Universiteit Utrecht

Faculty of Science

A Centrality Measure of European Air Traffic

Using Markov Chains and the Spacey Random Walk

Bachelor Thesis TWIN

Laura Deen

Physics and Astronomy // Mathematics

Supervisor:

Dr. S.W. Gaaf
Mathematical Institute
June 6, 2019

Abstract

A centrality measure is a real valued function that aims to identify the importance of nodes in a graph. The Markov chain and spacey random walk are both discrete stochastic processes that can be used to define a centrality measure. The stationary distributions of the Markov chain and the spacey random walk come in handy when considering centrality measures. The stationary distributions of these stochastic processes can both be found by considering eigenvector problems. For the first-order Markov chain, this translates into a regular eigenvector problem. For the higher-order Markov chain and the spacey random walk, this translates into a Z-eigenvector problem. These stationary distributions exist and are unique in some cases. We eventually apply these centrality measures on real-world data: we investigate centralities in European air traffic by means of a first-order Markov chain and a spacey random walk derived from a first-order Markov chain.

Contents

Introduction

We all know that we do not want to visit Schiphol Airport in the summer holidays: the long queues for security can make a beach trip to Spain begin with a lot of annoyance. Even on an average day at Schiphol, approximately *every* minute an aircraft departs or arrives. When we consider those numbers, it actually makes sense that queues tend to become annoyingly long.
However, Schiphol is not only great for people who want to go on a holiday. It is very tempting for businesses as well: Schiphol is responsible for 1.6 billion euros of real estate close to the aiport. Schiphol has a net gross revenue of 1.5 billion euros per year. There are 1.72 million tons of cargo that arrive at Schiphol every year [22]. And those quantities only keep increasing per year.

These numbers are dazzling. In a free market economy, however, these numbers are far from enough. In order to keep the numbers from decreasing, we have to expand the area where Schiphol is based. This gave rise to the idea to transfer holiday flights from Schiphol Airport to Lelystad Airport: a newly built airport in the Dutch province of Flevoland. With the introduction of Lelystad Airport, a lot of critique was unleashed as well. Especially the noise disturbance caused by aircrafts flying over the *Veluwe* is a big reason to stop Lelystad Airport from happening.

All the environmental and economic arguments aside, we can say something about the centrality of Schiphol Airport in Europe and how Lelystad will influence this in the future. We can look at all the airports in Europe and consider them as a network or a graph. Then, we can use mathematical tools to investigate the centralities in European air traffic.

Centrality Measures

In this thesis, we will be talking about graphs. A graph $G = (V_N, E)$ consists of a set of N nodes, V_N, and a set of edges $E \subseteq V_N \times V_N$. Edges represent the connections between the nodes. An adjacency matrix is the matrix corresponding to the graph: in such a matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, the nodes are represented by the row and column indices and every entry A_{ij} represents the weight of the edge connecting the nodes i and j. We say a graph is undirected if $A_{ji} = A_{ij}$, or otherwise stated that $\mathbf{A}$ is symmetric. If $\mathbf{A}$ is not symmetric, a graph is called directed. In this thesis, we will only discuss directed graphs.

A centrality measure aims to identify the relevant nodes in a graph. Such a centrality measure has to be a real-valued function that is invariant under relabelling the nodes [24]. What is more, an *eigenvector centrality* is a centrality that uses eigenvectors. A centrality measure could focus on different things that makes the node central. The centrality measure we want to define, focuses on two things: firstly, it depends on the number of connections a node has, which is also called the *degree*, and secondly, it depends on the probability of ending up in a node. Let us explore how we can apply centrality measures on a practical application such as air traffic.

A Centrality Measure of European Air Traffic

An airport that has lots of flights scheduled for other airports, should have a higher centrality than an airport that only has one destination you can go to. Also, the number of passengers that depart from or arrive at an airport gives an insight into how big and how central the airport actually is. With these two conditions, we aim to identify the relevant airports in the air network. A centrality measure gives the means to express the importance of an aiport in a mathematical quantity.

Markov chains have already been used to structure data, for instance to model air traffic, to rank journals and investigate a chain of forwarded and replied emails [21]. Apart from structuring data, we now want to say something about the centralities of the airports in this thesis. This is why we use Markov chains to define centrality, but also go one step further. In 2017, the spacey random walk was introduced by Benson et al. [3]. This turned out to give better results than a Markov chain does, and this is why we are going to investigate the spacey random walk as a centrality measure in this thesis.

Structure of the Thesis

In chapter 1, we start with discussing the theoretical concepts of a stochastic process and how to define it. Then, we fill focus on a specific example of a stochastic process in chapter 2: the first-order Markov chain. As "first-order" may already imply, there also exist so-called higher-order Markov chains. We will discuss higher-order Markov chains in chapter 4, but before we can do so, we have to introduce the concept of tensors in chapter 3. From higher-order Markov chains, it follows naturally to introduce a new stochastic process: in chapter 5, we will discuss vertex-reinforced random walks and in particular, the spacey random walk.
Our aim is then to model European air traffic data with Markov chains and a spacey random walk. In chapter 6, we will look at how we can implement our theoretical knowledge in Matlab. Then, in chapter 7, we will use this implementation on our data and look at the results.

