Middelweg and one in a small bank on the Uiterstegracht. The police station has to coordinate the police officers patrolling in the city, and send additional units to key points so that the robbers will not escape. Thankfully they have a group of mathem-

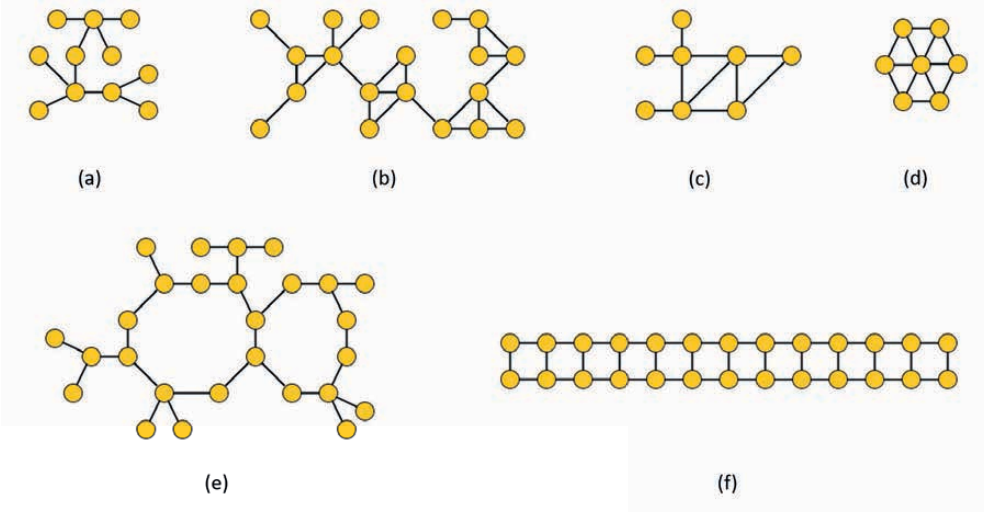


Figure 1.2.6. How may police officers are needed to catch a robber moving on these graphs?

aticians who can come up with a good strategy to catch the robbers. First, we take the map of Leiden and mark the locations of the robberies, see Figure 1.2.7.

We need to mathematically analyse this situation. We take the map of the centre of Leiden and we make a grid of all streets and the crossroads, see Figure 1.2.8. If we now erase the city map, in Figure 1.2.9 we obtain a graph which we can analyse using treewidth.

In these two maps, the green nodes denote bridges and the red nodes denote crossroads. We see that the bridges connect different parts of the city. Since both robberies occurred in the internal ring, we need to secure the bridges so that the robbers will not escape to the outer ring. Therefore we need to devise a police-robbers catching strategy only on the inner ring, and not on the whole city. The inner ring is a graph with 43 vertices, which we number. In Figure 1.2.10, the graph of the inner ring is depicted, as well as the locations of the robbers and the patrolling police officers.

There are currently five police officers patrolling in the centre, on nodes 5, 10, 18, 25 and 32. Additionally to them, we need to dispatch more units so that we can catch the robbers.

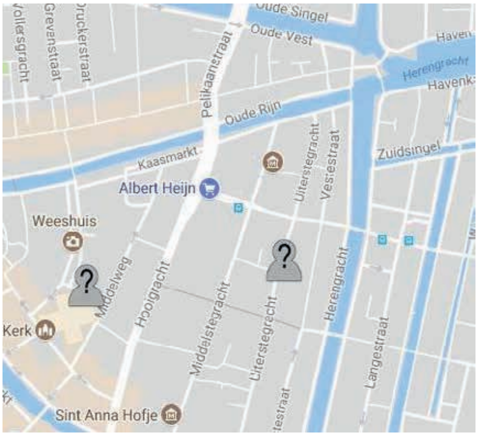


Figure 1.2.7. A map of the centre of Leiden and the location of the two robberies.

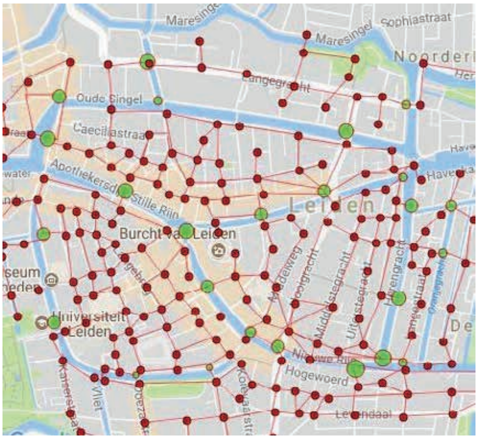


Figure 1.2.8. An artificial network overlapped on the map of the centre of Leiden.

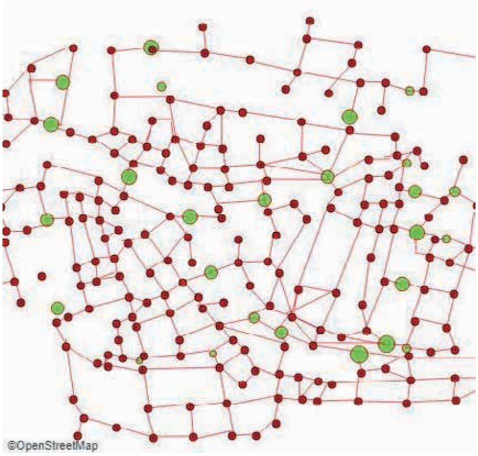


Figure 1.2.9. An artificial network of the centre of Leiden, which we can analyse.

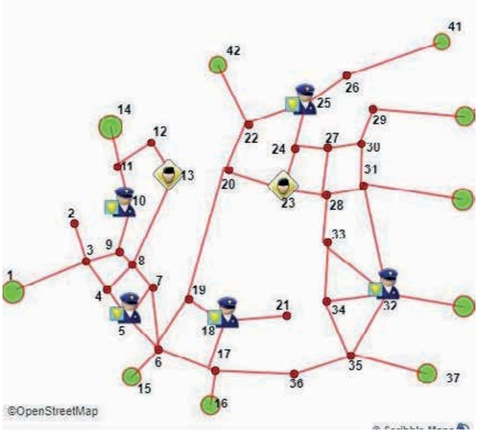


Figure 1.2.10. The graph of the inner ring, with the locations of the robbers and the patrolling police officers.

EXERCISE 2. Consider the graph in Figure 1.2.10.

(a) Are these five police officers capable of catching the two robbers? (Remember that a robber is caught if he is on a node, all the neighbouring nodes to that node are secured by policemen and there is a police officer free to move to the node where the robber is located.)

(b) Can we surely catch one of the two robbers? (If all five policemen focus on one robber?)

(c) How many additional officers do you need in order to catch the two robbers?

(d) How would you allocate the patrolling officers until the additional officers arrive?

1.3. Special functions

1.3.1. Indicator function

The indicator function of a subset A of a set X, denoted by $\mathbb{1}_A$, is a function from X to $\{0,1\}$. It returns the value 1 for all elements of X that are in A, and the value 0 for all elements of X not in A.

DEFINITION 1.3.1. The **indicator function** of a subset A of X is a function $\mathbb{1}_A : X \to \{0,1\}$ defined as

$$\mathbb{1}_A(x) = \begin{cases} 1, & x \text{ is in } A \\ 0, & x \text{ is not in } A. \end{cases}$$

EXAMPLE 1.3.1. Consider a set X and a subset $A \subseteq X$, see Figure 1.3.1. Then we have

$$\mathbb{1}_A(a) = 1, \quad \mathbb{1}_A(b) = 0.$$

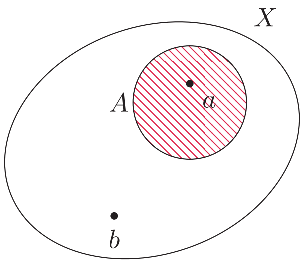


Figure 1.3.1. A set X with subset $A \subseteq X$.

1.3.2. Exponential functions and exponential growth

DEFINITION 1.3.2. An **exponential function** is a function of the form

$$y(x) = a \cdot b^x,$$

for fixed constants $a \neq 0$ and $b > 0$.

Exponential functions are characterized by the fact that the growth rate of the function is directly proportional to the value of the function. As it is then possible to have a very rapid growth rate, exponential functions can start from a low value and explode over time.
Any situation of exponential growth can be modelled as a relationship between a variable x growing at a rate $r > 1$ to the power of t, with x_0 the value at time zero:

$$x(t) = x_0 \cdot r^t.$$

EXAMPLE 1.3.2. In Figure 1.3.2, we show three examples of an exponential function with $x_0 = 1$ and $r = \frac{6}{5}$, $r = \frac{3}{2}$ and $r = 3$. The larger r is, the faster the function increases. What do you think would happen if $0 < r < 1$?

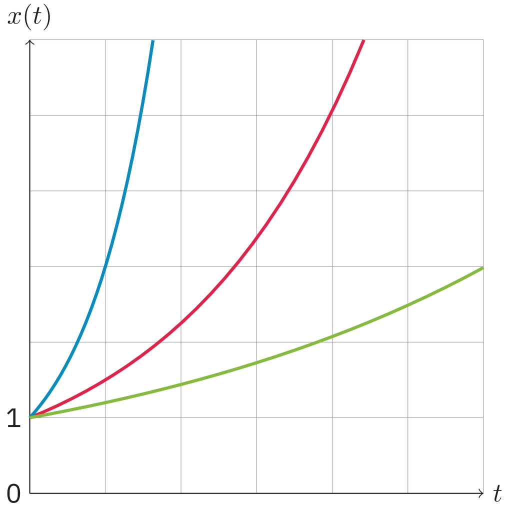


Figure 1.3.2. Partial plot of the exponential functions $x(t) = \left(\frac{6}{5}\right)^t$ in green, $x(t) = \left(\frac{3}{2}\right)^t$ in red and $x(t) = 3^t$ in blue.

In real life, we see plenty examples of exponential growth in time.

(1) **Australia's rabbit infestation**
In 1859, an English farmer named Thomas Austin brought 24 rabbits with him to his new home in Australia, and set them free on his land. Within six years, the total number of rabbits in Australia was 22 million.

(2) **Nuclear bomb**
In 1945, a group of physicist split an atom in the New Mexico desert. When they did, 2 new atoms split. After that, 4 atoms did, then 8, 16, 32 and so on, eventually producing the largest explosion then recorded.

(3) **Facebook**
In 2004, the social network Facebook was invented at Harvard. It was so popular that everyone who joined invited several of their friends, who then invited several of their friends. Now there are more than a billion people using it.

1.3.3. Trigonometric functions

Trigonometric functions are functions of an angle, that relate the angles of a triangle to the length of its sides. The most well-known trigonometric functions are the sine, cosine and tangent, denoted by $\sin x, \cos x$ and $\tan x$. We have that

$$\tan x = \frac{\sin x}{\cos x}.$$

A function f is called **periodic** if for some non-zero constant P, we have that $f(x) = f(x + P)$ for all values of x. Then P is called the **period** of the function. A periodic function thus repeats its values after each period.

A plot of the sine and cosine function is given in Figure 1.3.3 and a plot of the tangent function is given in Figure 1.3.4. As is clear from the figures, these functions are periodic. The sine and cosine function have a period of 2π, while the tangent function has a period of π. Since $\cos x = 0$ for $x = \pi n - \frac{\pi}{2}$, where n is any integer, we find that $\tan x$ has a *vertical asymptote* at these values, i.e., the function becomes arbitrarily large when x is close to the value. Furthermore, the functions have some symmetric properties:

$$\sin(-x) = -\sin x, \quad \cos(-x) = \cos x, \quad \tan(-x) = -\tan x.$$

Since the trigonometric functions are periodic, strictly speaking there does not exist a true inverse function. For example, $\sin(0) = 0 = \sin(\pi)$, so we do not find a unique element when we consider the inverse of 0. We can however define the inverse functions by restricting the domain. These are the functions $\arcsin x, \arccos x$ and $\arctan x$.

1.3.4. Polar forms of complex numbers and Euler's formula

DEFINITION 1.3.3. A **complex number** is a number of the form $a + ib$, where a and b are real numbers and i satisfies $i^2 = -1$. i is called the **imaginary number**. If $z = a + ib$, then

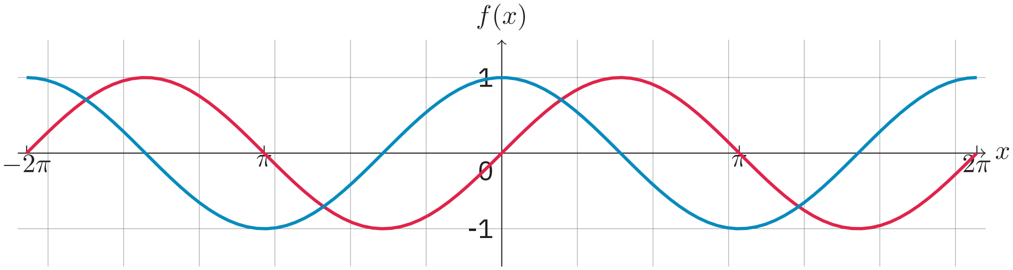


Figure 1.3.3. Partial plot of the trigonometric functions $f(x) = \sin x$ in red and $f(x) = \cos x$ in blue.

the real part of z is a, while the imaginary part of z is b. Two complex numbers are equal if their real parts are equal, and their imaginary parts are equal.

We can represent the complex number $z = a + ib$ in the two-dimensional plane. If we use Cartesian coordinates, we let the x-axis be real, the y-axis be imaginary and we view z as the point (a, b). The coordinates a and b then refer to how far the point lies along the x-axis and the y-axis, respectively.
We can also describe z by using **polar coordinates**. In this coordinate system, z is the point (r, θ), where

- r is the **radius**, the length of the ray connecting the origin to the point;
- θ is the **angle** between this ray and the positive x-axis.

In Figure 1.3.5, both coordinate systems are drawn in the two-dimensional plane. We can thus describe the same point z with either the Cartesian coordinates (a, b) or the polar coordinates (r, θ). To switch between the two systems, some handy formulas exist which have been given in Table 1.3.1.

Cartesian coordinates	Polar coordinates
(a, b)	$(\sqrt{a^2 + b^2}, \text{atan2}(b, a))$
$(r\cos\theta, r\sin\theta)$	(r, θ)

Table 1.3.1. Relation between Cartesian coordinates and polar coordinates.

In Table 1.3.1, atan2(b, a) refers to a generalisation of the inverse tangent function. It is given by

$$\text{atan2}(b, a) = \begin{cases} \arctan\left(\frac{b}{a}\right), & \text{if } a > 0, \\ \arctan\left(\frac{b}{a}\right) + \pi, & \text{if } a < 0, b \geq 0, \\ \arctan\left(\frac{b}{a}\right) - \pi, & \text{if } a < 0, b < 0. \end{cases}$$

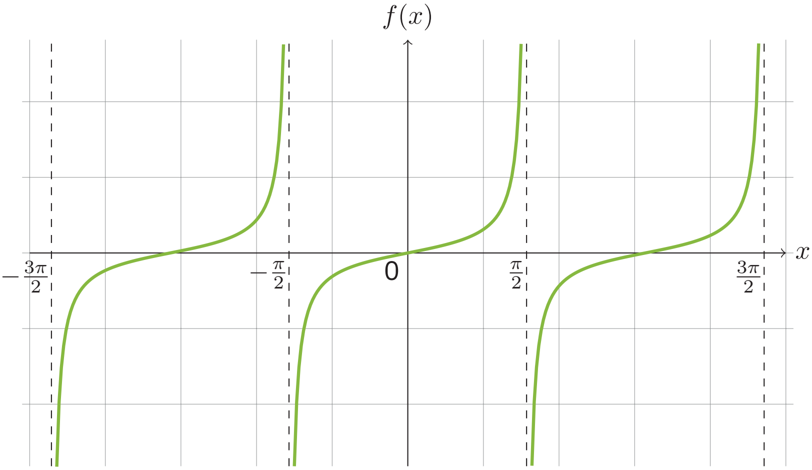


Figure 1.3.4. Partial plot of the trigonometric function $f(x) = \tan x$*. The vertical asymptotes are denoted with dotted lines.*

This distinction is necessary since $z = a - ib$ and $z = -a + ib$ are two complex numbers with *different* angles. However, for both we have that the quotient of the imaginary part over the real one gives the same result. If we would only consider the arctan function, this would produce the same angle. A plot of the arctan function is given in Figure 1.3.6.

An interesting relation occurs when we consider the complex exponential. Let e be Euler's number. Then **Euler's formula** states that for a real number x

$$e^{ix} = \cos x + i \sin x.$$

In Figure 1.3.7, we see this formula represented in a picture. In Chapter 3, Section 3.4 we present some exercises with Euler's formula. It is possible to derive the formula using **Taylor series**, which is a representation of a function as a series of power terms. A rough sketch of the proof is as follows.

We know that the exponential function has the following Taylor series:

$$e^x = 1 + x\frac{x^2}{2!} + \frac{x^3}{3!} + \dots.$$

Then we substitute ix for x, and use the fact that $i^2 = -1$ to obtain

$$e^{ix} = 1 + ix + \frac{(ix)^2}{2!} + \frac{(ix)^3}{3!} + \dots$$
$$e^{ix} = 1 + ix - \frac{x^2}{2!} - \frac{ix^3}{3!} + \dots.$$

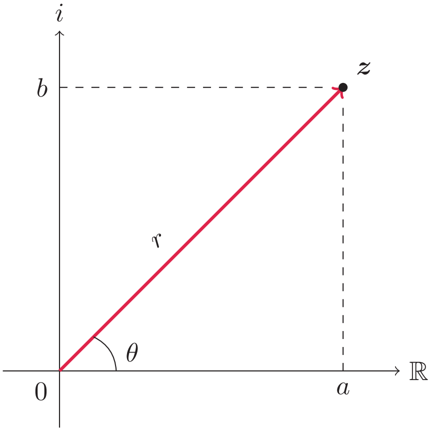


Figure 1.3.5. The complex number $z = a + ib$, using Cartesian and polar coordinates.

We then separate the real and the imaginary parts, and obtain

$$e^{ix} = \left(1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \dots\right) + i\left(x - \frac{x^3}{3!} + \frac{x^5}{5!} + \dots\right).$$

As it turns out, these are exactly the Taylor series for $\cos x$ and $\sin x$, respectively. We thus obtain Euler's formula

$$e^{ix} = \cos x + i \sin x.$$

1.3.5. Binomial distribution

The **binomial distribution** is a probability distribution on the number of successes in a series of experiments. We make the following assumptions:

- the number m of observations is fixed;
- each observation is independent of the other observations;
- each observation represents one of two outcomes: success or failure;
- the probability p of success is exactly the same for each trial.