Chapter 1

Stochastic Processes

In this chapter, we give an outline of what a general stochastic process actually is. We give some definitions to formalize our notion, which we will use throughout this thesis. This chapter solely aims on an abstract understanding of a stochastic process. We base all our definitions and theorems in this chapter on Shreve [23].

1.1 Measure Theory

In this section, we define some general notions from measure theory. These definitions are crucial to define a probability space. We need a probability space to have a fundamental, abstract understanding of a stochastic process. A probability space has three ingredients: a sample space, a sigma-algebra and a probability measure. We will see all these definitions in this section.

When we consider a stochastic variable, we need information. What events can occur? Which are most likely? All the possible outcomes of an experiment are given in the sample space, usually denoted by Ω. We quantify the information in a sample space by defining a σ-algebra. We need a σ-algebra and a probability measure to define our probability space. The following definitions explain those terms. We use the definitions given by Shreve [23].

Definition 1.1 (σ-algebra, [23, 1.1.1]) *A family $\mathcal{F}$ of subsets of the sample space Ω is called a σ-algebra if*

i. $\mathcal{F}$ contains the empty set, i.e. $\emptyset \in \mathcal{F}$;

ii. $\mathcal{F}$ is closed under complements, i.e. if $A \in \mathcal{F}$ then $A^C \in \mathcal{F}$;

iii. $\mathcal{F}$ is closed under countable unions, i.e. if $A_1, A_2, ...$ is a family of sets in $\mathcal{F}$, then $\bigcup_i A_i \in \mathcal{F}$.

From these three conditions, we can deduce two other properties of a σ-algebra.
Remark. From i) and ii), it follows that $\emptyset^C = \Omega \in \mathcal{F}$.

Remark. We can check that $\mathcal{F}$ is also closed under intersections, which follows from ii) and iii). If $A_1, A_2, ...$ is a family of sets in $\mathcal{F}$, then

$$\bigcap_i A_i = \left(\bigcup_i A_i^C \right)^C \in \mathcal{F}.$$

The σ-algebra is thus also closed under countable intersections.

A special type of σ-algebra is a Borel σ-algebra. We use the Borel σ-algebra to define the measurability of a random variable (definition 1.5), which makes it important to know what a Borel σ-algebra is.

Definition 1.2 (Borel σ-algebra) *The smallest σ-algebra that contains the interval $[a, b]$ is called the Borel σ-algebra $\mathcal{B}([a, b])$. This is also true for half-open and open intervals.*

The sets in the Borel σ-algebra are called Borel sets. Often, we use the Borel σ-algebra of $\mathbb{R}$, denoted by $\mathcal{B}(\mathbb{R})$. This contains all the Borel subsets of $\mathbb{R}$.

The last ingredient for a probability space is a probability measure. A probability measure is a function that assigns the probability of an event happening to an event.

Definition 1.3 (Probability Measure, [23, 1.1.2]) *A probability measure on a sample space Ω with σ-field $\mathcal{F}$ is a function $P : \mathcal{F} \to [0, 1]$ satisfying*

i. normalization: $P(\Omega) = 1$;

ii. σ-additivity: if $A_1, A_2, ...$ is a countable family of disjoint sets in $\mathcal{F}$, then

$$P\Big(\bigcup_i A_i\Big) = \sum_i P(A_i).$$

With a sample space, sigma-algebra and probability measure, we can define a probability space.

Definition 1.4 (Probability Space) *The triple $(\Omega, \mathcal{F}, P)$ is called a probability space, where Ω denotes the sample space, $\mathcal{F}$ a σ-algebra and P the probability measure on the sample space Ω with σ-algebra $\mathcal{F}$.*

Now, since we have defined a probability space, we can also say something more about stochastic or random variables. Whereas a probability space is very abstract, a random variable translates the abstract into the pragmatical. A random variable has to be measurable by some σ-algebra to make sense of it.

Definition 1.5 (Measurability, [23, 1.2.1]) *Let $(\Omega, \mathcal{F}, P)$ be a probability space. A random variable $X : \Omega \to \mathbb{R}$ is $\mathcal{F}$-measurable if and only if the set $\{X \in B\} = \{\omega \in \Omega : X(\omega) \in B\}$ is in $\mathcal{F}$ for every $B \in \mathcal{B}(\mathbb{R})$.*

Every random variable is defined on a probability space. It is also vital for a random variable to be measurable by a sigma-algebra. However, we often do not denote the probability space on which the random variable is defined. In this thesis, we will also not denote it explicitly –but that does not mean it is not there.

A σ-algebra can also be generated by a random variable or several random variables.

Definition 1.6 (Generated σ-algebra) *Let X be a random variable defined on the non-empty sample space Ω. The σ-algebra $\sigma(X)$, generated by X, is the collection of all subsets of Ω for which $\{\omega \in \Omega : X(\omega) \in B\}$ for every $B \in \mathcal{B}(\mathbb{R})$.*

Also in this case, a σ-algebra always contains a certain amount of information. In the next section, we see how the information that is given in a σ-algebra can affect the eventual probability or stochastic process.

1.2 Information

A σ-algebra gives information about the possible events and outcomes of an experiment. However, sometimes, information increases with time. This is illustrated in the idea of a filtration.

Definition 1.7 (Filtration, [23, 2.1.1]) *Let Ω denote the sample space. Let $N \geq 0$ be fixed and $\mathcal{F}_n$ a σ-algebra for each $n \in [0, N]$. Assume that for every $s \leq n$, every set of $\mathcal{F}_s$ is also in $\mathcal{F}_n$, or $\mathcal{F}_s \subseteq \mathcal{F}_n$. We call the collection of σ-algebras $\mathcal{F}_n$ a filtration for $0 \leq n \leq N$. We also denote the filtration by $\{\mathcal{F}_n : n \geq 0\}$.*

A filtration thus contains a certain amount of information. At different times, the probabilities for the same event can differ due to the change in information. The information in the σ-algebra is relevant when we want to determine the value of a random variable, but it is not sufficient to determine it completely. However, since this information is so important, we still want to use it to make a proper estimate of the random variable. This is why we introduce the concept of conditional expectation.

Definition 1.8 (Conditional Expectation [23, 2.3.1]) *Let $(\Omega, \mathcal{F}, P)$ be a probability space and $X \geq 0$ a random variable. Also, let $\mathcal{G}$ be a sub-σ-algebra of $\mathcal{F}$. The conditional expectation $\mathbb{E}[X|\mathcal{G}]$ of X given $\mathcal{G}$ satisfies*

i. measurability: $\mathbb{E}[X|\mathcal{G}]$ is $\mathcal{G}$-measurable;

ii. partial averaging:

$$\int_A \mathbb{E}[X|\mathcal{G}]dP = \int_A X dP \qquad \forall A \in \mathcal{G}.$$

If X is independent of $\mathcal{G}$, then $\mathbb{E}[X|\mathcal{G}] = \mathbb{E}[X]$. In this case, $\mathbb{E}[X|\mathcal{G}]$ is not a random variable, but simply a number.

Now that we have defined the probability space and we know how to quantify information by means of a filtration, we can turn to stochastic processes.

1.3 Definition of a Stochastic Process

In this section, we will discuss the definition of a stochastic process and of a specific case, the Markov process. Here, we will see a more abstract definition of a stochastic process, but later on, we will use it in practice.

Definition 1.9 (Stochastic Process [20, p. 77]) *A stochastic process $\{X(n), n \in N\}$ is a collection of random variables, that is, for every $n \in N$, $X(n)$ is a random variable.*

The set N could be either a discrete set of numbers or a continuous interval; this determines whether we call the stochastic process a discrete or continuous process. In this thesis, we will only consider discrete processes and usually denote $\{X(n) : n \geq 0\}$ where $n \in \mathbb{N}$. Also, we only consider time-homogeneous processes, which means that the process looks the same independent of the time we are looking at it.
Usually, we are dealing with *adapted* processes. For a given filtration $\{\mathcal{F}_n : n \geq 0\}$, a process is called adapted if every $X(n)$ is $\mathcal{F}_n$-measurable. We only consider adapted processes in this thesis.

We can define a specific kind of stochastic process: a Markov process. This is a widely used stochastic process. In a Markov process, the outcome space is given by the state space. The state space $\langle N \rangle$ is the set which contains all possible states. We say the state space $\langle N \rangle$ consists of N states. Usually, every state is labeled with a number. In practice, we thus have $\langle N \rangle = \{1, 2, \cdots, N\}$.

Definition 1.10 (Markov Process, [23, 2.3.6]) *Let $(\Omega, \mathcal{F}, P)$ be a probability space and $\{\mathcal{F}_n : n \geq 0\}$ a filtration, i.e. a collection of sub-σ-algebras of $\mathcal{F}$. Let $\{X(n) : n \geq 0\}$ be a stochastic process. The process $\{X(n) : n \geq 0\}$ is called a Markov process if for all $0 \leq s \leq n$ and for every non-negative, Borel-measurable function f, there exists another Borel-measurable function g such that*

$$\mathbb{E}[f(X(n))|\mathcal{F}_s] = g(X(s)).$$

An estimate of $f(X(n))$ at time s only depends on $X(s)$ here, not on any values before s. This actually illustrates that a Markov process only depends on a single event in the past. We will actually see that this definition, then, is only sufficient for what we call a first-order Markov process.

In summary, we have seen the most important features of probability theory that we will work with in this thesis. We encountered a formal definition of a Markov process at the end. In the next chapter, we will elaborate on the practical sides of this Markov process.

Chapter 2

First-Order Markov Chains

In this chapter, we discuss a special stochastic process: a first-order Markov chain. In a first-order Markov chain, a transition from one state to another is determined only by the present state you are in. We give the definition of the Markov chain and some examples of a Markov chain. We also see whether we can determine a stationary state of the first-order Markov chain.

2.1 The Definition of the First-Order Markov Chain

In the last chapter, we have seen a formal definition of a Markov process. In this chapter, we are more interested in the practical implications of a Markov process or, what it is called more often, a Markov chain. This is why we will define the first-order Markov chain according to its transition probabilities.

Definition 2.1 (First-Order Markov Chain [20]) *Let $\{X(n) : n \geq 0\}$ be a stochastic process. We say $\{X(n) : n \geq 0\}$ is a first-order Markov chain when the future state depends on the present state only. The probability for a transition from state j to state i is given by*

$$\begin{aligned} P\{X(n+1) = i | X(n) = j,\, X(n-1) = i_{n-1}, ..., X(1) = i_1, X(0) = i_0\} \\ = P\{X(n+1) = i | X(n) = j\} = P_{ij}, \end{aligned} \tag{2.1.1}$$

for all states $i_0, i_1, ..., i_{n-1}, i, j \in \langle N \rangle$, the state space, and all $n \geq 0$.