Under these assumptions, we can describe each binomial distribution by using the parameters m and p. Summarizing, we perform an experiment m times. The outcome of each experiment is either success or failure, where success occurs with probability p. Each outcome is independent from all other outcomes.

A **random variable** X is a variable whose possible values are outcomes of a random process. We define a random variable by giving the *state space*, the set of all possible values, and by giving the probability distribution, the random process that dictates its outcomes. Let X be a random variable that follows the binomial distribution with parameters m and p.

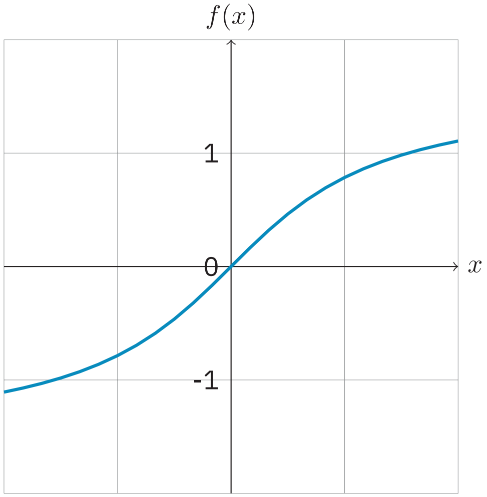


Figure 1.3.6. Partial plot of the function $f(x) = \text{arctan}(x)$.

Then X has state space $\{0, 1, \ldots, m\}$, and the probability that X is equal to k is given by

$$\mathbb{P}(X = k) = \binom{m}{k} p^k (1-p)^{m-k}$$

where

$$\binom{m}{k} = \frac{m!}{k!(m-k)!}$$

is the **binomial coefficient**. The symbol $\binom{m}{k}$ is read as 'm choose k', as this is the number of ways to choose k different elements from a total of m elements, where the order of elements does not matter. The factorial of m is denoted by $m!$ and equal to the product $m \cdot (m-1) \cdot (m-2) \cdot \ldots 1$.

Consider the following example. Suppose that we have a total of 5 colours, and we wish to know how many combinations there are of 3 different colours, where the order of the colours does not matter. Then $m = 5$ and $k = 3$, and

$$\binom{5}{3} = \frac{5!}{3!2!} = 10.$$

We could also reason in a different way. For the first choice we have a total of 5 possible colours, for the second choice we have 4 possible colours and for the third choice we have 3 possible colours. The total of combinations of three colours is then $5 \cdot 4 \cdot 3 = 5!/2!$. However, the order of colours does not matter, so we still have to divide by the number of ways in which we can order 3 colours, which is $3 \cdot 2 \cdot 1 = 3!$.

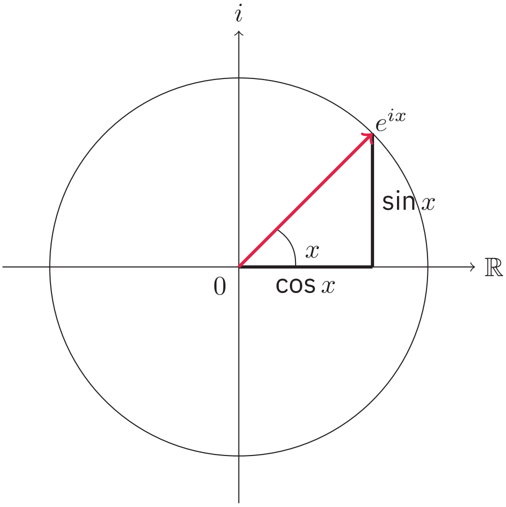


Figure 1.3.7. Graphical representation of Euler's formula.

EXAMPLE 1.3.3. Consider a coin toss, where possible outcomes are heads or tails. Suppose that we have a fair coin, i.e., the probability for heads is the same as it is for tails. If we toss the coin 10 times, then the number of heads has a binomial distribution with parameters $m = 10$ and $p = \frac{1}{2}$. The probability of getting four heads is equal to

$$\mathbb{P}(X = 4) = \binom{10}{4} \frac{1}{2}^4 \left(1 - \frac{1}{2}\right)^{10-4} = \frac{105}{512} \approx 0.205.$$

1.3.6. Normal distribution

The **normal distribution** or **Gaussian** distribution is a probability distribution that is used in many fields of science. The parameters for the normal distribution are the mean μ and the standard deviation σ. Here, μ quantifies the expected value of the data and σ quantifies how much variation there is in the data.

The binomial distribution is a **discrete** probability distribution. The notion discrete here refers to the fact that the state space is *countable*, i.e., each element of the set can be listed one at a time. Examples of a countable set are $\{-1, 5, 10, 2, 80\}$ or $\{0, 1, 2, \ldots\}$. The normal distribution is a **continuous** probability distribution, where continuous refers to the fact that the state space is *uncountable*. Contrary to countable sets, it is not possible to list each element of an uncountable set one at a time. An example of such a set is the set of real numbers.

If X is a random variable with a discrete or continuous distribution, we call X a discrete or continuous random variable, respectively. For a discrete random variable, we can write down the probability that it equals a specific value. For a continuous random variable, this is not possible, as there are an uncountable number of possible values. We can however specify the probability that a continuous random variable falls in a range of values by using the **density function**. The probability that a continuous random variable X assumes a value in $[a, b]$ is given by the integral of the density function over that interval. This corresponds to the area delimited by the graph of the density function, the x-axis and the vertical lines $y = a$ and $y = b$.

The density function of the normal distribution is shaped like a 'bell curve': symmetric around its mean and decreases on both sides of the curve. An example can be seen in Figure 1.3.8.

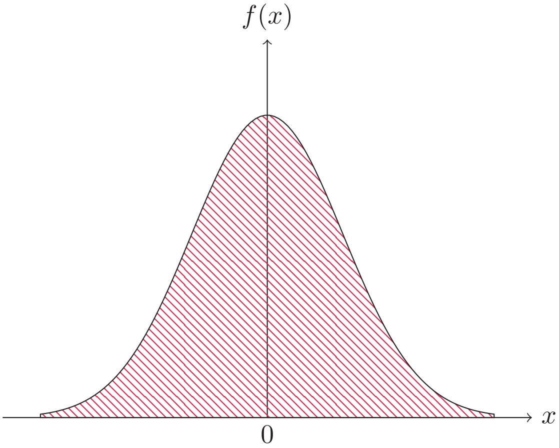


Figure 1.3.8. Example of the density of the normal distribution, for $\mu = 0$ and $\sigma = 1$. This is also called the standard normal distribution. The probability that X assumes a value in $[a, b]$ is equal to the area delimited by the density function, the x-axis and the vertical lines $y = a$ and $y = b$.

Intuitively, the normal distribution will occur when most observations are around the average, and there are few outliers. Because of this, we encounter many examples of the normal distribution in real life, for example:

- **heights of people**: most people are around average height, with few very tall or very short people;
- **size of products made by a machine**: if a product is made in a factory, the machine is set to produce the product to be a certain size. Though there may be some error, usually it is not far from the average;
- **errors in measurements**: think for example of measuring a length with a ruler, then usually your mistake would be small, not large.

1.3.7. Average value

DEFINITION 1.3.4. The arithmetic **mean** or **average value** of the numbers $a_1, \ldots, a_m$ is

$$A = \frac{1}{m} \sum_{i=1}^{m} a_i.$$

We average the total value over the total amount of numbers.

EXAMPLE 1.3.4. Consider the numbers $a_1 = 1$, $a_2 = 8$, $a_3 = -4$ and $a_4 = 5$. Then the average value is equal to

$$A = \frac{1}{4}(1 + 8 - 4 + 5) = \frac{5}{2}.$$

1.3.8. Expected value

Suppose that we have a random variable X, that can assume the values $x_1, x_2, \ldots$ with corresponding probabilities $p_1, p_2, \ldots$. If we draw X a couple of times, we obtain values x_i, over which we can compute the average value. But what if we would repeat drawing X for a long time, even indefinitely? Then as the number of experiments grows, the average value will grow closer and closer to the **expected value** of X, denoted by $\mathbb{E}[X]$.

DEFINITION 1.3.5. The **expected value** of a random variable X that can assume the values $x_1, x_2, \ldots$ with corresponding probabilities $p_1, p_2, \ldots$ is defined as

$$\mathbb{E}[X] = \sum_{i \geq 1} x_i p_i.$$

In other words, it is the probability weighted average of all possible values.

EXAMPLE 1.3.5. Let X be the outcome of a six-sided die, with equal probabilities for each outcome. Then X can assume the values

$$x_1 = 1, x_2 = 2, x_3 = 3, x_4 = 4, x_5 = 5, x_6 = 6$$

with probability $p_i = \frac{1}{6}$ each. Then the expected value is equal to

$$\mathbb{E}[X] = \sum_{i=1}^{6} x_i p_i = \frac{1}{6} \sum_{i=1}^{6} x_i = \frac{7}{2}.$$

So if we would roll a die a large amount of times, the average would be close to $\frac{7}{2}$. (You can try for yourself if it works, by throwing a die several times and calculating the average.)

EXAMPLE 1.3.6. Let X have a binomial distribution with parameters m and p. Then the expected value of X is

$$\mathbb{E}[X] = mp.$$

1.3.9. Matrices

DEFINITION 1.3.6. A **matrix** is a table of numbers, symbols or expressions, arranged in rows and columns. If the matrix M has m rows and n columns, we say that M is a $m \times n$-matrix, pronounced as 'm by n'. We write

$$M = (m_{ij})_{1 \leq i \leq m, 1 \leq j \leq n},$$

where m_{ij} is the element found on the intersection of row i and column j.
If either m or n is equal to 1, we may also refer to M as a **vector**.

A matrix is thus entirely defined by its dimensions m and n, and its elements m_{ij}. We usually write the matrix with brackets surrounding it.

EXAMPLE 1.3.7. Consider

$$A = \begin{bmatrix} 1 & 0 & 8 \\ 16 & -3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 10 \\ -2 \\ 7 \\ 50 \end{bmatrix}.$$

Then A is a 2×3-matrix and B is a 4×1-matrix, or a vector of length four.

If the elements of two matrices are numbers, we can define arithmetic operations on them. The easiest operations are addition, subtraction, scalar multiplication and transposition, since these can be done per element.

DEFINITION 1.3.7. Let A and B $m \times n$-matrices. Let $c \in \mathbb{R}$ be a constant. Then

- the **sum** of A and B is a $m \times n$-matrix with elements

$$(A+B)_{ij} = A_{ij} + B_{ij}, \quad 1 \leq i \leq m, 1 \leq j \leq n;$$

- the **scalar product** cA of c and A is a $m \times n$-matrix with elements

$$(cA)_{ij} = cA_{ij}, \quad 1 \leq i \leq m, 1 \leq j \leq n;$$

- the **transpose** $A^\top$ is a $n \times m$-matrix with elements

$$(A^\top)_{ji} = A_{ij}, \quad 1 \leq i \leq m, 1 \leq j \leq n.$$

EXAMPLE 1.3.8. Consider

$$A = \begin{bmatrix} 1 & 0 & 8 \\ 16 & -3 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -6 & 7 \end{bmatrix}.$$

Then

$$A+B = \begin{bmatrix} 2 & 2 & 11 \\ 16 & -9 & 11 \end{bmatrix}, \quad 2A = \begin{bmatrix} 2 & 0 & 16 \\ 32 & -6 & 8 \end{bmatrix}, \quad B^\top = \begin{bmatrix} 1 & 0 \\ 2 & -6 \\ 3 & 7 \end{bmatrix}.$$

EXERCISE 3. Let A and B be $m \times n$-matrices and $c \in \mathbb{R}$ be a constant. Show that the following properties are true.

(a) $(cA)^\top = cA^\top$;
(b) $(A+B)^\top = A^\top + B^\top$;
(c) $(A^\top)^\top = A$.

It becomes a bit more difficult when we consider multiplication of two matrices. Two matrices can only be multiplied when the number of columns of the first matrix equals the number of rows of the second matrix. We could also say that their *inner dimensions* should be the same.

DEFINITION 1.3.8. Let A be a $m \times n$-matrix and B a $n \times p$-matrix. Then the **product** AB of A and B is a $m \times p$-matrix with elements

$$(AB)_{ij} = \sum_{r=1}^{n} A_{ir} B_{rj}, \quad 1 \leq i \leq m, 1 \leq j \leq p.$$

In the summation, we multiply elements from A and B, where in A we travel along a row and in B we travel along a column.

EXAMPLE 1.3.9. Consider

$$A = \begin{bmatrix} 2 & 3 & 4 \\ 1 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 2000 \\ 1 & 100 \\ 0 & 10 \end{bmatrix}.$$

Then the product of A and B is

$$AB = \begin{bmatrix} 2\cdot 0 + 3 \cdot 1 + 4 \cdot 0 & 2 \cdot 2000 + 3 \cdot 100 + 4 \cdot 10 \\ 1 \cdot 0 + 0 \cdot 1 + 0 \cdot 0 & 1 \cdot 2000 + 0 \cdot 100 + 0 \cdot 10 \end{bmatrix} = \begin{bmatrix} 3 & 4340 \\ 0 & 2000 \end{bmatrix}.$$

We encounter applications of matrices in many scientific fields, for example they can be used for:

- **linear transformation**: a linear transformation is a function $L : \mathbb{R}^m \to \mathbb{R}^m$. It can be defined by writing down an $m \times m$-matrix M. We represent the point (x, y) by a column vector

$$\begin{bmatrix} x \\ y \end{bmatrix}$$

For every $x \in \mathbb{R}^m$, we then have $L(x) = Mx$ so we only need to multiply the matrix and the vector to obtain the value of the function. An example of a linear transformation is rotation, in which we rotate a vector in $\mathbb{R}^m$ over an angle to obtain a new vector in $\mathbb{R}^m$.

EXAMPLE 1.3.10. Let $m = 2$ and consider the **rotation matrix**

$$M = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}.$$

This matrix rotates points in the two-dimensional plane counterclockwise through an angle θ about the origin. If we rotate the point (x, y) over an angle of θ, we obtain the point

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{bmatrix}.$$

In Figure 1.3.9, we see that if we rotate the point $(1, 0)$ over an angle of $\frac{\pi}{4}$, we obtain the point

$$\begin{bmatrix} \cos\frac{\pi}{4} & -\sin\frac{\pi}{4} \\ \sin\frac{\pi}{4} & \cos\frac{\pi}{4} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos\frac{\pi}{4} \\ \sin\frac{\pi}{4} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\sqrt{2} \\ \frac{1}{2}\sqrt{2} \end{bmatrix}.$$

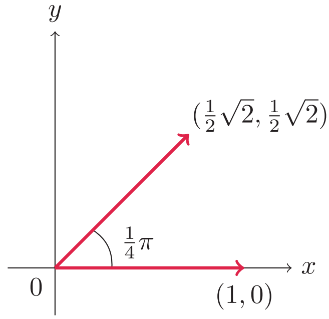


Figure 1.3.9. Rotation of $(1, 0)$ counterclockwise through an angle of $\frac{\pi}{4}$ about the origin.

- **stochastic matrix**: a stochastic matrix is a square matrix whose rows are probability vectors, i.e., all elements are non-negative and the rows sum to 1. Computers run Markov simulations, based on stochastic matrices, in order to model events ranging from weather forecasting to quantum mechanics. We will see the stochastic matrix, used in Markov chains, in Chapter 3, Section 3.2.

CHAPTER 2

First examples

In this chapter we present the following three research articles on networks:

1. The Marvel universe
2. A network of Thrones
3. The NS and the train schedules

Each section begins with a short abstract where the main ideas of the article are given, together with references for the mathematical notions needed.

2.1. The Marvel universe

The following article is based on the work of Alberich, Miro-Julia and Rosselló, see [1].
The authors construct a bipartite graph, with nodes corresponding to Marvel characters and comic books. The projection of the graph on its set of nodes-characters leads to a collaboration network: nodes represent characters, and edges link characters appearing in the same comic book. Furthermore, using the distributions of the edges in the original network, a random bipartite graph is also studied, in which edges are generated at random according to those distributions. The projection of this random bipartite graph on its set of nodes-characters produces a random collaboration network. The paper investigates the differences between the original and the random network by considering the average degree and clustering, as well as the small-world and scale-free phenomena.
Relevant concepts are degrees of nodes and bipartite graphs (Section 1.1), small-world and scale-free phenomena (Section 1.2), and average values (Section 1.3).

ABSTRACT. In the Marvel Universe collaboration network, two Marvel characters are considered linked if they jointly appear in the same Marvel comic book. We show that this network is not a random network, and that it has most, but not all, characteristics of "real-life" collaboration networks, such as movie actors or scientific collaboration networks.

2.1.1. Motivation

A **collaboration network** is a network in which nodes represent people and edges, that link pairs of nodes, denote the existence of some kind of collaboration between them.
A well-known collaboration network is the Movie Actors network, also known as the Hollywood network. In it, nodes represent actors and actresses, and a link is added between two nodes when they have jointly appeared in the same film. Scientific collaboration networks have also been thoroughly studied in the last years. In such a network, nodes represent scientists and links denote the co-authorship of a scientific piece of work contained in some database.
All collaboration networks present the same basic features:

(a) on average, every pair of nodes can be connected through a short path within the network;

(b) the probability that two nodes are linked is greater if they share a neighbor;
(c) the fraction of nodes with k neighbors decays roughly as a function of the form $k^{-\tau}$ for some positive exponent τ.

A network satisfying properties (a) and (b) is called a **small world**, and if it satisfies (c) then it is called **scale-free**. We analyze a new collaboration network, that is artificial, but mimics real-life networks: the **Marvel Universe collaboration network**. In it, the nodes correspond to Marvel Comics characters (Spider-Man, the Fantastic Four, the XMen, Captain America, etc.) and two nodes are linked when the corresponding characters have jointly appeared in the same Marvel comic book. The Marvel Universe network captures the social structure of the Marvel Universe, because most pairs of characters that have jointly appeared in the same comic book have fought shoulder to shoulder or against each other, or have had some other strong relationship, like family ties or kidnapping. On the other hand it is a purely artificial social network, whose nodes correspond to invented entities and whose links have been raised by a team of writers without any preconception.

Does the Marvel Universe network resemble real-life collaboration networks?

2.1.2. The Marvel Universe network

The Marvel Universe network consists of 6486 characters, appearing in 12942 comic books. From the data contained in the Marvel Chronology Project (MCP) database, we have built up a **bipartite graph**, with nodes corresponding to either Marvel characters or comic books, and edges from every character to all the books where it has appeared.