Here, the transition probabilities are subject to

$$P_{ij} \geq 0, \qquad \sum_{i=0}^{\infty} P_{ij} = 1, \qquad i = 0, 1, ... \, .$$

All the transition probabilities of a Markov chain with N states can be written down in a matrix $\mathbf{P} \in \mathbb{R}^{N \times N}$. Then, it is a matter of convention whether you should choose to use a column-stochastic or row-stochastic matrix. In a column-stochastic matrix, all the elements in every column sum to one. In a row-stochastic matrix, all the elements in every row sum to one. In this thesis, we will use the convention of a column-stochastic matrix throughout.

Example 2.1 Consider the three cities A = Amsterdam, L = London and P = Paris. Let A be the first state, L the second and P the third. Also, consider an aircraft that moves between those cities. The transition matrix for this Markov chain could for instance be given by

$$\mathbf{P} = \begin{bmatrix} 0 & 4/5 & 1/2 \\ 1/3 & 0 & 1/2 \\ 2/3 & 1/5 & 0 \end{bmatrix}. \tag{2.1.2}$$

As we have seen in the introduction, a graph has a corresponding adjacency matrix. If a graph has a corresponding adjacency matrix, an adjacency matrix should also have a corresponding graph. In the case of a Markov chain, we can also picture the transition matrix as a graph. The transition matrix is shown graphically in figure 2.1.

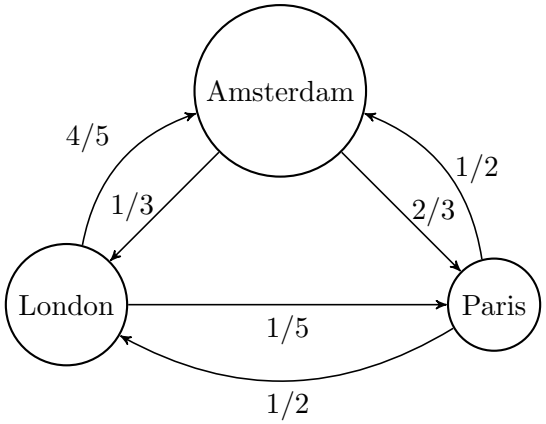


Figure 2.1: The transition probabilities for the Markov chain as given in (2.1.2).

Say we are in an aircraft in Amsterdam. The probability that we will go to London is 1/3, whereas that we will go to Paris has a probability 2/3. We can also say something about the probability that we end up in London from Amsterdam after two transitions, denoted as P^2_{21}. We can compute this by

$$\mathbf{P}^2 = \begin{bmatrix} 0 & 4/5 & 1/2 \\ 1/3 & 0 & 1/2 \\ 2/3 & 1/5 & 0 \end{bmatrix} \begin{bmatrix} 0 & 4/5 & 1/2 \\ 1/3 & 0 & 1/2 \\ 2/3 & 1/5 & 0 \end{bmatrix} = \begin{bmatrix} 3/5 & 1/10 & 2/5 \\ 1/3 & 11/30 & 1/6 \\ 1/15 & 8/15 & 13/30 \end{bmatrix}. \tag{2.1.3}$$

Note that all the columns in $\mathbf{P}^2$ still sum to one and all elements are positive.

We can also express the transition in another way [14]. Consider a first-order Markov chain with N states. Let $\mathbf{x}_n = (x_1, x_2, ..., x_N)^T$ denote the probability distribution at the nth transition. That is, $(\mathbf{x}_n)_j$ is the probability to find the state $X(n) = j$. Also, for every n, we have that every $x_i \geq 0$ and $\sum_{i=1}^{\infty} x_i = 1$. A transition then becomes

$$\mathbf{x}_{n+1} = \mathbf{P}\mathbf{x}_n \qquad \text{or} \qquad \mathbf{x}_{n+1} = \mathbf{P}^{n+1}\mathbf{x}_0. \tag{2.1.4}$$

By considering the transitions in this way, we can make it easier to find a stationary distribution, which we attempt in the next section.

2.2 Stationary Distribution of the First Order Markov Chain

In this section, we are going to introduce some definitions and notations of a first-order Markov chain. We use these definitions to provide the conditions for the existence of a stationary distribution. We will see what a stationary distribution of a first-order Markov chain actually is, how we can compute them and how we can use them as a centrality measure.

At first, we want to say something about the accessibility of a state. We could say that a state is accessible when we can reach the state in the Markov chain. Of course, we can visit every state by starting in this specific state. However, for a state to be accessible, we want to be able to visit it while transitioning.

Definition 2.2 (Accessibility) *A state j is accessible if $P_{ij}^n > 0$ for some $n \in \mathbb{N}_{\geq 1}$, where P_{ij}^n is the nth transition.*

If we can access a state j from a state i and vice versa, we say i and j communicate with each other. We also write $i \leftrightarrow j$. This implies that state i communicates with itself. Say $n = 0$, then $P_{ii}^0 = P\{X_0 = i | X_0 = i\} = 1$.It also follows that state j communicates with state i if state i already communicates with state j. If states in a certain subset of the state space all communicate with each other, we say they belong to the same class.