The table below summarizes the results obtained:

Number of characters:	6486
Number of books:	12942
Mean books per character:	14.9
Mean characters per book:	7.47
Distribution of characters per book:	$P_b(k) \sim k^{-3.12}$
Distribution of books per character:	$P_c(k) \sim k^{-0.66} 10^{-k/1895}$

From the above table it turns out that a Marvel character appears typically in about 14.9 comic books (the number of appearances spans from 1 to[1] 1625). The average number of characters per comic book is 7.47, with a range spanning from 1 to[2] 111.

Moreover, $P_b(k)$ is the probability that a comic book has k characters appearing in it, and this distribution is very similar to the one that can be found in real-life networks. On the other hand, the probability that a character appears in k comic books, given by $P_c(k)$, is different from what is normally found in bipartite graphs associated to collaboration networks.

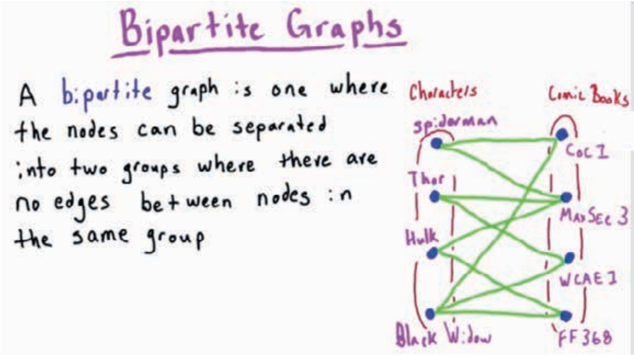
[1] This greatest value corresponds to Spider-Man.

[2] This last value is achieved by Issue 1 of Contest of Champions, where the Grandmaster and the Unknown took every superhero in the planet and selected two teams to battle it out.

Indeed here the exponent of k is only 0.66, which is much smaller than other values published for similar networks, that usually ranges from 2 to 3.

2.1.2.1. The null random model

We compare the results to a null random model. A reasonable random model would seem to be one with its same set of nodes and whose links have been generated by simply tossing a coin, so each link exists, independently of the other ones, with a fixed probability. This is a **random network**. We start from a random bipartite graph, called a MU-BR graph, with 6486 nodes-characters and 12942 nodes-books, and whose edges have been randomly created following exactly the same distributions $P_c(k)$ and $P_b(k)$ of outgoing and ingoing edges obtained in the previous section. Then, a **MU-R graph** is the projection of this random bipartite graph on its set of nodes-characters: i.e., its nodes correspond to characters and its links represent to be connected to the same book in a MU-BR graph. Our Marvel Universe (MU) graph has $N_c = 6486$ nodes-characters and $L_c = 168267$ links, that is pairs of characters that have collaborated in some comic book. It is important to note that the actual number of collaborations is 569770, but this value counts *all* collaborations, repetitions included.



Bipartite Graphs

A bipartite graph is one where the nodes can be separated into two groups where there are no edges between nodes in the same group

The **degree** of a node is the number of edges incident to the node, with loops counted twice. In this context, the number of characters that have jointly appeared with a given character in some comic book corresponds to the degree of this character. An application of the degree sum formula,

$$\text{deg}(\text{nodes}_1) + \text{deg}(\text{nodes}_2) + ... + \text{deg}(\text{nodes}_n) = 2 \times \text{number of total links},$$

implies that the average value for this degree in the MU collaboration network is

$$\frac{2L_c}{N_c} = 51.88,$$

i.e., a Marvel character has collaborated, on average, with 52 other characters. The range of this number of collaborators runs from 0 to 1933; the number of partners of Captain Amer-

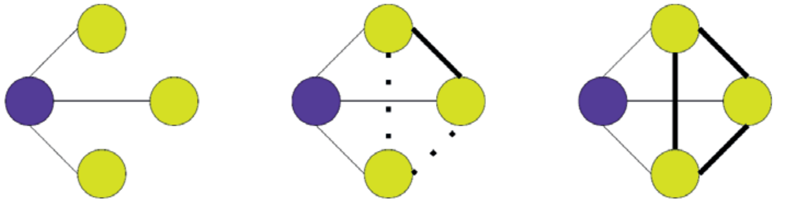
ica. Here we find a big difference between the values obtained in the MU network and in its null random model MU-R. In the MU-R graph we would expect all 569770 collaborations to form different links, which is about 3.4 times the actual number of links in the MU network. As a consequence, the average degree in MU-R is the average degree in MU multiplied by this same factor, and would therefore become 175.69. If the MU collaboration network had been created in a purely random way, a Marvel character would have collaborated on average with more than 175 other characters!
It is shown that in the Hollywood graph and in several scientific collaboration networks the actual average degree is consistently smaller than the theoretical average degree of the corresponding random model, but not by such a large factor as the one found here. This indicates that Marvel characters are made to collaborate repeatedly with the same characters, which reduces their total number of co-partners well below the expected number in the random model, and that they collaborate quite more often with the same people than real movie actors or scientists do.

2.1.2.2. Clustering

In most social networks, two nodes that are linked to a third one have a higher probability to be linked between them: two acquaintances of a given person probably know each other. This effect is measured using the **clustering coefficient**. Given a node v in a network, let k_v be its degree, i.e., the number of neighbors of v, and let N_v be the number of links between these k_v neighbors of v. If all these nodes were linked to each other, then N_v would be equal to the number of unordered pairs of nodes belonging to this set of k_v neighbors, which is $k_v(k_v - 1)/2$. The clustering coefficient C_v of node v rates the difference between the actual value N_v and this greatest value by taking their quotient:

$$C_v = \frac{N_v}{k_v(k_v - 1)/2}.$$

Thus, this coefficient C_v measures the fraction of neighbors of node v that are linked. Notice that $0 \leq C_v \leq 1$. The clustering coefficient C of a network is then defined as the mean value of the clustering coefficients of all its nodes. It represents the probability that two neighbors of an arbitrary node are linked.



(a) No pairs formed among neighbors: ***C* = *0***

(b) One pair formed among neighbors: ***C* = *1 / 3***

(c) Three pairs formed among neighbors: ***C* = *3 / 3***

In a uniform random network with n nodes and m links, it can be proved that the expected value of the clustering coefficient is nothing but the probability that two randomly selected nodes are connected; in other words

$$C_{\text{random}} = \frac{2m}{n(n-1)}.$$

Against what happens with real-life social networks, it turns out that the clustering coefficient of MU is small. Its value is

$$C_{\text{Marvel}} = 0.012,$$

while the clustering coefficient of a random network with 6486 nodes and 168267 links is $C_{\text{random}} = 0.008$. Thus, roughly

$$C_{\text{Marvel}} \sim 1.5 \times C_{\text{random}},$$

and not several orders of magnitude larger.
This result separates the Marvel Universe from all other, real-life, collaboration networks. But if we use the MU-R as a null random network to compare the clustering coefficient of the MU network the analysis changes quite drastically. The expected value of the clustering coefficient of the null random model MU-R is

$$C_{\text{MU-R}} = 0.0066.$$

Thus, the measured clustering coefficient is about double the one predicted by MU-R

$$C_{\text{Marvel}} \sim 2 \times C_{MU-R},$$

and this agrees with what is observed in real-life networks.
Our analysis shows that the Marvel Universe behaves “realistically” when compared to MU-R, but not when compared to a uniform random network. Real-life collaboration networks have as clustering coefficient roughly twice the one of their null random model, and the latter turns out to be highly clustered. The clustering coefficient of the MU network is also roughly twice the one of its null random model, but this null random model is not highly clustered, having a clustering coefficient only three times that of a random network with the same number of nodes and links. We believe that, as we already argued in connection with the average degree, this is a hint of the artificiality of the bipartite graph which projects into MU. It seems that Marvel writers have not assigned characters to books in the same way as natural interactions would have done it, with the global effect that the combination of the distributions $P_c(k)$ and $P_b(k)$ is very different from what would be found in real-life networks, yielding non-clustered graphs. But, once we have these distributions, the Marvel Universe behaves realistically and is significantly different from a random network.

2.1.3. Conclusion

Real-life collaboration networks of very different origins, sizes and styles present common basic features: they are scale-free and they exhibit the small-world property. We have studied the Marvel Universe, which is a collaboration network that is artificial and has been created with no special intention during the past 40 years by a team of comic book writers. Although to some extent the Marvel Universe tries to mimic human relations, it is completely different from a random network, and we have shown that it cannot completely hide its artificial origins. As in real-life collaboration and, in general, social networks, its nodes are on average at a short distance of each other, but its clustering coefficient is quite smaller than what's usual in real-life collaboration networks.

From this comparison we deduce that the artificiality of the Marvel Universe network lies mainly on the distributions of edges in the bipartite graph which yields it, because the relationship between the Marvel Universe network's data and those of its null random model is similar to that of real-life collaboration networks' data and their corresponding null random models.

2.2. A network of Thrones

The following is an article by Beveridge and Shan, see [3].

The third book of the Game of Thrones series, namely A Storm of Swords, is used to construct a social network. Characters of the book are represented by nodes, connected by integer-weighted edges: higher weights correspond to stronger relationships between those characters. With its 107 nodes and 353 edges, the network looks very complex, underlining the interweaving of the story. The authors use modularity to detect seven communities inside the network, and six different centrality measures to distinguish a few of the most influential people amongst all the characters.

Relevant concepts are modularity (Section 1.2) and degrees of nodes (Section 1.1).

The internation hit HBO series *Game of Thrones*, adapted from George R. R. Martin's epic fantasy novel series *A Song of Ice and Fire*, features interweaving plot lines and scores of characters. With so many people to keep track of in this sprawling saga, it can be a challenge to fully understand the dynamics between them.

To demystify this sage, we turn to *network science*, a new and evolving branch of applied graph theory that brings together traditions from many disciplines, including sociology, economics, physics, computer science, and mathematics. It has been applied broadly across the sciences, the social sciences, the humanities and in industrial settings.

In this article we perform a network analysis of Game of Thrones to make sense of the intricate character relationships and their bearing on the future plot (but we promise: no spoilers!).

First, a quick introduction to *Game of Thrones:* Westeros and Essos, separated by the Narrow Sea, are homes of several noble houses (see Figure 2.2.1). The narrative starts at a time of peace, with all the houses unified under the rule of King Robert Baratheon, who holds the Iron Throne. Early on, King Robert dies in a hunting accident, and the young, cruel Prince Joffrey ascends the throne, backed by his mother's house, Lannister. However, the prince's legitimacy, and even his identity, are seriously questioned across the kingdom. As a result, war breaks out, with multiple claimants to the Iron Throne.

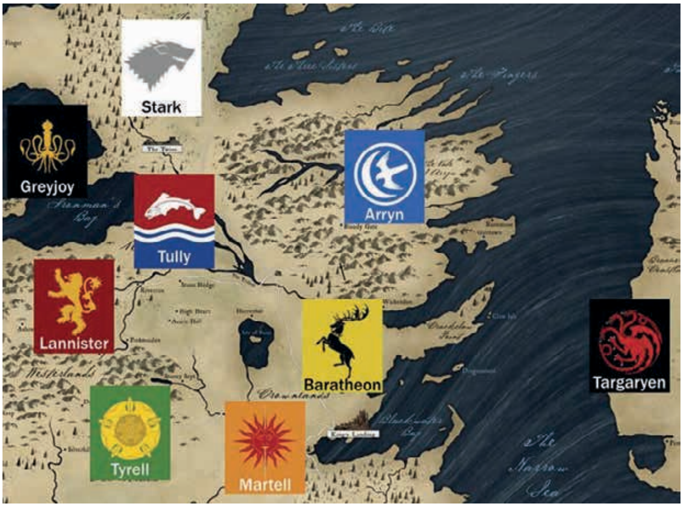


Figure 2.2.1. The Game of Thrones world: Westeros, the Narrow Sea, and Essos (from left to right). Sigils represent the locations of the noble houses at the beginning of the saga.

Driven by cause or circumstance, characters from the many noble families launch into arduous and intertwined journeys. Among these houses are the honorable Stark family (Eddard, Catelyn, Robb, Sansa, Arya, Bran, and Jon Snow), the pompous Lannisters (Tywin, Jaime, Cersei, Tyrion, and Joffrey), the slighted Baratheons (led by Robert's brother Stannis) and the exiled Daenerys, the last of the once-powerful House Targaryen.

2.2.1. The Social Network

Our first task is to turn the *Game of Thrones* world into a social network. Our network, shown in Figure 2.2.2, has sets of vertices V and edges E. The 107 vertices represent the characters, including ladies and lords, guards and mercenaries, councilmen and consorts, villagers and savages. The vertices are joined by 353 integer-weighted edges, in which higher weights correspond to stronger relationships between those characters.

We generated the edges using *A Storm of Swords*, the third book in the series. We opted for this volume because the main narrative has matured, with the characters scattered geographically and enmeshed in their own social circles. We parsed the ebook, incrementing the edge weight between two characters when their names (or nicknames) appeared within 15 words of one another. Afterward, we performed some manual validation and cleaning. Note that an edge between two characters doesn't necessarily mean that they are friends—it simply means that they interact, speak of one another, or are mentioned together.

The complex structure of our network reflects the interweaving plotlines of the story. Notably, we observe two characteristics found in many real-world networks. First, the network contains multiple denser subnetworks, held together by a sparser global web of edges. Second, it is organized around a subset of highly influential people, both locally and globally. We now describe how to quantify these observations using the analytical tools of network science.

2.2.2. Community Detection

The network layout and colors in Figure 2.2.2 clearly identify seven communities: the Lannisters and King's Landing, Robb's army, Bran and friends, Arya and companions, Jon Snow and the far North, Stannis's forces, and Daerenys and the exotic people of Essos. Remarkably, these communities were identified from only the network structure, as we explain below.

We want to divide the network into coherent communities, meaning that there are many edges within communities and few edges between communities. We detect our network communities by using a global metric called *modularity*. Let $w_{ij} \geq 0$ denote the weight of the edge between vertices i and j, where $w_{ij} = 0$ when there is no edge. Let $k_i = \sum_{j \in V} w_{ij}$ denote the weighted degree of vertex i. Intuitively, the modularity Q compares our given network to a network with the same weighted degrees, but in which all edges are rewired at random. This random network should be community-free, so it makes a good baseline for comparison.

Suppose that vertices i and j belong to the same community C. We would expect that w_{ij} is at least as large as the number of edges between them in our randomly rewired network. A touch of combinatorial probability shows that the expected number of such random edges is $k_i k_j / 2m$, where m is the total number of edges in the network. Summing over all vertices in a community C, we have

$$\sum_{i,j \in C} \left(w_{ij} - \frac{k_i k_j}{2m} \right) \geq 0.$$

Meanwhile, if C is not actually a community, then this quantity may be negative. The modularity Q of a vertex partition $C_1, \ldots, C_l$ of the network is

$$Q = \frac{1}{2m} \sum_{k=1}^{l} \sum_{i,j \in C_k} \left(w_{ij} - \frac{k_i k_i}{2m} \right),$$

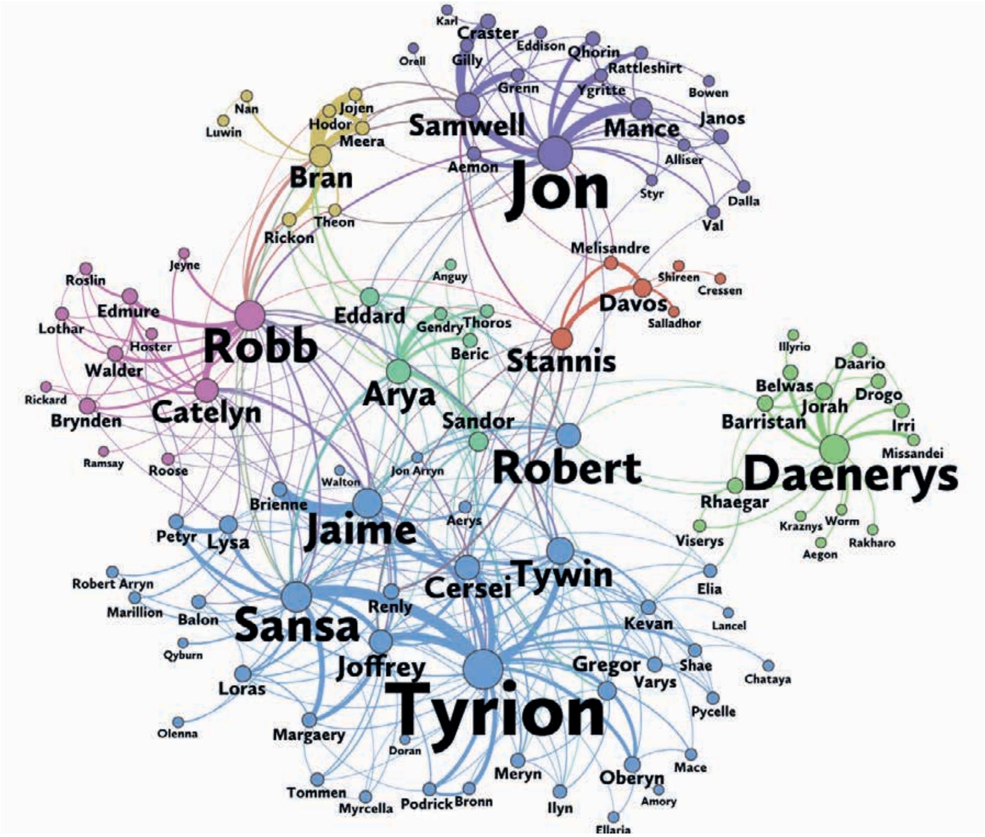


Figure 2.2.2. The social network generated from A Storm of Swords. The color of a vertex indicates its community. The size of a vertex corresponds to its PageRank value, and the size of its label corresponds to its betweenness centrality. An edge's thickness represents its weight.

where we have normalized this quantity so that $-1 \leq Q \leq 1$.

Our goal is to partition the vertices into communities so as to maximize Q. Finding this partition is computationally difficult, so we use a fast approximation algorithm called the Louvain method.

Crucially, the algorithm determines the number of communities; it is not an input. In our case, we discover the seven communities in Figure 2.2.2. The King's Landing community accounts for 37 percent of the network. When we perform community detection on this major subnetwork, we obtain four communities. A high resolution version of Figure 2.2.2 and the network of subcommunities of King's Landing can be found at *maa.org/math-horizons-supplements*.

2.2.3. Centrality Measures

Network science can also identify important vertices. A person can play a central role in multiple ways. For example, she could be well connected, be centrally located, or be uniquely positioned to help disperse information or influence others. Figure 2.2.3 displays the importance of 14 prominent characters, according to six centrality measures, which we explain below.

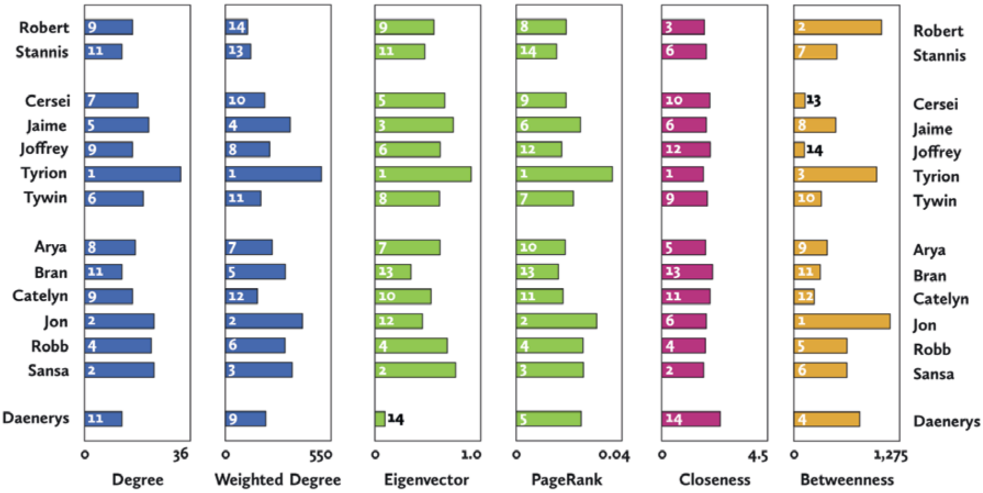


Figure 2.2.3. Centrality measures for the network. Larger values correspond to greater importance, except for closeness centrality, where smaller values are better. Numbers in the bars give the rankings of these characters.

Degree centrality is the number of edges incident with the given vertex. *Weighted degree centrality* is defined similarly by summing the weights of the incident edges. In our network, degree centrality measures the number of connections to other characters, while weighted degree centrality measures the number of interactions.

Eigenvector centrality is weighted degree centrality with a feedback loop: A vertex gets a boost for being connected to important vertices. The importance x_i of vertex i is the weighted sum of the importance of its neighboring vertices: $x_i = \sum_{j \in V} w_{ji} x_j$ for each $i \in V$. Solving the resulting linear system gives the eigenvector centrality. (This name comes from linear algebra: We actually find an eigenvector for eigenvalue $\lambda = 1$ of the matrix W with entries w_{ij} .)

Let's compare the weighted degree and eigenvector centralities for our network. The late King Robert receives a huge boost: He has only 18 connections, but half of them are to other prominent players! Most leading characters also benefit from the feedback loop, being directly involved in the political intrigue and sweeping military turmoil that grips the realm. The exceptions are isolated from the main action: Bran (presumed dead and on the run), Jon Snow (marginalized in the far North), and Daenerys (exiled across the Narrow Sea).

PageRank is another variation on this theme. This measure was the founding idea behind Brin and Page's Google search engine. Each vertex has an inherent importance $\beta \geq 0$, along with an importance acquired from its neighbors. Unlike eigenvector centrality, a vertex does not get full credit for the total importance of its neighbors. Instead, that neighbor's importance is divided equally among its direct connections. In other words, a vertex of very high degree passes along only a small fraction of its importance to each neighbor.

The PageRank y_i of vertex i is given by

$$y_i = \alpha \sum_{j \in V} \frac{w_{ji}}{k_j} y_j + \beta,$$

where $\alpha + \beta = 1$ and $\alpha, \beta \geq 0$. Researchers typically use $\beta = 0.15$ to find an effective balance between inherent importance and the neighborhood boost.

PageRank does not penalize our three far-flung characters and actually has the opposite effect on Daenerys. In fact, the PageRank ordering is nearly identical to the degree centrality ordering, except Daenerys jumps from 12th place to fifth place. So PageRank correctly identifies the charismatic Daenerys as one of the most important players, even though she has relatively few connections.

This brings us to two centrality measures whose definitions take a more global view of the network. The *closeness centrality* of a vertex is the average distance from the vertex to all other vertices. (Unlike the other centrality measures, lower values correspond to greater importance.) The closeness values for our list of main characters is quite compressed, except for the faraway Daenarys. However, Tyrion and Sansa have a slight edge over everyone else.

The final centrality measure is the most subtle. The *betweenness centrality* of a vertex measures how frequently that vertex lies on short paths between other pairs of vertices. Mathematically, the betweenness z_i of vertex i is

$$z_i = \sum_{j,k \in V} \frac{\sigma_{jk}(i)}{\sigma_{jk}},$$

where σ_{jk} is the number of (j,k)-shortest paths and $\sigma_{jk}(i)$ is the number of these (j,k)-shortest paths that go through vertex i. A vertex that appears on many short paths is a broker of information in the network: Efficient communication between different parts of the network will frequently pass through such a vertex. Such connectors have the potential to be highly influential by inserting themselves into the dealings of other parties.

Betweenness centrality gives a distinctive ranking of the characters. This is the only measure in which Tyrion does not come out on top: He places third, behind Jon Snow (thanks to his ties to both House Stark and the remote denizens of the North) and Robert Baratheon (the only person directly connected to all four noble houses of the leading characters). Meanwhile, Daenarys rises to fourth place (her best showing) because of the hub-and-spoke nature of the eclectic Essos community.

There is no single "right" centrality measure for a network. Each measure gives complementary information, and taking them in concert can be quite revealing. In our network, three characters stand out consistently: Tyrion, Jon, and Sansa. Acting as the Hand of the King, Tyrion is thrust into the center of the political machinations of the capitol city. Our analysis suggests that he is the true protagonist of the book.

Meanwhile, Jon Snow is uniquely positioned in the network, with connections to highborn lords, the Night's Watch militia, and the savage wildlings beyond the Wall. The real surprise may be the prominence of Sansa Stark, a de facto captive in King's Landing. However, other players are aware of her value as a Stark heir and they repeatedly use her as a pawn in their plays for power. If she can develop her cunning, then she can capitalize on her network importance to dramatic effect.

Meanwhile, Robert and Daenarys stand out by overperforming in certain centrality measures. They provide a clear counterpoint to one another and return our attention to the Iron Throne itself. Robert's memory unifies the crumbling network of the recent past, while Daenarys will surely upend the current network when she returns to Westeros in pursuit of the throne.

2.2.4. A Networked Life

We have visited the realms of Westeros and Essos to tour the basic tools of network science. We performed an empirical analysis of our network, finding communities and identifying influential people. Our network analysis confirmed some expectations and provided new insights into this richly imagined saga. We have considered a fanciful application of network science to give an enticing taste of its capabilities. More serious applications abound, and network science promises to be invaluable in understanding our modern networked life.

2.2.5. Further Reading

(1) A.-L. Barabási, *Network Science.* barabasilab.neu.edu/networksciencebook/downlPDF.html.

(2) D. Easley, J. Kleinberg. *Networks, Crowds and Markets.* Cambridge University Press, Cambridge, 2010.

(3) S. Fortunato. *Community detection in graphs.* Physics Reports 486 nos. 3–5 (2010) 75–174.

(4) M. Newman, *Networks: An Introduction.* Oxford University Press, Oxford, 2010.

2.3. The NS and the train schedules

The following article is based on the research work done by Kroon and Schrijver for the NS, see [6].
The authors develop the algorithms used by the NS to determine the train schedules. They construct a graph in which nodes represent the arrival and departure times of a train on a station, and edges indicate stages of the journey or waiting times at stations. The edges are weighted with time intervals, with the property that the next arrival or departure time should lie in that interval. A time schedule is then obtained by determining a function (potential) on the graph that satisfies the time restrictions imposed.
Relevant concepts are directed graphs and in/out degrees of nodes (Section 1.1).

2.3.1. Preliminaries

DEFINITION 2.3.1. We define a **network** to be a directed graph: a collection of points (vertices), denoted by V and a collection of directed edges (arrows), denoted by E. The network will be denoted by (V, E). For two vertices u, v, we denote by (u, v) an arrow starting from u and going to v. For a vertex u, we define its **indegree**, denoted by δ_{in} as the number of arrow starting from some vertex w and going to vertex u and its **outdegree**, denoted by δ_{out}, as the number of arrows leaving from vertex u.

DEFINITION 2.3.2. A function $g : E \to \mathbb{R}$ will be called a **voltage** if there exists a function $p : V \to \mathbb{R}$ (called the **potential**) such that

$$g(u, v) = p(v) - p(u)$$

for all edges $(u, v) \in E$.

2.3.2. The railway infrastructure network

When one thinks of the railway the infrastructure network comes immediately in mind, see Figure 2.3.1 (left): the network of stations which are connected through train tracks. A more detailed network which lies on top of the infrastructure network, is the network of train lines, see Figure 1 (right): the direct train connections, each one with an origin, a destination, a frequency and a list of stations where the train has to stop.
If you zoom in further, for example in the network of platforms and tracks in the train station in Gouda, then you see a much more detailed network, see Figure 2.3.2.

2.3.3. From infrastructure to scheduling and mathematics

The concepts of potential, voltage and current, crucial in the theory of electromagnetism, can be used in the logistics of the railways as well. In this article we will focus on determining the train time schedule. We will first state the requirements the time schedule needs

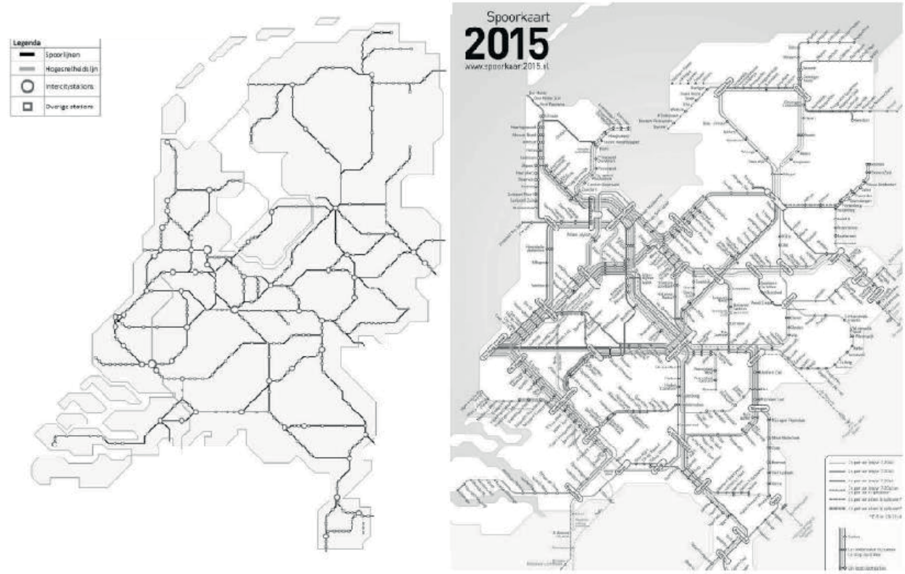


Figure 2.3.1. Left: the network of stations and tracks, right: the network of train lines.

to satisfy. These requirements will be described using a network. Afterwards the train time schedule will be described as a potential on this network. The vertices of this network will represent the departure and arrival times.

2.3.4. The time schedule and the network of requirements

The basis of the Dutch train time schedule is the Hourly Scheduling Scheme (Basis Uur Patroon BUP): the time schedule is periodic and repeats itself every 60 minutes. Thus, if at time instant t a train departs from station S, then trains from the same station depart at time instances

$$\ldots, t-120, t-60, t+60, t+120, \ldots$$

as well, all of them with the same destination. Additional special trains may be added, during rush hours or night trains for example, but the underlying basis is the BUP. An illustration of this scheme is as follows: at every hour h, at h:26 (10:26, 11:26, . . ., 22:26), a (Sprinter) train departs from Zwolle station with destination Emmen, from early in the morning until late in the evening. During a period of one hour every departure and arrival can be given by indicating the minutes of the hour, thus as a number modulo 60 ($x \mod 60$).

DEFINITION 2.3.3 (Modulo Arithmetic). For some given number x, we say that two numbers a and b are equivalent **modulo** x when they both give the same residual when divided

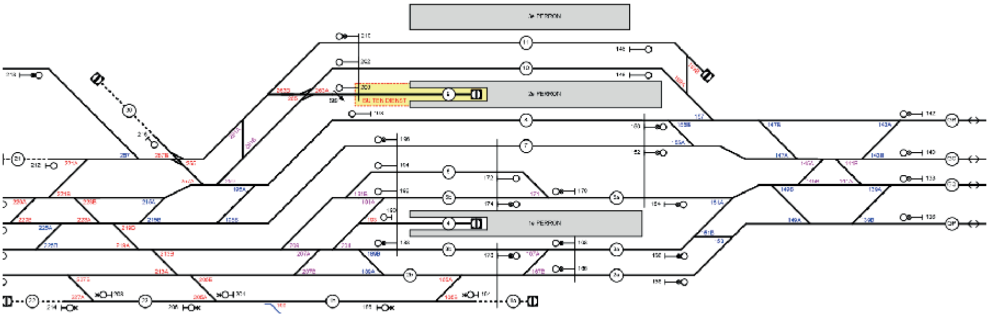


Figure 2.3.2. A detailed network of platforms and tracks in the train station in Gouda.

by x. For example, the numbers 3 and 1 are equivalent modulo 2 and the numbers 2 and 62 are equivalent modulo 60. We write $1 = 3 \mod 2$ and $2 = 62 \mod 60$.

For every hourly train connection we would like to schedule we define two points, one point representing the departure time of that train and one point for its arrival time. Let's see an example, look at train line 15, which is an hourly train from Amsterdam to Amersfoort. Its program is given in Figure 2.3.3. The index 15 corresponds to the specific train line. This train stops in Hilversum and waits there for one minute. Hence we represent this ride by adding four points in our network, those are

$$v_{15,Amd,D}, \quad v_{15,Hvs,A}, \quad v_{15,Hvs,D}, \quad v_{15,Amf,A}.$$

Amd, Hvs and Amf stand for Amsterdam, Hilversum and Amersfoort and A,D stand for Arrival and Departure time. An example: $v_{15,Amd,D}$ corresponds to the time the train of line 15 should depart from Amsterdam Central Station.

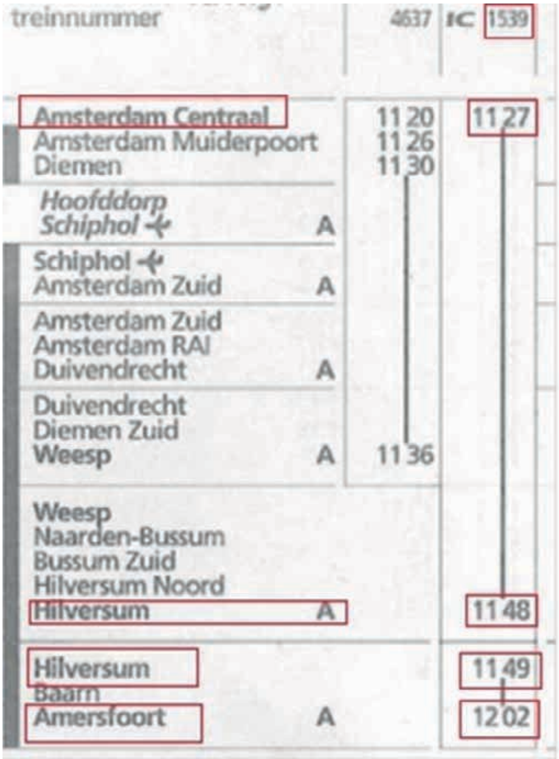
Afterwards, we insert brackets to distinguish the different stages and stations where the train from line 15 has to stop, this is

$$(v_{15,Amd,D}, v_{15,Hvs,A}), (v_{15,Hvs,A}, v_{15,Hvs,D}), (v_{15,Hvs,D}, v_{15,Amf,A}).$$

Hence this expression represents a route of a train. Can you explain what this expression indicates?

The next step is to assign the travel times and the waiting times at the stations where a stop is scheduled. For every bracket (u, v), which now represents a stage of the route (travel between two stations) or a stop at some station, we assign a time interval $I(u, v)$ with the property that the next arrival or departure time should lie in this interval. In the Amsterdam-Amersfoort example we have

$$\begin{aligned} I(v_{15,Amd,D}, v_{15,Hvs,A}) &= [20, 22] \\ I(v_{15,Hvs,A}, v_{15,Hvs,D}) &= [1, 2] \\ I(v_{15,Hvs,D}, v_{15,Amf,A}) &= [12, 13]. \end{aligned}$$



treinnummer		4637	IC 1539
Amsterdam Centraal		11 20	11 27
Amsterdam Muiderpoort		11 26	
Diemen		11 30	
Hoofddorp			
Schiphol	A		
Schiphol			
Amsterdam Zuid	A		
Amsterdam Zuid			
Amsterdam RAI			
Duivendrecht	A		
Duivendrecht			
Diemen Zuid			
Weesp	A	11 36	
Weesp			
Naarden-Bussum			
Bussum Zuid			
Hilversum Noord			
Hilversum	A		11 48
Hilversum			11 49
Baarn			
Amersfoort	A		12 02

Figure 2.3.3. The train programs from lines 46 and 15.

The formulas above say that the travel time from Amsterdam to Hilversum should be between 20 minutes and 22 minutes, the waiting time at Hilversum station should be between 1 minute and 2 minutes and the travel time from Hilversum to Amersfoort should be between 12 minutes and 13 minutes. We can represent this as a directed graph, see Figure 2.3.4.

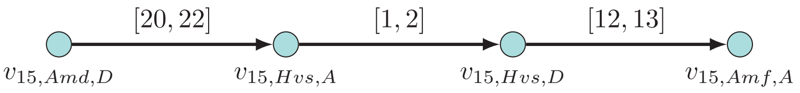


Figure 2.3.4. Graphical representation of the train connection Amsterdam-Amersfoort.

We find a time schedule for this specific train line, given all these constraints, by determining a potential p, such that for every bracket (u, v) we have $p(v) - p(u) \in I(u, v)$ or equivalently by determining a voltage g so that for every bracket (u, v) we have $g(u, v) \in I(u, v)$.

Applying this to the train line 15 described, we find

$$\begin{aligned} p(v_{15,Amd,D}) &= 27 \\ p(v_{15,Hvs,A}) &= 48 \\ p(v_{15,Hvs,D}) &= 49 \\ p(v_{15,Amf,A}) &= 2. \end{aligned}$$

EXERCISE 4. Given the above computation, what is the actual time schedule for line 15 between Amsterdam and Amersfoort? Show that the potential above defines a voltage g which also satisfies the desired requirements.