Example 2.2 Recall the Markov chain with transition matrix given in (2.1.2) and its visualization given in 2.1. From the figure, it becomes very clear that all states communicate. This Markov chain has only one class, i.e. {Amsterdam, London, Paris}. We also call this Markov chain irreducible.

Definition 2.3 (Irreducibility) *A Markov chain is said to be irreducible if all states communicate with each other. So, in order for a Markov chain to be irreducible, $P_{ij}^n > 0$ for all states i and j and for some n.*

If there are irreducible Markov chains, there should also be reducible Markov chains. These are the Markov chains that consist of more than one class.

Example 2.3 Consider the four Scandinavian cities C = Copenhagen, O = Oslo, R = Reykjavik and S = Stockholm. Let C be the first state, O the second, R the third and S the fourth. An aircraft moves between these cities, however, it does not fly to all locations. Sometimes, an aircraft even stays at the airport. The transition matrix for this Markov chain could be given by

$$\mathbf{P} = \begin{bmatrix} 1/2 & 1/2 & 0 & 1/4 \\ 1/2 & 1/2 & 0 & 1/4 \\ 0 & 0 & 1 & 1/4 \\ 0 & 0 & 0 & 1/4 \end{bmatrix}. \tag{2.2.1}$$

The transition matrix is shown graphically in figure 2.2.

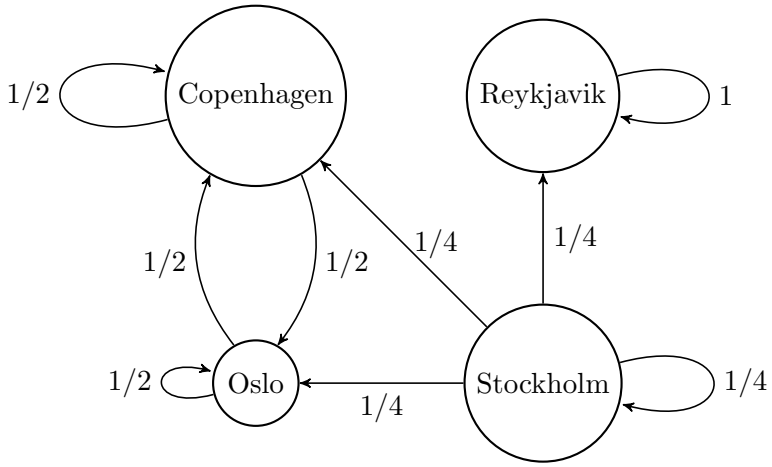


Figure 2.2: The transition probabilities for the Markov chain as given in (2.2.1).

Here, the classes are given by {Copenhagen, Oslo}, {Reykjavik} and {Stockholm}. We can visit Copenhagen from Stockholm, but we cannot visit Stockholm from Copenhagen. The same is true for Oslo and Copenhagen. Since Reykjavik is an absorbing state, i.e. $P_{33} = 1$, no other state is accessible from it.

Reykjavik is called an absorbing state, since we want to distinguish its properties from the other states. This already shows us that we can make another distinction between states. We thus distinguish recurrent and transient states.

Definition 2.4 (Recurrence and Transience) *State i is called recurrent if* $\sum_{n=1}^{\infty} P_{ii}^n = \infty$ *and called transient if* $\sum_{n=1}^{\infty} P_{ii}^n < \infty$.

When starting in a recurrent state j, it is often possible to return to the same state j. We can compute the expected number of transitions it takes to come back in the original state. Let us denote the minimum number of transitions it takes to come to state j with

$$N_j := \min\{n > 0 : X(n) = j\}.$$

We then denote the expected number of transitions it takes to come back in the original state with

$$m_j := E[N_j | X(0) = j].$$

We also distinguish a positive recurrent and a null recurrent state.

Definition 2.5 (Positive and Null Recurrence) *A state is positive recurrent if $m_j < \infty$, whilst a state is null recurrent if $m_j = \infty$.*

Example 2.4 Consider a first-order Markov chain with two states. We distinguish the states that an airplane crashes (state 1) and that it does not (state 0). Suppose we are travelling with very unsafe airplanes, where 1 out of 10 airplanes crashes. The transitions are then given by figure 2.3.

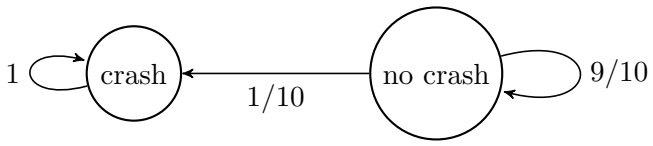


Figure 2.3: The transition probabilities for a Markov chain where the two states are an airplane crash or no crash.

When an airplane crashes, it cannot uncrash. This is why we stay in the crash state once we come in. The transition matrix is given by

$$\mathbf{P} = \begin{bmatrix} 9/10 & 0 \\ 1/10 & 1 \end{bmatrix}. \tag{2.2.2}$$

We can also look at the probabilities after transition n times, which gives us

$$\mathbf{P}^n = \begin{bmatrix} 9^n/10^n & 0 \\ 1 - 9^n/10^n & 1 \end{bmatrix}. \tag{2.2.3}$$

Then, for states 0 and 1, we have

$$\sum_{n=1}^{\infty} P_{00}^n = \sum_{n=1}^{\infty} \frac{9^n}{10^n} = \sum_{n=1}^{\infty} \left(\frac{9}{10}\right)^n = \frac{9/10}{1-9/10} = \frac{9/10}{1/10} = 9 < \infty,$$
$$\sum_{n=1}^{\infty} P_{11}^n = \sum_{n=1}^{\infty} 1 = \infty.$$

This means we call the crash state a recurrent state and the no crash state a transient state. Once we come into the crash state, we will never come out, whereas we can pass through the no crash state and come out of it. This illustrates the difference between a recurrent and a transient state.

We have said that the crash state is an absorbing state. This also means it is a positive recurrent state, since $m_1 = E[N_1|X(0) = 1] = 1 < \infty$, since it only takes one transition to come back in the crash state.
If however, we compute the same for state 1, the no crash state, we should have that $m_0 = E[N_0|X(0) = 0] = \infty$. One out of ten times, the airplane crashes according to our Markov chain. Then, it stays in the crash state. It can never come back in the no crash state once it enters the crash state, which is why $m_j = \infty$. We thus call the no crash state a null recurrent state.

If we do not have an absorbing state such as in the previous example, it becomes a bit more difficult to determine in which state we will be after a long time. Consider the Markov chain from example 2.1 again, of the three cities with the transition matrix given in (2.1.2). We can see that we can come back to any state in a finite number of steps, so every state in (2.1.2) is positive recurrent. Let us have a look at what happens when we look at the probabilities after a lot of transitions:

$$\mathbf{P}^{10} = \begin{bmatrix} 401/931 & 1202/3125 & 834/2029 \\ 136/375 & 771/1942 & 283/750 \\ 981/4748 & 626/2867 & 415/1961 \end{bmatrix}, \quad \mathbf{P}^{25} = \begin{bmatrix} 110/269 & 1949/4762 & 1217/2975 \\ 593/1565 & 766/2023 & 511/1349 \\ 150/707 & 397/1872 & 1655/7802 \end{bmatrix},$$

$$\mathbf{P}^{50} = \begin{bmatrix} 9/22 & 9/22 & 9/22 \\ 25/66 & 25/66 & 25/66 \\ 7/33 & 7/33 & 7/33 \end{bmatrix}, \quad \mathbf{P}^{100} = \begin{bmatrix} 9/22 & 9/22 & 9/22 \\ 25/66 & 25/66 & 25/66 \\ 7/33 & 7/33 & 7/33 \end{bmatrix}.$$

What we can infer from these matrices, is that our system tends to go to a certain value when we perform a transition an infinite number of times. In fact, we can see that $\mathbf{P}^{50} = \mathbf{P}^{100}$. In reality, this means that a system will behave like a steady solution when time goes to infinity. We call these probabilities the long-run proportions or stationary distributions of the Markov chain.

At first, we follow the interpretation of the long-run proportions from Ross [20]. We call x_j the stationary distribution of transitions that come from state j. Then, $x_j P_{ij}$ is the stationary distribution of transitions that come from state j and go to state i.

Theorem 2.6 [20] *If an irreducible Markov chain is transient or null recurrent, $x_i = 0$ for all i. If the Markov chain is positive recurrent, the stationary distributions are given by*

$$x_i = \sum_j x_j P_{ij}, \qquad \sum_i x_i = 1. \tag{2.2.4}$$

Proof. First, note that

$$P\{X_{n+1}=i\}=\sum_{j=0}^{\infty}P\{X_{n+1}=i|X_n=j\}P\{X_n=j\}=\sum_{j=0}^{\infty}P_{ij}P\{X_n=j\}, \tag{2.2.5}$$

where we used conditioning on X_n to come up with the expression. As $n\to\infty$, we have

$$\lim_{n\to\infty}P\{X_{n+1}=i\}=\lim_{n\to\infty}\sum_{j=0}^{\infty}P_{ij}P\{X_n=j\}=\sum_{j=0}^{\infty}P_{ij}\lim_{n\to\infty}P\{X_n=j\}, \tag{2.2.6}$$

assuming we can take the limit inside the summation. Then, note that as $n\to\infty$, we have that $\lim_{n\to\infty}P\{X_{n+1}=i\}=x_i$ and $\lim_{n\to\infty}P\{X_n=j\}=x_j$ precisely, which gives us the same expression as in (2.2.4). Also, note that $\sum_i x_i=\sum_i\lim_{n\to\infty}P\{X_{n+1}=i\}=1$ follows from the fact that all probabilities should sum to 1. □

The calculation given in (2.2.4) resembles calculating an eigenvalue. A more general approach is to regard the stationary distributions as an eigenvalue problem. Then, it is easy to look at the stationary distributions as

$$\lim_{n\to\infty}\mathbf{x}_n=:\bar{\mathbf{x}}=(\bar{x}_1,\bar{x}_2,...,\bar{x}_N)^T=\lim_{n\to\infty}\mathbf{P}\mathbf{x}_{n-1}=\mathbf{P}\bar{\mathbf{x}}.$$

Theorem 2.7 [14] *Let* $\mathbf{P}$ *be a transition probability matrix, then there exists a non-negative vector* $\bar{\mathbf{x}}$ *such that* $\mathbf{P}\bar{\mathbf{x}}=\bar{\mathbf{x}}$*, where* $\sum_{i=1}^{n}\bar{x}_i=1$*. What is more, if* $\mathbf{P}$ *is irreducible, then* $\bar{\mathbf{x}}$ *must be positive and unique.*

In the long run, the Markov chain thus tends to go to its stationary distribution. A stationary distribution is a vector of which the entries say something about the centrality of the state. States with a high centrality will also have an entry with a high value in the stationary distribution vector. The states that are often visited, are more central than states that are not visited that often. Also, states that are connected with many other states, are more easily visited than a state that is not connected to many states. This is why a stationary distribution of a first-order Markov chain can be interpreted as a centrality measure. This is also illustrated by the following example.