This is the procedure to schedule one specific train line. But in reality you have a whole network of train lines which might have stations in common. Hence you must avoid the scenario two trains meeting each other or having to use the same platform at some station at the same time. All these issues make the computations more complicated but the main idea remains the same. You make the network of departures and arrivals (as in Figure 2.3.4) with the minimum and maximum travel and waiting time and then you find a potential satisfying the requirements.

CHAPTER 3

Ongoing research

This chapter focuses on the research done within the NETWORKS programme. It contains four sections on different research topics:

1. Networks with communities
2. Markov chains and their applications
3. Networks in biology
4. The phenomenon of synchronisation

Each section begins with a short introduction and proceeds with a theoretical analysis. In each section a number of examples and exercises have been included.

3.1. Networks with communities

The following is a research work by Clara Stegehuis. The author gives few examples of networks that can be divided into communities: ideally, there should be relatively many connections between nodes in the same community, and relatively few connections between nodes in different ones. A way to discover communities is by maximising the modularity of the network. Some examples are shown by considering a telephone networks, LinkedIn and Facebook.
Relevant concepts are the degree of a node (Section 1.1) and modularity (Section 1.2).

In Figure 3.1.1 below the telephone network of Belgium is depicted. The points in this network correspond to households; there is a line between two households if there has ever been a phone call between them. It is evident that phone calls in Belgium occur mostly between two communities. One community of French speaking people, who frequently contact each other and a community of Flemish speaking people, who also frequently contact each other. Between the two communities there are but a few connections, which is not so weird because of the language barrier. But such a structure can also be observed in countries where there are no linguistic barriers. In the same figure, with the telephone network of Great Britain, we observe a similar community structure. People from the same province contact each other more often than people from different provinces. London connects all different communities in the phone line network with each other.

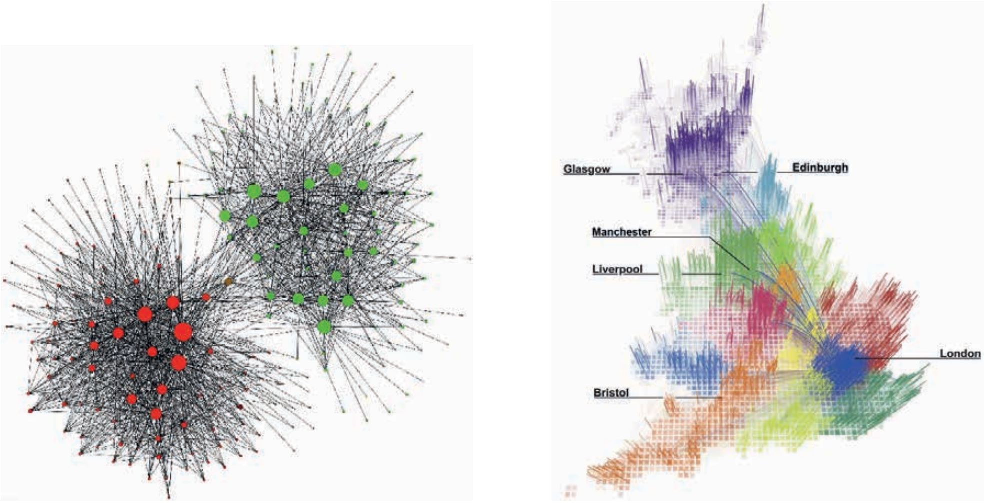


Figure 3.1.1. The telephone networks of Belgium (left) and Great Britain, ordered in communities.

Besides these specific examples of networks there exist also much more networks where we observe such a community structure. Think for example of networks in the human brain or social media. In Figure 3.1.2 you can see for example an image of Clara’s LinkedIn network. Points in this network depict her LinkedIn connections, two points being connected means that two of her friends are also friends with each other. At first instance this network looks like a complete chaos. But if we rearrange the points in an appropriate way, the community structure is clearer. A question that arises at this point is how someone could mathematically identify this community structure from the “chaos” of the first figure. This is a quite challenging question that has motivated substantial research.

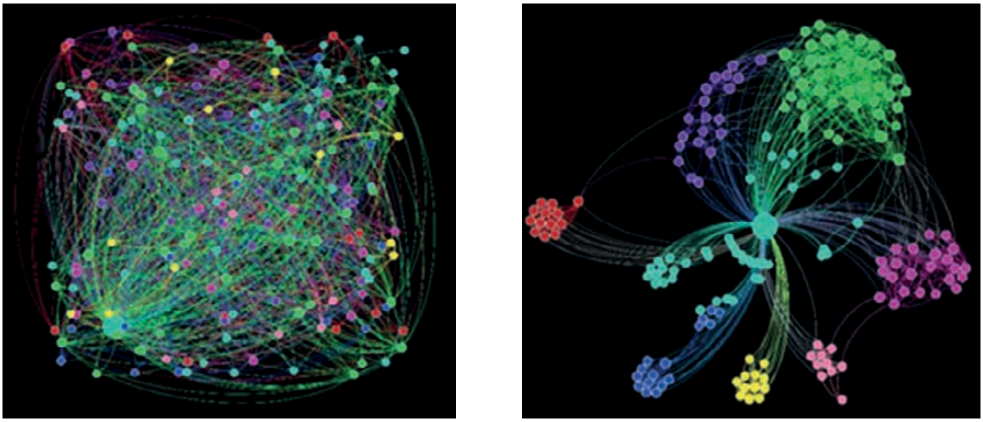


Figure 3.1.2. Clara's LinkedIn network, in random order (left) and ordered in communities.

3.1.1. What is a community?

In order to discover where the communities are located in a network, we must first strictly define what a community is. If we look at the previous figures, it is intuitively clear that a community must satisfy the following two properties:

- there are relatively many connections between points in the same community;
- there are relatively few connection between points in different communities.

One of the first methods in order to discover communities in a network uses these two properties. This method maximises the modularity. Suppose that we partition a network in some way into K communities. The modularity of this partition is defined as

$$M = \sum_{C=1}^{K} \left(\frac{L_C}{L} - \left(\frac{{k_C}^2}{2L} \right) \right) = \sum_{C=1}^{K} \frac{L_C}{L} - \sum_{C=1}^{K} \left(\frac{k_C}{2L} \right)^2 .$$

In the expression[3] above, L denotes the total number of connections in the network, L_C denotes the total number of connections in community C and k_C denotes the sum of the degrees (all connections leaving from some point in community C) of all points in community C. Intuitively modularity measures how much more connections there exist between the communities from what we would expect. This can be seen from the following argument. Suppose we want to maximise the modularity by maximising the first term in the formula for M. This would mean that we want as many connections as possible in each community. Then the best partition is given by considering the whole network as one community and connecting all points with each other. Then all connections in the network lie in one community, which is not what we observe in reality. That is why we subtract the second term. This second term describes the expected number of connections in community C if we would construct the whole network from scratch with the same number of total connections and the same degrees, but the connections placed completely at random.

EXERCISE 5. Compute the modularity from the formula above for the complete graph in Figure 3.1.3 and the graph with community structure in 3.1.4. Choose your own communities. What do you notice about the value of the modularity?

3.1.2. Overlapping communities

There are still many open problems that mathematicians try to solve. In social media for example most people belong to different communities: family, sport friends or colleagues for example. In such networks it is not possible to associate to each point a unique community because the communities often overlap. In such networks it is much more difficult to identify the various communities. An example of a Facebook network with overlapping communities is shown in Figure 3.1.5.

[3] We show that this definition of modularity is equivalent to Equation 1.2.1 in Section 1.2. Let w_{ij} be the number of connections between node i and j. Then

$$2L_C = \sum_{i,j \in C} w_{ij}$$

since we count each edge twice, once for i and once for j. Furthermore, let h_i be the degree of node i. As k_C is the sum of the degrees of all nodes in community C, we have

$$k_C = \sum_{i \in C} h_i .$$

It follows that

$$M = \sum_{C=1}^{K} \left(\frac{L_C}{L} - \left(\frac{k_C}{2L} \right)^2 \right) = \frac{1}{2L} \sum_{C=1}^{K} \sum_{i,j \in C} \left(w_{ij} - \frac{h_i h_j}{2L} \right) .$$

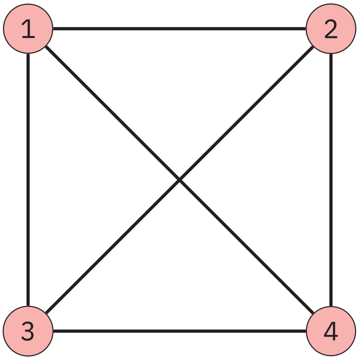


Figure 3.1.3. A complete graph, i.e., a graph in which each pair of nodes is connected.

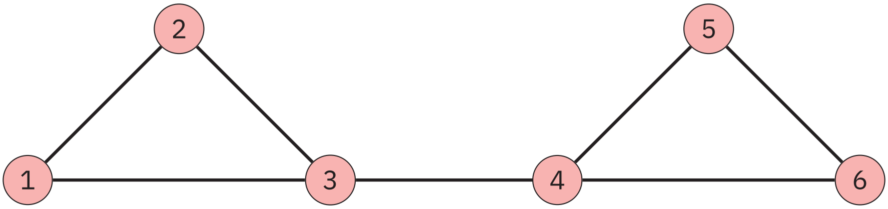


Figure 3.1.4. A graph with community structure.

The red circles denote friends who are all members of the same sport association. The purple triangles denote friends who also practice this sport, but are not members of the sport association. The green squares denote friends who are colleagues from work. There are also some friends who belong to neither of those categories. Our Facebook user is denoted by the blue node with number 125. After ordering the friends in four communities, members of the sport association, sport enthousiasts that are not members of the association, colleagues, and other friends, we get the picture in Figure 3.1.6. Can you say something about this network using modularity?

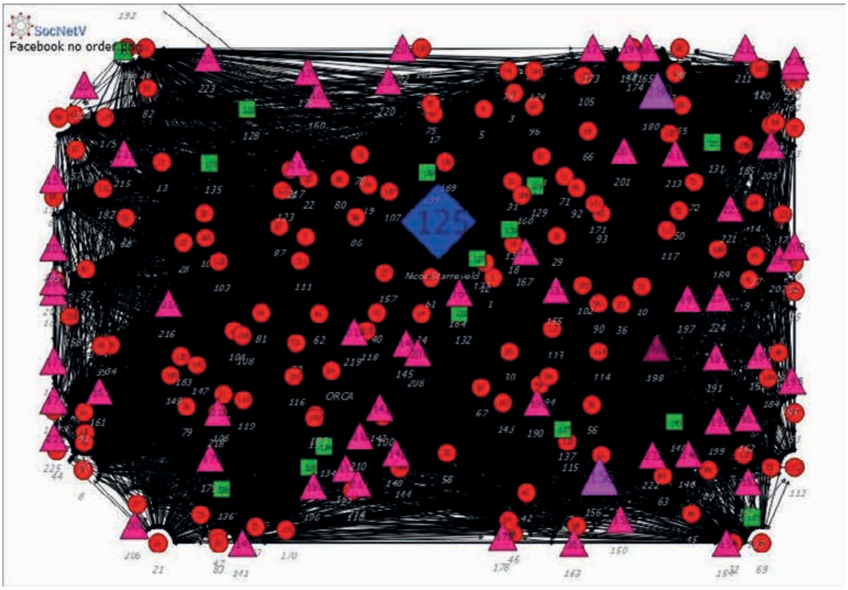


Figure 3.1.5. An unordered Facebook network, with some overlap in communities.

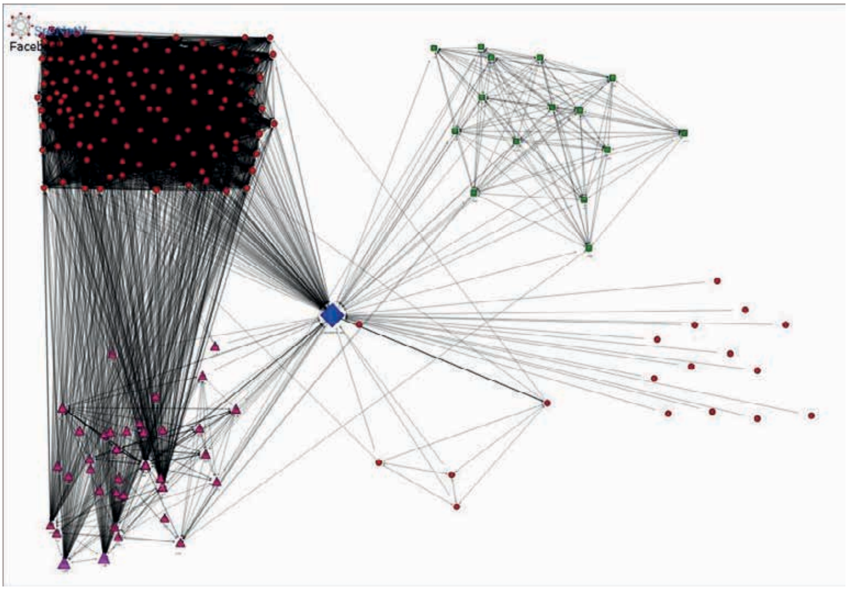


Figure 3.1.6. The Facebook network from Figure 3.1.5, ordered into four communities.

3.2. Markov chains and their applications

The following research work is presented by Jan-Pieter Dorsman. The weather and the Page Rank algorithm are considered as two instances of Markov chains. For the former, a one-step transition matrix is defined to predict with certain probability the weather in the coming future. For the latter, a simplified version of the algorithm is introduced, to show how Google's search engine instantly returns ordered pages from a collection of more than 25 billion documents, matching the search criteria.
Relevant concepts are directed graphs (Section 1.1) and matrix multiplication (Section 1.3).

3.2.1. The weather as Markov chain

A **Markov chain** is a special form of a stochastic process. It describes something that moves step-by-step through a number of states, and exhibits transitions from one state to another (or the same) state. These transitions happen with a certain probability, and thus the process changes randomly over time. What makes a Markov chain so special, is that the so-called **Markov property** holds:

"The future, given the present, does not depend on the past."

We will consider (a simplified version of) the weather as a Markov chain, using the following assumptions.

- If it is sunny today, it will also be sunny tomorrow with probability 90%, but rainy with 10% probability;
- If it is rainy today, it will be sunny tomorrow with 40% probability, but it will remain rainy with 60% probability.

This defines a Markov chain with two possible states: 'Sunny' and 'Rainy'. It is clear that the Markov property holds, as the weather for tomorrow only depends on the weather today.
We can summarise the situation in a graph, as in Figure 3.2.1.
We can also define the Markov chain by writing down a **one-step transition matrix** P. If we associate the rows and columns of P with the possible states, then element (i, j) is the transition probability from state i to state j. If we let the first row and first column be 'Sunny', then we have that

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.4 & 0.6 \end{bmatrix}$$

For example, $P_{1,2} = 0.1$ so if it is sunny today, it will be rainy tomorrow with 10% probability.
Possible questions that we could ask ourselves are:

- What is the probability, if it is rainy today, that it will be sunny the day after tomorrow?
- If it is sunny on Monday, how many sunny days will we have on average from Monday until (and including) Thursday?

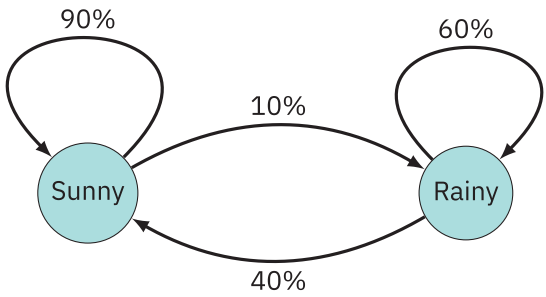


Figure 3.2.1. The weather as a Markov chain with two possible states: `Sunny' and `Rainy'. The transition probabilities are denoted on the edges.

- What percentage of time will it be sunny in the long run?

Using our probability matrix P, we can answer each of these questions.

Question. What is the probability, if it is rainy today, that it will be sunny the day after tomorrow?

Suppose that it is rainy today. If it is sunny tomorrow (with probability 0.4), then the day after tomorrow it is sunny with probability 0.9. The event of being sunny tomorrow and then also sunny the day after tomorrow happens with probability $0.4 \cdot 0.9 = 0.36$. If it is rainy tomorrow (with probability 0.6), then the day after tomorrow it is sunny with probability 0.4. This event happens with probability $0.6 \cdot 0.4 = 0.24$. Since it will be either sunny or rainy tomorrow, we add these results to obtain the total probability that it will be sunny the day after tomorrow: $0.36 + 0.24 = 0.60$.

But what if we wanted to know the probability for eight days from now? We could reason in the same way, but we now have a lot more combinations to consider. To simplify the calculations, we can use matrix multiplication. We have that

$$\begin{aligned} P^2 &= \begin{bmatrix} 0.9 & 0.1 \\ 0.4 & 0.6 \end{bmatrix} \cdot \begin{bmatrix} 0.9 & 0.1 \\ 0.4 & 0.6 \end{bmatrix} \\ &= \begin{bmatrix} 0.9 \cdot 0.9 + 0.1 \cdot 0.4 0.9 \cdot 0.1 + 0.1 \cdot 0.6 \\ 0.4 \cdot 0.9 + 0.6 \cdot 0.4, 0.4 \cdot 0.1 + 0.6 \cdot 0.6 \end{bmatrix} = \begin{bmatrix} 0.85 & 0.15 \\ 0.6 & 0.4. \end{bmatrix} \end{aligned}$$

This is the **two-step transition matrix**, and can be read in the same way as P. Note that we obtained the probability from rainy today to sunny the day after tomorrow, $P^2_{2,1} = 0.6$, with the exact same calculations as before.

We now can easily find the probability, if it is rainy today, that it will be sunny in eight days from now by calculation P^8. We have that

$$P^4 = P \cdot P \cdot P \cdot P = P^2 \cdot P^2 = \begin{bmatrix} 0.85 & 0.15 \\ 0.6 & 0.4. \end{bmatrix} \cdot \begin{bmatrix} 0.85 & 0.15 \\ 0.6 & 0.4. \end{bmatrix} = \begin{bmatrix} 0.8125 & 0.1875 \\ 0.75 & 0.25 \end{bmatrix}$$

and

$$P^8 = P^4 \cdot P^4 = \begin{bmatrix} 0.8125 & 0.1875 \\ 0.75 & 0.25 \end{bmatrix} \cdot \begin{bmatrix} 0.8125 & 0.1875 \\ 0.75 & 0.25 \end{bmatrix} = \begin{bmatrix} 0.801 & 0.199 \\ 0.797 & 0.203 \end{bmatrix}.$$

We thus find that the probability from rainy today to sunny eight days from now is 0.797, in a much simpler way than by mental calculation.