Example 2.5 The use of stationary distributions is easily illustrated by our first example about the flights between three cities. Recall the transition probability matrix given in (2.1.2):

$$\mathbf{P}=\begin{bmatrix}0 & 4/5 & 1/2\\ 1/3 & 0 & 1/2\\ 2/3 & 1/5 & 0\end{bmatrix}. \tag{2.2.7}$$

Using theorem 2.6, we get the set of equations

$$\begin{cases}x_0=4/5x_1+1/2x_2\\ x_1=1/3x_0+1/2x_2\\ x_2=2/3x_0+1/5x_1\\ x_0+x_1+x_2=1.\end{cases} \tag{2.2.8}$$

This yields $x_0=9/22$, $x_1=25/66$ and $x=7/33$. From this, we can infer that Amsterdam is more central than London, and London is more central than Paris.

What we have seen in this chapter, is that a first-order Markov chain is only dependent on the present state. As we have seen in example 2.1, we can compute the probability that an aircraft from Amsterdam will end up in London in two transitions. However, the first-order Markov chain gives us the probability that Amsterdam $\rightarrow$ Paris $\rightarrow$ London, Amsterdam $\rightarrow$ London $\rightarrow$ London and Amsterdam $\rightarrow$ Amsterdam $\rightarrow$ London will happen. We can also say something more about a general transition $A \rightarrow B \rightarrow C$, which we call a second-order Markov chain. In order to give a mathematical description of such a Markov chain, we first need to know what tensors are, which we will explore in the next chapter.

Chapter 3

Tensors

In the last chapter about first-order Markov chains, we found out that there exists a stochastic process where the present state depends on the past state. We already briefly discussed the idea of a second-order Markov chain, where a transition depends on the present state and the state before. We need tensors to describe these higher-order stochastic processes. We will discuss higher-order Markov chains in chapter 4. However, in order to describe a higher-order Markov chain, we first need a basic understandig of tensors.
In this chapter, we will elaborate on some tensor properties. We will mainly extend some of the matrix properties we know to higher-order tensor properties.

3.1 Tensor Definition and Unfoldings

In this section, we will explore what a tensor actually is and how we can picture them.

Definition 3.1 (Tensor [8]) *A tensor is a multidimensional array of numbers. We write a tensor of order m as $\underline{\mathbf{A}} \in \mathbb{R}^{N_1 \times N_2 \times \ldots \times N_m}$, sometimes denoted by $\underline{\mathbf{A}}$.*[1] *This means $\underline{\mathbf{A}}$ has $\prod_{i=1}^{m} N_i$ entries.*

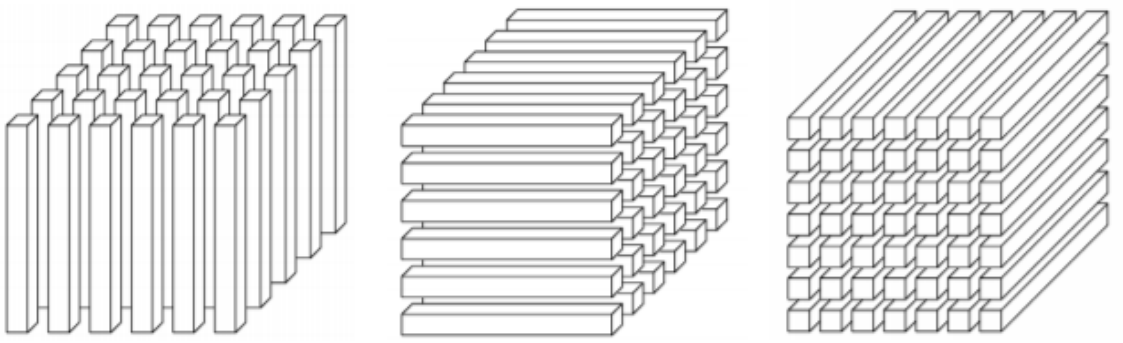
In a more formal way, Kolda et al. describe a tensor of order m as "an element of the tensor product of m vector spaces, each of which has its own coordinate system" [10, p. 455].

In this thesis, it is only necessary to consider cubical tensors. This means that every mode is the same size, or that $N_1 = N_2 = \ldots = N_m = N$ for $\underline{\mathbf{A}} \in \mathbb{R}^{N_1 \times N_2 \times \ldots \times N_m}$.
Practically, an mth order tensor has m dimensions when we picture it. Therefore, it is relatively easy to picture the first few order tensors. A first-order tensor $\underline{\mathbf{A}}_1 \in \mathbb{R}^N$ is a vector; a second-order tensor $\underline{\mathbf{A}}_2 \in \mathbb{R}^{N \times N}$ is a matrix, and in this case a square one - to make it easier to picture. A third-order tensor $\underline{\mathbf{A}}_3 \in \mathbb{R}^{N \times N \times N}$ is a cube, filled with numbers.