Question. If it is sunny on Monday, how many sunny days will we have on average between Monday until (and including) Thursday?

We know that the matrix for Tuesday is given by P, the matrix for Wednesday by P^2 and the matrix for Thursday by P^3. We have that

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.4 & 0.6 \end{bmatrix}, \quad P^2 = \begin{bmatrix} 0.85 & 0.15 \\ 0.6 & 0.4. \end{bmatrix}, \quad P^3 = \begin{bmatrix} 0.825 & 0.175 \\ 0.7 & 0.3 \end{bmatrix}.$$

Another way to think of the transition probabilities in the matrix are as average fractions. Saying that the probability of sunny today to sunny tomorrow is 0.9, is the same as saying that if it is sunny today, then tomorrow on average we have 0.9 sunny days. To find the average occurrences, we thus have to add the transition matrices. We see that

$$P + P^2 + P^3 = \begin{bmatrix} 0.9 & 0.1 \\ 0.4 & 0.6 \end{bmatrix} + \begin{bmatrix} 0.85 & 0.15 \\ 0.6 & 0.4. \end{bmatrix} + \begin{bmatrix} 0.825 & 0.175 \\ 0.7 & 0.3 \end{bmatrix} = \begin{bmatrix} 2.575 & 0.425 \\ 1.7 & 1.3 \end{bmatrix}.$$

If it is sunny on Monday, then we will have on average 2.575 sunny days between Tuesday until (and including) Thursday.

Question. What percentage of time will it be sunny in the long run?

Consider the different powers of P that we have calculated so far:

$$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.4 & 0.6 \end{bmatrix}, \quad P^2 = \begin{bmatrix} 0.85 & 0.15 \\ 0.6 & 0.4. \end{bmatrix}, \quad P^3 = \begin{bmatrix} 0.825 & 0.175 \\ 0.7 & 0.3 \end{bmatrix},$$
$$P^4 = \begin{bmatrix} 0.8125 & 0.1875 \\ 0.75 & 0.25 \end{bmatrix}, \quad P^8 = \begin{bmatrix} 0.801 & 0.199 \\ 0.797 & 0.203. \end{bmatrix}$$

It seems that there is a pattern forming. If we would continue indefinitely, the powers of the matrix will converge to a limiting matrix, given by

$$P^\infty = \begin{bmatrix} 0.8 & 0.2 \\ 0.8 & 0.2 \end{bmatrix}.$$

As the column for ‘Sunny’ has both elements equal to 0.8, we see that in the long run it will be sunny 80% of the time.

(The fine print for this limiting matrix is that 0.8 and 0.2 form the normalised left eigenvector of P corresponding to eigenvalue 1. This is a bit too complex to get into at the moment, but it suffices to say that this limiting matrix can be obtained using P, so we are not just guessing that eventually we will obtain the numbers 0.8 and 0.2.)

Since it isn't always either rainy or sunny, we could also consider more realistic models for the weather. Consider the Markov chain in Figure 3.2.2, where we now consider four states. Consequently, the transition matrix P becomes a 4×4-matrix. Though the model may be more realistic, it will also be more computationally complex, i.e., the number of calculations increase.

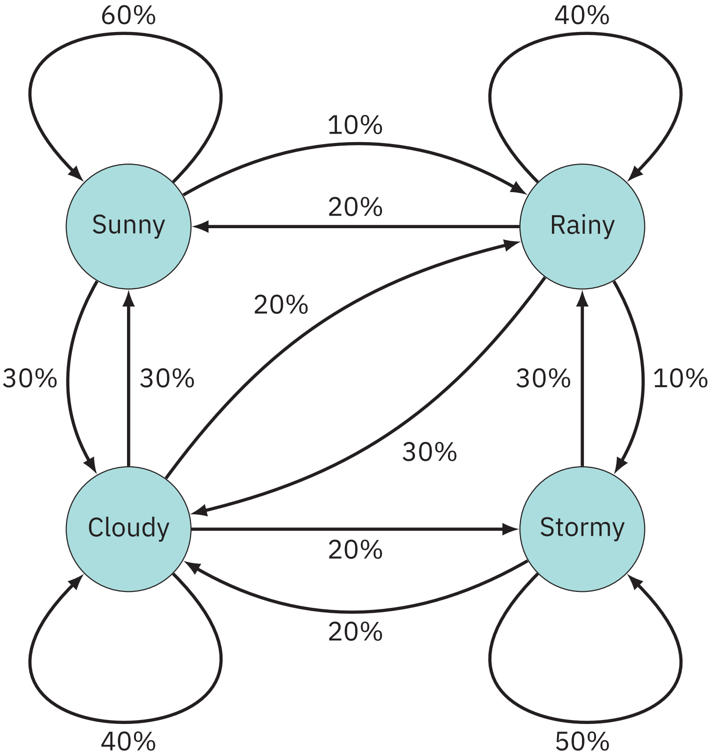


Figure 3.2.2. A Markov chain of the weather with four possible states.

The corresponding transition matrix and limiting matrix are

$$P = \begin{bmatrix} 0.6 & 0.1 & 0.3 & 0 \\ 0.2 & 0.4 & 0.3 & 0.1 \\ 0.3 & 0.2 & 0.4 & 0.1 \\ 0 & 0.3 & 0.2 & 0.5 \end{bmatrix}, \quad P^{\infty} = \begin{bmatrix} 0.351 & 0.220 & 0.321 & 0.108 \\ 0.351 & 0.220 & 0.321 & 0.108 \\ 0.351 & 0.220 & 0.321 & 0.108 \\ 0.351 & 0.220 & 0.321 & 0.108 \end{bmatrix}.$$

We can conclude the following from this limiting matrix. If it is cloudy today, an arbitrary day in the future will be cloudy with probability 32.1%. Note that in fact, it does not matter what the weather today is. We can also conclude that if it is cloudy today, in the long run 32.1% of all days will be cloudy. Again, it does not matter what the weather today is.
The dependence on today's state of weather does become apparent when we adapt the Markov chain, as in Figure 3.2.3. We now have two *absorbing states*, states that the system can never leave. Indeed, we see that if it is cloudy (or stormy) today, then with probability 100% the weather will be cloudy (or stormy) again tomorrow. The corresponding transition matrix and limiting matrix are

$$P = \begin{bmatrix} 0.6 & 0.1 & 0.3 & 0.0 \\ 0.2 & 0.4 & 0.3 & 0.1 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}, \quad P^{\infty} = \begin{bmatrix} 0.00 & 0.00 & 0.95 & 0.05 \\ 0.00 & 0.00 & 0.82 & 0.18 \\ 0.00 & 0.00 & 1.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 1.00 \end{bmatrix}.$$

From this limiting matrix we can conclude the following. If it is cloudy today, an arbitrary day in the distant future will be cloudy with probability 100%. However, if it is sunny today, an arbitrary day in the distant future will be cloudy with probability 95%. We thus see that in this case, the exact probability depends on the weather today.
Similarly, we cannot say what percentage of the time it will be cloudy in the future, as this also depends on the weather today. At most, we can say that if it is cloudy today, in the long run 100% of all days will be cloudy. However, this is not a complete surprise as 'Cloudy' is an absorbing state. So if we encounter one cloudy day, we will forever encounter cloudy days. Luckily, the actual weather is more complicated than this model!
Finally, consider the Markov chain in Figure 3.2.4. We consider two consecutive powers of the transition matrix:

$$P^{101} = \begin{bmatrix} 0 & 0.464 & 0.536 & 0 \\ 0.454 & 0 & 0 & 0.546 \\ 0.454 & 0 & 0 & 0.546 \\ 0 & 0.464 & 0.536 & 0 \end{bmatrix}, \quad P^{102} = \begin{bmatrix} 0.454 & 0 & 0 & 0.546 \\ 0 & 0.464 & 0.536 & 0 \\ 0 & 0.464 & 0.536 & 0 \\ 0.454 & 0 & 0 & 0.546 \end{bmatrix}.$$

If we continue on with matrix multiplication, we observe the same pattern in the powers of P. Though the value of the non-zero elements changes, the placement of the non-zero elements is constantly alternating according to this pattern. We thus do not find a limiting

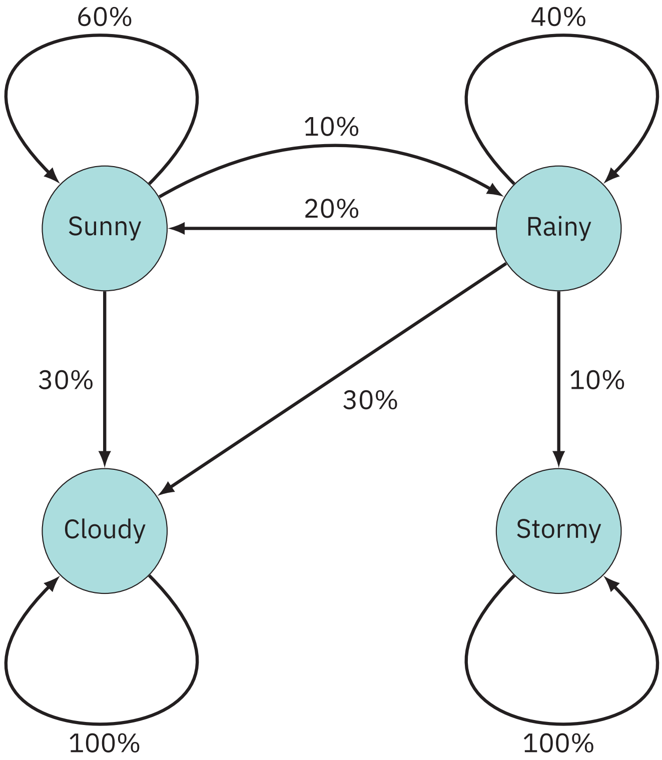


Figure 3.2.3. A Markov chain of the weather with two absorbing states, i.e., states that are impossible to leave.

matrix. However, we do see that the elements in the matrix converge to a limiting value. By averaging the values over the column, we could conclude something about the behaviour in the long run. If it is cloudy today, an arbitrary day in the distant future will be cloudy with probability 26.8%. It does not really depend on the weather today. In the same vain, if it is sunny today, it will be cloudy 26.8% of the time in the long run. It also does not really depend on the weather today.

3.2.2. Google's search engine

We now consider an application of Markov chains in Google's search engine. We can think of this search engine as a mathematical function $f(x) = y$, where

- x is the search query;
- y contains all websites where 'x' appears, presented in a certain sequence.

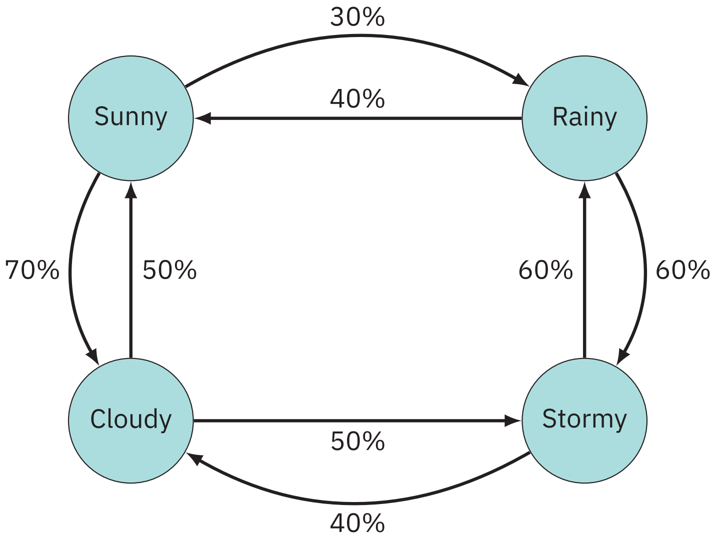


Figure 3.2.4. A Markov chain of the weather with four possible states.

The function f is determined using Google's **PageRank algorithm**, which searches through a huge network. The idea behind the PageRank algorithm is as follows. We construct a network in which the nodes are websites, and directed edges are hyperlinks pointing to a website. We consider each incoming hyperlink as a 'vote' for a website. Consider for example the graph in Figure 3.2.5. Then websites A and B each get 1 vote, and websites C and D each get 2 votes. If we consider the fraction per website, then we find

$$(0.167, 0.167, 0.33, 0.33).$$

If we rank the websites according to these fractions, websites C and D would be on top and websites A and B would be on the bottom.

But this only counts the absolute number of votes that a website receives, without taking into account whether these votes come from important websites or not so important websites. Intuitively, we would consider votes of important websites to have more worth than votes from not so important websites. To incorporate this difference of importance, the PageRank algorithm is as follows.

(1) Start with an arbitrary distribution of votes.
(2) Divide the votes of each website equally over the websites where the websites links to.
(3) Repeat Step 2 until the distribution of votes does not change anymore.

Let's apply this algorithm to our example. We start with $(0.25, 0.25, 0.25, 0.25)$. In the second step, the division of votes is:

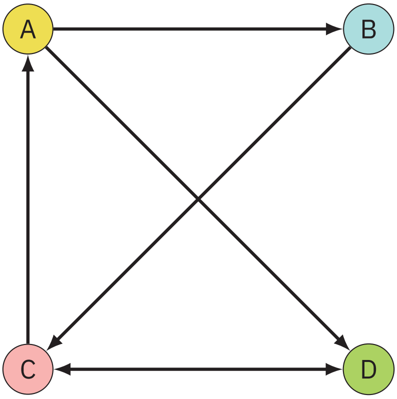


Figure 3.2.5. A network of four websites, with hyperlinks between them.

- A receives 0.125 votes from C;
- B receives 0.125 votes from A;
- C receives 0.25 votes from B and 0.25 votes from D;
- D receives 0.125 votes from A and 0.125 votes from C.

The new distribution of votes is then

$$(0.125, 0.125, 0.5, 0.25)$$

We repeat the second step, and the division of votes is:

- A receives 0.25 votes from C;
- B receives 0.0625 votes from A;
- C receives 0.125 votes from B and 0.25 votes from D;
- D receives 0.0625 votes from A and 0.25 votes from C.

The new distribution of votes is then

$$(0.25, 0.0625, 0.375, 0.3125).$$

If we continue on like this, we see that after 43 iterations, the distribution of votes does not change any more and the final distribution is equal to

$$(0.2, 0.1, 0.4, 0.3).$$

The final ordering of the websites is hence C, D, A, B.

How does this algorithm relate to Markov chains? Let the state space consist of the websites A, B, C and D. Let the transition probabilities from i to j be the percentage of its vote that website i gives to website j. For example, A has two outgoing links, one to B and one to D. Then it will always give half of its votes (50%) to B and half of its votes to D.

The algorithm is hence actually a Markov chain, with transition matrix

$$P = \begin{bmatrix} 0.0 & 0.5 & 0.0 & 0.5 \\ 0.0 & 0.0 & 1.0 & 0.0 \\ 0.5 & 0.0 & 0.0 & 0.5 \\ 0.0 & 0.0 & 1.0 & 0.0 \end{bmatrix}.$$

If we determine the limiting matrix of this Markov chain, we find

$$P^{\infty} = \begin{bmatrix} 0.2 & 0.1 & 0.4 & 0.3 \\ 0.2 & 0.1 & 0.4 & 0.3 \\ 0.2 & 0.1 & 0.4 & 0.3 \\ 0.2 & 0.1 & 0.4 & 0.3 \end{bmatrix}.$$

It turns out that the PageRank algorithm is essentially looking for the rows of P^{∞}. Where it gets complicated, is that Google indexes not just 4 websites, but about 50 billion of them. In addition to that, the network is also constantly changing. Websites appear, change and disappear, so the network is more stochastic in nature. Finally, as users we expect an answer in just a few milliseconds, so we also need a fast algorithm to efficiently search through a network of such large size. These are all important considerations to be made when researching efficient algorithms.

3.3. Networks in biology

The following is a research work by Birgit Sollie. A simple model for chemical reactions is studied by constructing directed bipartite graphs. Several networks are constructed, in which some nodes represent the number of molecules: a chemical reaction is detected by looking at the changes from one network to another. The number of molecules can also be recorded in a data sequence, where each chemical reaction leads to a new data point. To quickly determine new data points, the reactions is presented in a matrix form.
Relevant concepts are directed and bipartite graphs (Section 1.1) and matrix multiplication (Section 1.3).

The human body is composed of many different types of cells. Together, they create tissues, organ systems and everything in the human body. A cell is the smallest unit of life. As everything is composed of it, they are often called the ‘building blocks of life’. All cells have a *membrane* that envelops the cell. Within the membrane is the cytoplasm, a watery liquid. There are about 50 thousand billion cells in our body. More than half of our body is water, for example lung cells consist of 90% water. Bone cells however consist only of 10-20% water. Within a certain cell several types of molecules can be found, such as nutrients, oxygen or proteins. A collision can occur between two molecules, with three possible results:

- the two molecules stick together;
- the two molecules exchange something;
- nothing happens.

If two molecules stick together or exchange something, we call this a chemical reaction.

3.3.1. Network of chemical reactions

We can use network theory to model such chemical reactions in cells. To illustrate this, we consider an example.

EXAMPLE 3.3.1. Consider four molecules: B, G, P and R. A representation of the four molecules in a cell is given in Figure 3.3.1. The following chemical reactions can occur.

1. If B and G collide, then a molecule P is formed.
2. If B and R collide, then a molecule G is formed.
3. If two molecules P collide, then a molecule R is formed.

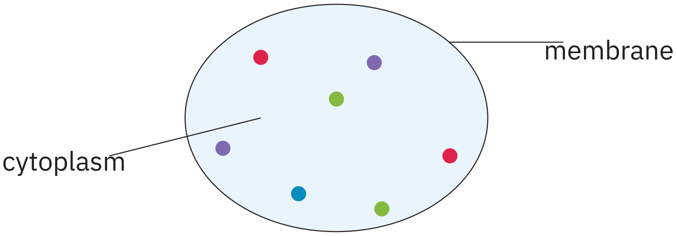


Figure 3.3.1. A graphical representation of a cell with four molecules B, G, P and R, each denoted by a different colour.

We can write this as

$$\mathsf{B} + \mathsf{G} \to \mathsf{P}, \quad \mathsf{B} + \mathsf{R} \to \mathsf{G}, \quad 2\mathsf{P} \to \mathsf{R}.$$

We can also model these reactions in a network, by constructing a directed bipartite graph, see Figure 3.3.2. In this network, each of the molecules B, G, P and R is assigned a node. Additional to that, we have three artificial nodes (here denoted by rectangles) that symbolize the chemical reactions. Each black arrow is the input for a reaction. The output is given by a red arrow. The edge from P to 3 has a weight of 2, since there should be two molecules P to produce a molecule R.

Note that this is indeed a bipartite graph, since we can separate the nodes into the sets $V_1 = \{\mathsf{B, G, P, R}\}$ and $V_2 = \{1, 2, 3\}$ and there are no arrows between nodes of one set.

Perhaps a better way to draw this network is as in Figure 3.3.3.

We now have a fixed network, containing all reactions that can happen in a cell. The number of molecules changes over time in the cell, so reactions will take place with a certain probability. These probabilities are often unknown, but we would like to know them.

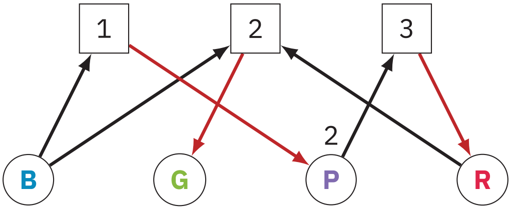


Figure 3.3.2. A network of three chemical reactions, involving four molecules.

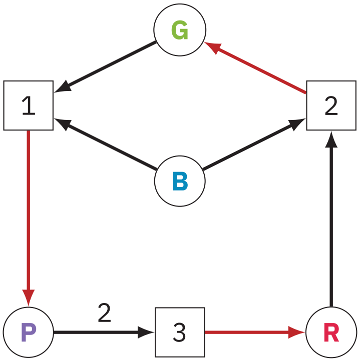


Figure 3.3.3. A different representation of the network in Figure 3.3.2.

3.3.2. Networks with numbers

To determine these probabilities, we consider our network with numbers. In each node, we put the number of molecules instead of the name of the molecule (where we use the same colours, so as not to confuse). Consider for instance the two networks in Figure 3.3.4, where a reaction takes place from the first network to the second. Can you see which reaction this is?

We constantly count the number of molecules in the cell. If the reaction in Figure 3.3.4 happens, then we can write this down as the sequence

$$(\mathbf{8}, \mathbf{5}, \mathbf{2}, \mathbf{0}), (\mathbf{8}, \mathbf{5}, \mathbf{0}, \mathbf{1}), \ldots$$

This is the data obtained from the cell. Using this data, we can estimate the probabilities that we are looking for.

There may occur other chemical reactions after 2P → R. Consider Figure 3.3.5. Which reactions took place from the first network to the second, and how many of these reactions?

We see that the reactions in Figure 3.3.5 are consecutively

$$2\mathrm{P} \to \mathrm{R}, \quad \mathrm{B} + \mathrm{R} \to \mathrm{G}, \quad \mathrm{B} + \mathrm{G} \to \mathrm{P}.$$

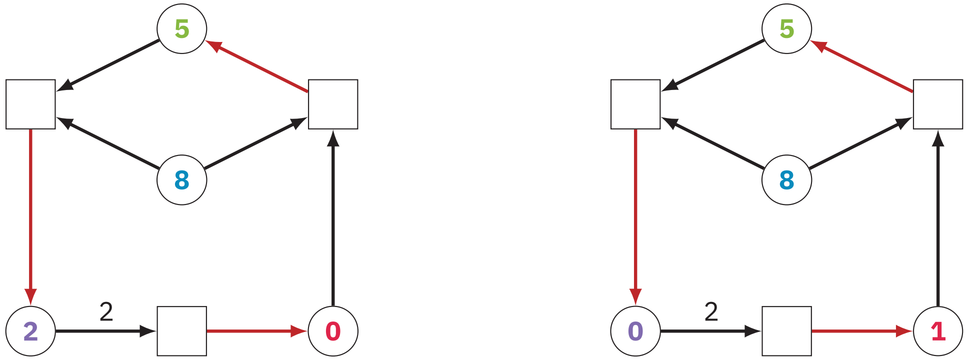


Figure 3.3.4. The chemical reaction 2P → R occurs from the first network to the second network.

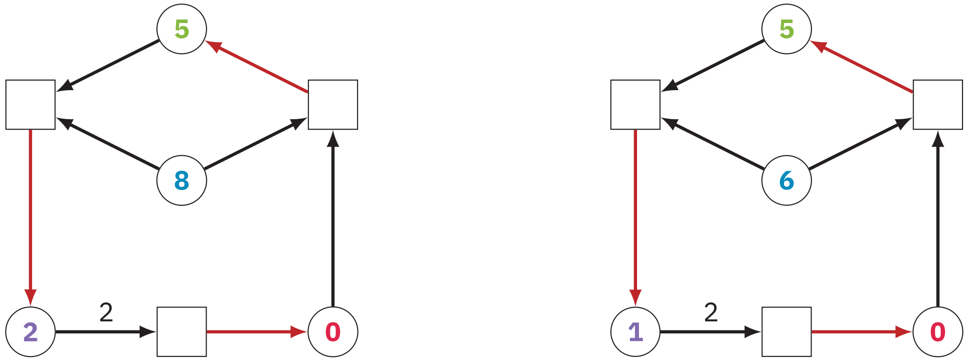


Figure 3.3.5. A number of reactions occur from the first network to the second network.

Our data sequence then becomes

$$(8, 5, 2, 0), (8, 5, 0, 1), (7, 6, 0, 0), (6, 5, 1, 0), \ldots$$

Starting with the same number of molecules, other chemical reactions can happen as well, see Figure 3.3.6. We have consecutive reactions

$$\mathrm{B} + \mathrm{G} \to \mathrm{P}, \quad 2\mathrm{P} \to \mathrm{R}, \quad \mathrm{B} + \mathrm{R} \to \mathrm{G}$$

with corresponding data sequence

$$(8, 5, 2, 0), (7, 4, 3, 0), (7, 4, 1, 1), (6, 5, 1, 0), \ldots$$

We now end up at the same configuration as in Figure 3.3.5, but the chemical reactions that occurred were different. We thus see that, starting from the configuration $(8, 5, 2, 0)$, a lot

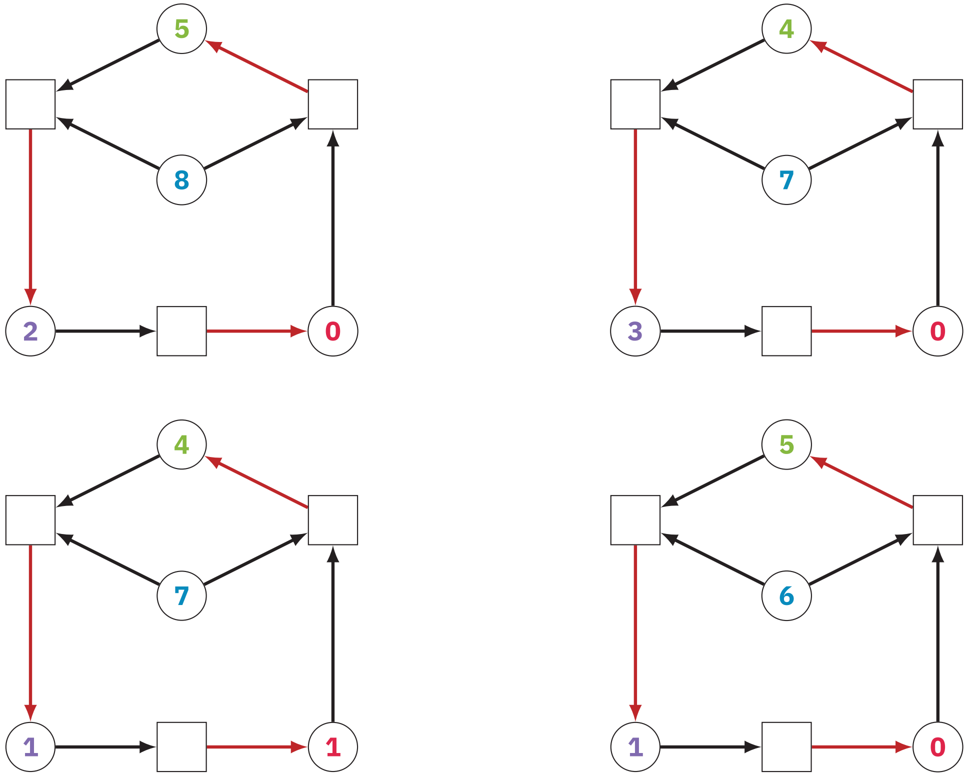


Figure 3.3.6. Chemical reactions occur between one network to the other, from left to right, starting at the top left and ending at the bottom right.

of different scenarios can unfold. What will happen, will ultimately depend on the probabilities. We would also like to know what would happen after a long time. Will there be an equilibrium reached? Or will a molecule run out, so stop existing? But then we would have to construct a lot of these networks, to obtain data in the long run.

A way in which we can quickly determine data sequences is using matrix multiplication. We first construct matrices associated with the network. As we have four molecules and three possible reactions, these will be 4×3-matrices. We write down an out-matrix O and an in-matrix I. Use the ordering of the molecules and the numbering of the reactions as in Example 3.3.1. We have that

$$O = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \\ 0 & 1 & 0 \end{bmatrix}.$$

In the out-matrix, the element (i, j) is equal to 1 if molecule i is an output of reaction j, and otherwise 0. In the in-matrix, the element (i, j) is equal to 1 if molecule i is an input of reaction j, and otherwise 0.

Taking the difference of the out- and in-matrix now gives us a matrix that qualifies how the chemical reactions change the number of molecules in the network. Let M be the difference of the out- and in-matrix, then

$$M = O - I = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & -2 \\ 0 & -1 & 1 \end{bmatrix}.$$

If we now read along the columns of M, we see the changes that the reactions make. For example, we see that the first reaction takes one molecule B and one molecule G and produces a molecule P.

Let b be the initial configuration, in vector form. Let r be the vector of length 3, in which element i is equal to 1 if reaction i occurs and otherwise 0. We then find the end configuration e by calculating

$$e = b + Mr.$$

Consider the initial state (**8**, **5**, **2**, **0**). We want to know what happens when both the first reaction and the third reaction occur. Then we calculate

$$\begin{bmatrix} \mathbf{8} \\ \mathbf{5} \\ \mathbf{2} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -1 & -1 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & -2 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{7} \\ \mathbf{4} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}.$$

We thus immediately find the end configuration, for which we had to draw four networks in Figure 3.3.6. Using this matrix multiplication is thus a efficient way to determine what happens when certain reactions occur.

3.4. The phenomenon of synchronisation

The following is a research work by Janusz Meylahn. Synchronisation processes on complex networks are investigated in detail. Particular attention is given to the noisy Kuramoto model, in which the behaviour of a set of coupled oscillators is described. The evolution of the phase of the oscillators is described using differential equations. To solve the equations, order parameters are introduced to describe the dynamics as the evolution of a density. A critical threshold is then found, i.e. a value of the parameter where below this value the system behaves significantly different than above.

Relevant concepts are trigonometric functions, complex numbers and Euler's formula (Section 1.3).

We can consider many real-world processes by constructing a network and observing the process on the network. Think for instance of

- the spreading of a rumour;
- searching for information on the internet;
- the formation of polymers;
- the synchronization of neurons firing in the brain.

There are many different processes and corresponding networks that we may consider. In some processes, we may be interested in how different parts of the network interact with each other. In other processes, we are more interested in (shortest) paths in the network. A process can either take place on each node, or it can move on the network. We could also consider processes in a discrete space or a continuous space. Finally, we can consider a dynamic or a static network. A dynamic network is ever evolving, while a static network is fixed.

In this talk, we will focus on synchronised processes. Synchronisation is the coordination of events to operate a system in unison. Simply put, multiple parts work together to produce one whole. Examples of synchronisation are:

- fireflies flashing in the jungle;
- electricity generators on a power grid;
- an audience clapping after a concert;
- neurons firing in the brain;
- the gravitational synchronisation of meteors.

(Tip: if you are looking for your next popular science book to ready, try *Sync: The Emerging Science of Spontaneous Order* by Steven Strogatz.) In the real world we thus observe many examples of synchronisation. Can we say something about this synchronisation using a mathematical model?

We first have to consider what a **stochastic process** is. If we would intuitively describe the ingredients needed for a stochastic process, we could say that we would need some randomness, a recipe describing the situation as a function of said randomness, and some idea of time. A stochastic process more or less represents the value of some system, which is randomly changing over time.

EXAMPLE 3.4.1. Consider a game between two people, where a coin is flipped. If heads is up, then you get 1 euro from your opponent. If tails is up, then you have to pay 1 euro to your opponent. Then the outcomes of the coin-flip is a stochastic process. If we flip the coin 8 times, a sequence could be

$$\omega = \{H, T, T, T, T, H, T, H\}.$$

See Section 3.4.4 for an exercise on the coin-flip game.

3.4.1. (Noisy) Kuramoto model

The Kuramoto model is a mathematical model used to describe synchronization. It is a model for the behaviour of a set of coupled oscillators. An oscillator is a system that behaves according to a repetitive pattern. Oscillators occur in real-life, think for example of the beating of a human heart or the swinging of a pendulum clock.

In this model we consider N oscillators, with $\theta_i(t)$ the phase of the i-th oscillator. These phases evolve over time, according to a system of *coupled stochastic differential equations*, given by

$$d\theta_i(t) = \frac{K}{N}\sum_{j=1}^{N} \mathsf{sin}[\theta_j(t) - \theta_i(t)]dt + DdW_i(t).$$

Here, $K \in (0, \infty)$ is the interaction strength, $D \in (0, \infty)$ is the noise strength and $(W_i(t))_{t\geq 0}$ are noise processes. This equation qualifies how the phase of the i-th oscillator changes in time, dependent on the phases of the other oscillators and with some extra noise term. The noise process is usually a random process, which is why this is called a *stochastic* differential equation.

The network behind the Kuramoto can be seen in Figure 3.4.1, where the nodes are the oscillators. The edges between oscillators denote the dependence seen in the differential equations. Since each oscillator depends on each other oscillator, we obtain a *complete graph*, i.e., every node is connected to every other node.

We want to solve this set of coupled differential equations, to obtain an expression for $\theta_i(t)$. We are able to do this as N grows very large, by rewriting the differential equations. Define the order parameters $r_N(t)$ and $\psi_N(t)$ by

$$r_N(t)e^{i\psi_N(t)} = \frac{1}{N}\sum_{j=1}^{N} e^{i\theta_j(t)}.$$

Here, i is the complex number and e is Euler's number. The order parameters represent the following:

- $r_N(t)$ is the synchronization level of the population of oscillators, in other words, how synchronised are the oscillators at time t;
- $\psi_N(t)$ denotes the average phase of the oscillators.

In Figure 3.4.2, an example is shown of two phase distributions, for different values of r and ψ. In the left image, the phases are rather drawn out, leading to a small synchronization level. In the right image, the phase synchronization level is almost 1.

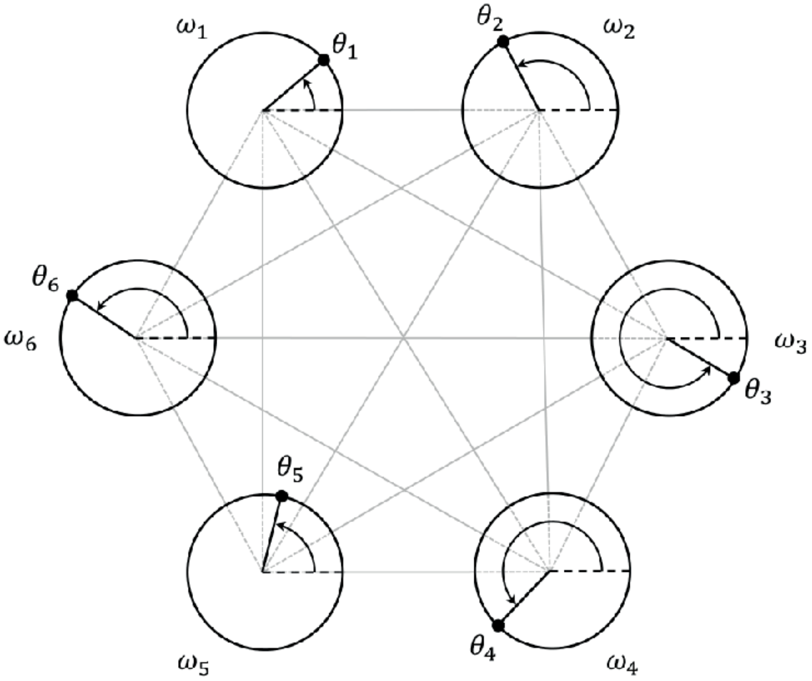


Figure 3.4.1. A network of the Kuramoto model for $N = 6$.

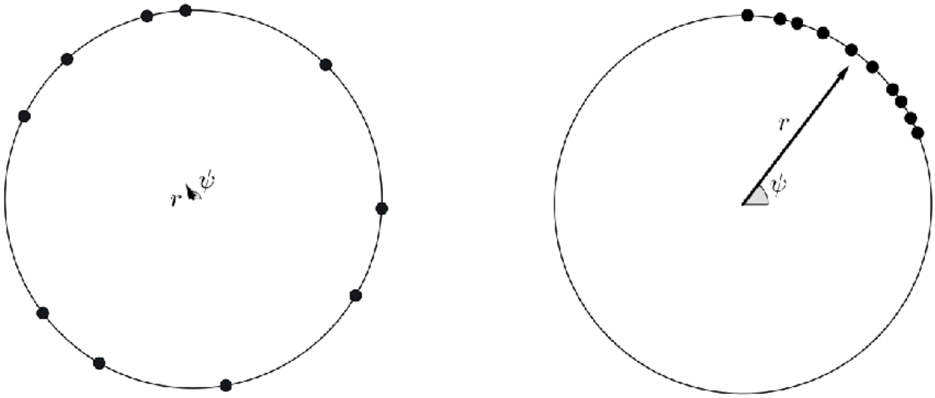


Figure 3.4.2. Phase distributions with $r = 0.095$ (left) and $r = 0.929$.

We can rewrite the differential equations using the order parameters. At the end of the talk, an exercise (reference) will go into detail. For now, we have the result

$$d\theta_i(t) = Kr_N(t)\,\text{sin}[\psi_N(t) - \theta_i(t)]dt + DdW_i(t).$$

Now the explicit dependence on the other oscillators has disappeared into the implicit dependence on $\psi_N(t)$. As N gets ever larger, we can describe the evolution of the oscillators as the evolution of a *density*. More on densities can be found in the exercises (reference).

We want to understand the long time behaviour of the oscillators. Since we can describe the evolution of the oscillators as the evolution of a density, we then want to understand the long time behaviour of the density. In particular, does the density of the system stop evolving at some point, i.e., is there some limiting state?

It turns out that there are two possible limiting states. There exists a *critical threshold* K_c such that

(1) for $K < K_c$ the system relaxes to an unsynchronised state $r = 0$;
(2) for K> K_c the system relaxes to a partially synchronised state $r > 0$.

The threshold K_c is the turning point between unsynchronised and synchronised. After K_c, the system is partially synchronised. The closer r is to 1, the more synchronised the system is. In this model we find that $K_c = 2$, as can be seen in Figure 3.4.3.

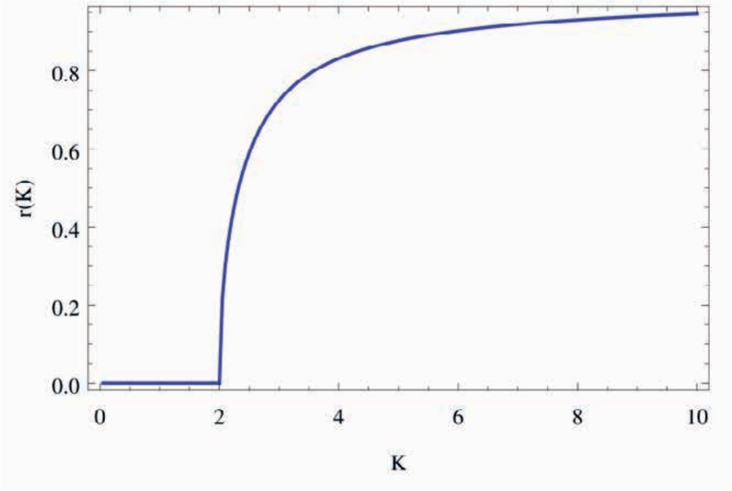


Figure 3.4.3. The order parameter r as a function of K. The critical threshold K_c is equal to 2.

3.4.2. 'Complex Network'

A real-life application of the theory of synchronisation networks can be found when considering the suprachiasmatic nucleus (SCN). The SCN is a tiny region of the brain, that is responsible for controlling circadian rhythms. These are biological processes that display an oscillation of about 24 hours. The SCN generates neuronal and hormonal activities that regulate many different body functions in such a 24-hour cycle, using around 20.000 neurons. If we consider a network of the SCN, then this is indeed a complex network.
Through research, a few things can be said about this network. The SCN has a strong community structure, and interaction between the communities is negative. (A representation of two interacting communities is given in Figure 3.4.4.) This is the case in all mammals. The SCN is a rich and robust network, and the structure of the network may play a role in this. Malfunctioning of the network can cause health problems, ranging from epilepsy to narcolepsy.

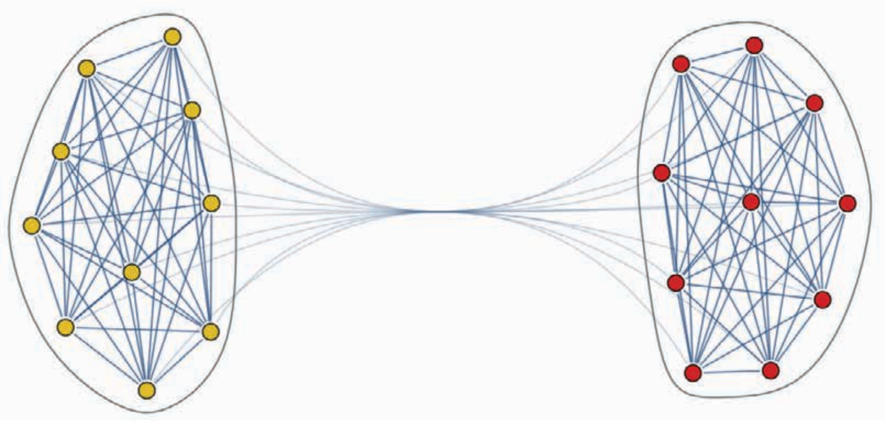


Figure 3.4.4. A representation of two interacting communities in the SCN.

3.4.3. Conclusion

There are many different types of processes to study on networks. Many real-world processes can be modelled using networks, so networks really play an important role. An particularly interesting example of a process is synchronisation. This occurs in many biological processes, some of which in the human body. For instance, synchronisation occurs in neurons. We could tell neuroscientist something of value by using mathematical research.

3.4.4. Exercises

Stochastic Processes

EXERCISE 6.

(a) Given a sequence of coin-flip outcomes ω, construct a function $X_n(\omega)$ representing your wealth after the n-th flip.

(b) Calculate the probability of having three euros after five coin flips, given that the coin is head with probability $1/2$.

Rewriting the Kuramoto model

EXERCISE 7.

(a) Write the following complex numbers in polar coordinate form:

1. $-\sqrt{3}+3i$
2. $-1-i$
3. $\sqrt{3}-i$
4. $-2+5i$

(b) Euler's formula for complex numbers is

$$e^{i\theta} = \cos\theta + i\sin\theta.$$

Use this formula and that $e^{a+b} = e^a e^b$ to 'prove' the trigonometric identities

$$\cos(a+b) = \cos a \cos b - \sin a \sin b$$
$$\sin(a+b) = \cos a \sin b + \sin a \cos b$$

(c) Use Euler's formula to obtain expressions for $\cos\theta$ and $\sin\theta$ in terms of complex exponentials.

EXERCISE 8 (Kuramoto's trick).

(a) Rewrite the order parameter

$$r_N(t)e^{i\psi_N(t)} = \frac{1}{N}\sum_{j=1}^{N} e^{i\theta_j(t)}$$

using Euler's formula to obtain an expression containing $\sin[\theta_j(t) - \theta_i(t)]$.

(b) Use the fact that two complex numbers are equal if both their real parts and imaginary parts are equal, to obtain two equations for real numbers.

(c) Use the appropriate equation from the two equations from (b) to plug into the stochastic differential equation:

$$d\theta_i(t) = \frac{K}{N}\sum_{j=1}^{N} \sin[\theta_j(t) - \theta_i(t)] + DdW_i(t).$$

The density of oscillators

EXERCISE 9. The steady-state density for the oscillators $p(\theta)$ is a function describing what fraction of the oscillators have a phase angle in a small interval around θ. It turns out that this steady-state density has to satisfy the differential equation

$$\frac{dp(\theta)}{d\theta} = -2Kr \sin \theta p(\theta).$$

(a) What form should $p(\theta)$ have to satisfy this equation?

For the density to be consistent, it needs to take the same value at $\theta = 0$ and $\theta = 2\pi$. In addition to his, the density should be a *probability density* which means that integrating it over θ should give 1, i.e.

$$\int_0^{2\pi} p(\theta) d\theta = 1.$$

(b) Write down an exact expression for the steady-state density of the oscillators.

CHAPTER 4

Solutions to exercises

SOLUTION 1 (Chapter 1, Section 1.2.4).

(a) 2 police officers, since the graph is a tree.

(b) 3 police officers, in the tree-like parts we only need two, but on the cycle we need three.

(c) 3 police officers, in the same way as (b).

(d) 4 police officers, you can trap the robber with one on the middle node and three on the side.

(e) 3 police officers, in the same way as (b).

(f) 3 police officers, you can then trap the robber in a corner of the graph.

SOLUTION 2 (Chapter 1, Section 1.2.4). Note: we assume that police officers use helicopters to chase the robber and can thus jump to any node. We also assume that the robber only moves to a free neighbouring node if he sees the helicopter coming to land on a node he is on. (What will change when we have a 'smart' robber, that can also move before the helicopter lands on his node?)

(a) No, we need at least three police officers for the first robber and also at least three for the second robber.

(b) Yes, focus on the robber on node 13. Then we can trap him using two police officers, and need a third one to catch him.

(c) With one additional police officer, both robbers can be caught.

(d) We trap the robber on node 13, by allocating two police officers to node 12 and node 10. We then wait for an additional officer. We can catch the other robber using the remaining three officers.

Place police officers on node 24, 28 and 23. Then the robber is forced to move to 20. Keep an officer on 23 and place the others on 22 and 20, then the robber is forced to move to 19. Keep an officer on 20 and place the others on 6 and 19, then the robber is forced to move to 18. Keep an officer on 19 and place the others on 17 and 18, then the robber is forced to move to 21. Keep an officer on 18 and use one of the others to catch the robber on 21.

SOLUTION 3 (Chapter 1, Section 1.3.9). We show this per element:

(a) $[(cA)^\top]_{ij} = [cA]_{ji} = c[A]_{ji} = c[A^\top]_{ij}$;

(b) $[(A+B)^\top]_{ij} = [A+B]_{ji} = [A]_{ji} + [B]_{ji} = [A^\top]_{ij} + [B^\top]_{ij} = [A^\top + B^\top]_{ij}$;

(c) $[(A^\top)^\top]_{ij} = [A^\top]_{ji} = [A]_{ij}$.

SOLUTION 4 (Chapter 2, Section 3.1.1). For the graph in Figure 3.1.3, consider the communities and their corresponding modularity:

$$\begin{aligned} C_1 = \{1,2,3,4\}, &\quad M = 0, \\ C_1 = \{1,2\}, C_2 = \{3,4\}, &\quad M = -\frac{1}{6}, \\ C_1 = \{1\}, C_2 = \{2\}, C_3 = \{3\}, C_4 = \{4\}, &\quad M = -\frac{3}{2}. \end{aligned}$$

We do not improve if we consider communities, so there is no community structure. For the graph in Figure 3.1.4, consider the communities and their corresponding modularity:

$$\begin{aligned} C_1 = \{1,2,3,4,5,6\}, &\quad M = 0, \\ C_1 = \{1,2\}, C_2 = \{3,4\}, C_3 = \{5,6\}, &\quad M = \frac{4}{49}, \\ C_1 = \{1,2,3\}, C_2 = \{4,5,6\}, &\quad M = \frac{5}{14}. \end{aligned}$$

We improve if we consider communities, so there is some community structure.

SOLUTION 5 (Chapter 2, Section 2.3.4). The time schedule is as follows:

departure from Amsterdam	h:27
arrival in Hilversum	h:48
departure from Hilversum	h:49
arrival in Amersfoort	(h+1):02

For example, the train departs from Amsterdam at 09.27, arrives in Hilversum at 09.48, departs from Hilversum at 09.49 and arrives in Amersfoort at 10.02. A voltage g that also satisfies the requirements is given by

$$\begin{aligned} g(v_{15,Amd,D}, v_{15,Hvs,A}) &= 21 \\ g(v_{15,Hvs,A}, v_{15,Hvs,D}) &= 1 \\ g(v_{15,Hvs,D}, v_{15,Amf,A}) &= -47. \end{aligned}$$

SOLUTION 6 (Chapter 3, Section 3.4.4).

(a) We receive 1 euro if heads is up and pay 1 euro if tails is up. Then the sequences

$$\omega = \{H, T, T, H, T, T, \ldots\}, \quad \omega = \{+1, -1, -1, +1, -1, -1, \ldots\}$$

are equivalent. We now easily have that

$$X_n(\omega) = \sum_{k=1}^{n} \omega_k,$$

where ω_k is the k-th element of ω.

(b) The coin is either heads or tails, occurring with the same probability. Let A be the event that we have three euros after five coin clips. Then

$$\mathbb{P}(A) = \frac{\text{number of sequences of length 5, with four heads and one tail}}{\text{total number of sequences of length 5}}$$

is the probability that event A happens. Indeed, since we each time either win 1 euro or lose 1 euro, we have to win four times and lose once. The sequences in which this happens are

$$\{-1,1,1,1,1\}, \{1,-1,1,1,1\}, \{1,1,-1,1,1\}, \{1,1,1,-1,1\}, \{1,1,1,1,-1\}.$$

In each flip, there are two possible outcomes. It follows that the total number of sequences of length 5 is equal to $2^5 = 32$. We find that $\mathbb{P}(A) = \frac{5}{32}$.

(Note that we are in fact considering a binomial distribution throughout this exercise, where 'success' is defined as H, with probability $\frac{1}{2}$.).

SOLUTION 7 (Chapter 3, Section 3.4.4).

(a) Use the formulas in Table 1.3.1, Section 1.3.4 to go from Cartesian coordinates to polar coordinates.

1. $a = -\sqrt{3}$ and $b = 3$, so

$$\begin{aligned} r &= \sqrt{(-\sqrt{3})^2 + 3^2} = \sqrt{12} = 2\sqrt{3}, \\ \theta &= \text{arctan}\left(\frac{3}{-\sqrt{3}}\right) + \pi = \text{arctan}(-\sqrt{3}) + \pi = \frac{2}{3}\pi. \end{aligned}$$

2. $a = b = -1$, so

$$\begin{aligned} r &= \sqrt{(-1)^2 + (-1)^2} = \sqrt{2}, \\ \theta &= \text{arctan}\left(\frac{-1}{-1}\right) - \pi = \text{arctan}(1) - \pi = -\frac{3}{4}\pi. \end{aligned}$$

3. $a = \sqrt{3}$ and $b = -1$, so

$$\begin{aligned} r &= \sqrt{(\sqrt{3})^2 + (-1)^2} = \sqrt{4} = 2, \\ \theta &= \text{arctan}\left(\frac{-1}{\sqrt{3}}\right) = -\frac{1}{6}\pi. \end{aligned}$$

4. $a = -2$ and $b = 5$, so

$$\begin{aligned} r &= \sqrt{(-2)^2 + 5^2} = \sqrt{29}, \\ \theta &= \text{arctan}\left(\frac{5}{-2}\right) + \pi = -\,\text{arctan}\left(\frac{5}{2}\right) + \pi. \end{aligned}$$

(b) Apply Euler's formula to $\theta = a + b$:

$$e^{i(a+b)} = \cos(a+b) + i\sin(a+b).$$

We also have that

$$\begin{aligned} e^{i(a+b)} &= e^{ia}e^{iab} \\ &= (\cos a + i\sin a)(\cos b + i\sin b) \\ &= \cos a\cos b - \sin a\sin b + i(\cos a\sin b + \sin a\cos b). \end{aligned}$$

Two complex numbers are only equal when both their real and imaginary part are equal, so

$$\begin{aligned} \cos(a+b) &= \cos a\cos b - \sin a\sin b \\ \sin(a+b) &= \cos a\sin b + \sin a\cos b. \end{aligned}$$

(c) Using Euler's formula, we find that

$$\begin{aligned} e^{i\theta} &= \cos\theta + i\sin\theta \\ e^{-i\theta} &= \cos(-\theta) + i\sin(-\theta) = \cos\theta - i\sin\theta. \end{aligned}$$

It follows that

$$\cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2}, \quad \sin\theta = \frac{e^{i\theta} - e^{-i\theta}}{2i}.$$

SOLUTION 8 (Chapter 3, Section 3.4.4).

(a) First, multiply the equation on both sides with $e^{-i\theta_i(t)}$ to obtain:

$$r_N(t)e^{i(\psi_N(t)-\theta_i(t))} = \frac{1}{N}\sum_{j=1}^{N} e^{i(\theta_j(t)-\theta_i(t))}.$$

Then apply Euler's formula to both sides of the equation:

$$r_N(t)[\cos(\psi_N(t)-\theta_i(t))+i\sin(\psi_N(t)-\theta_i(t))] = \frac{1}{N}\sum_{j=1}^{N}[\cos(\theta_j(t)-\theta_i(t))+i\sin(\theta_j(t)-\theta_i(t))].$$

(b) Equate the real and the imaginary parts:

$$\begin{aligned} r_N(t)\cos(\psi_N(t)-\theta_i(t)) &= \frac{1}{N}\sum_{j=1}^{N}\cos(\theta_j(t)-\theta_i(t)), \\ r_N(t)\sin(\psi_N(t)-\theta_i(t)) &= \frac{1}{N}\sum_{j=1}^{N}\sin(\theta_j(t)-\theta_i(t)). \end{aligned}$$

(c) Substitute the last equation from (b) into

$$d\theta_i(t) = \frac{K}{N}\sum_{j=1}^{N} \sin(\theta_j(t) - \theta_i(t)) + DdW_i(t)$$

to obtain

$$d\theta_i(t) = Kr_N(t)\sin(\psi_N(t) - \theta_i(t)) + DdW_i(t).$$

SOLUTION 9 (Chapter 3, Section 3.4.4).

(a) The derivative of $p(\theta)$ is again $p(\theta)$ multiplied by some term, so we guess

$$p(\theta) = e^{f(\theta)} + C,$$

where C is a constant and $f(\theta)$ a function such that

$$\frac{df(\theta)}{d\theta} = -2Kr\sin\theta.$$

It follows that $f(\theta) = 2Kr\cos\theta$ and

$$p(\theta) = e^{2Kr\cos\theta} + C.$$

(b) Note that $\cos\theta = \cos(\theta + 2\pi)$, so

$$p(0) = e^{2Kr} + C = p(2\pi)$$

for any value of C. We pick $C = 0$. Next, we should have

$$\int_0^{2\pi} e^{2Kr\cos\theta} d\theta = 1.$$

Note that we can multiply $p(\theta)$ with any constant, and it would still satisfy the differential equation and the boundary conditions. Consider

$$p(\theta) = \frac{1}{Z}e^{2Kr\cos\theta},$$

then we find

$$Z = \int_0^{2\pi} e^{2Kr\cos\theta} d\theta,$$

which is a special function called a *modified Bessel function of the first kind*.

Bibliography

[1] Alberich, R., Miro-Julia, J., & Rosselló, F. (2002). *Marvel Universe looks almost like a real social network*. arXiv preprint cond-mat/0202174.

[2] Barabási, A. L., & Bonabeau, E. (2003). *Scale-free networks*. Scientific american, 288(5), 60-69.

[3] Beveridge, A., & Shan, J. (2016). *Network of thrones*. Math Horizons, 23(4), 18-22.

[4] Easley, D., & Kleinberg, J. (2010). *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press

[5] Jansen, B. (2017, January). *Micro-course on Structural Graph Parameters, part 1: Treewidth*. Presented at the 4th Training Week of the Networks Consortium, Doorn, The Netherlands.

[6] Kroon, L., & Schrijver, L. (2015). *Spoornetwerken*. Nieuw Archief voor Wiskunde, 5(3), 165-173.

NET
WORKS

by address: University of Amsterdam
Faculty of Science - Korteweg-de Vries
Institute
PO Box 94248
1090 GE Amsterdam
the Netherlands
+31 (0)20 525 6499
info@thenetworkcenter.nl
www.thenetworkcenter.nl

design www.dezagerij-ontwerp.nl • **lay-out** Vincent Schmeits

WWW.THENETWORKCENTER.NL