Since it is hard to picture a third order tensor, really hard to picture a fourth order tensor and even harder to picture higher-order tensors, it makes life a lot easier to *decompose* the tensor. Kolda and Bader [10] described all the known ways of decomposing a tensor, which we will explain here.
In essence, if we want to decompose a vector, we say we pick an element of the vector. If we take a vector $\mathbf{a}$, we say its ith element is a_i. Similarly, we can also pick an element out of a matrix $\mathbf{A}$, which becomes A_{ij}. Then, an element from a third order tensor $\underline{\mathbf{A}}$ will be $\underline{A}_{ijk}$, and so on. An

[1] Physicists also make use of tensors a lot. These tensors, however, are often not the *mathematical* tensors, but what a mathematician would call a tensor field. In this thesis, we solely use the mathematical definition of a tensor as a multidimensional array of numbers.

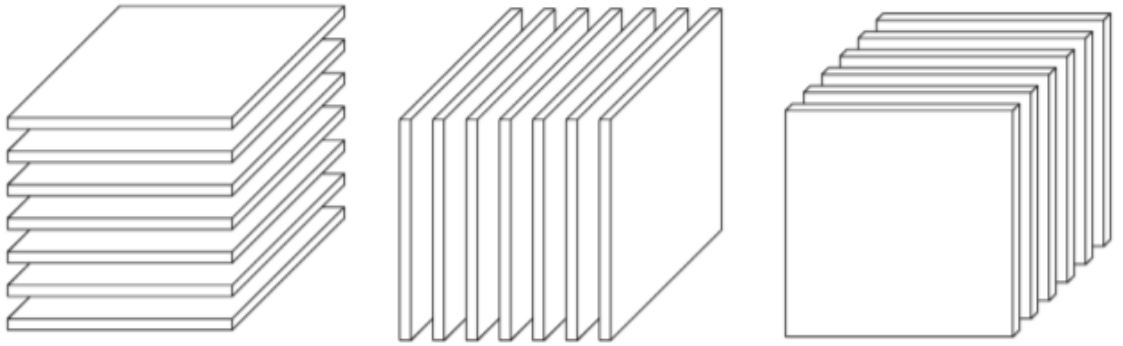
element from an mth order tensor $\underline{\mathbf{A}}_m$ will thus be $\underline{A}_{i_1,i_2,\ldots,i_m}$.
In two dimensions, we can pick different rows and columns out of a matrix. Consider a matrix $\mathbf{A}$. We denote the ith row of the matrix $\mathbf{A}$ with $A_{i\times}$ and the jth column with $A_{\times j}$.
For tensors, we do not use rows and columns anymore, but fibers and slices. A tensor $\underline{\mathbf{A}}$ has row, column and tube fibers, which arise when we fix every index but one. A row, column and tube fiber of a third-order tensor will be, respectively, $\underline{A}_{i\times k}$, $\underline{A}_{\times jk}$ and $\underline{A}_{ij\times}$. This comes down to a one-dimensional subtensor.



(a) $\underline{A}_{i\times k}$ (b) $\underline{A}_{\times jk}$ (c) $\underline{A}_{ij\times}$

Figure 3.1: Row, column and tube fibers of a three-dimensional tensor $\underline{\mathbf{A}}$. Image taken from Kolda et al. [10].

Also, a tensor has horizontal, lateral and frontal slices, which arise from fixing every index but two. These slices will be $\underline{A}_{i\times\times}$, $\underline{A}_{\times j\times}$ and $\underline{A}_{\times\times k}$ in a third-order tensor. These come down to two-dimensional subtensors.



(a) $\underline{A}_{i\times\times}$ (b) $\underline{A}_{\times j\times}$ (c) $\underline{A}_{\times\times k}$

Figure 3.2: Horizontal, lateral and frontal slices of a three-dimensional tensor $\underline{\mathbf{A}}$. Image taken from Kolda et al. [10].

Of course, other, higher-dimensional subtensors also exist. You can imagine, when considering a fourth-order tensor, you can have several subtensors that have order three. Subtensors are not always easy to picture anymore when we go to orders higher than or equal to four, but we can still use them. When going to mth order tensors, we can obtain subtensors up to $(m-1)$th order. These subtensors have no special name.

3.2 Eigenvalues and Eigenvectors

When we speak of eigenvalues and eigenvectors of matrices, i.e. second-order tensors, we consider a matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ and speak of

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x},$$

where $\mathbf{x} \in \mathbb{R}^N$ is an eigenvector of $\mathbf{A}$ and $\lambda \in \mathbb{R}$ an eigenvalue. We only consider real eigenvalues, since complex eigenvalues have no meaning in probability theory.

We now want to extend this notion to higher-order tensors. At first, it may seem difficult to multiply a tensor with a vector: how do we define eigenvectors when we have m different possibilities of multiplying a tensor with a vector or matrix? This is why Chang et al. [4] defined an N-vector which resembles tensor multiplication. Consider the mth-order square tensor $\underline{\mathbf{A}} \in \mathbb{R}^{N\times N\times\cdots\times N}$ and the vector $\mathbf{x} = (x_1, ..., x_N)$, then we define

$$\underline{\mathbf{A}}\mathbf{x}^{m-1} := \left(\sum_{i_2=1}^{N} \sum_{i_3=1}^{N} \cdots \sum_{i_m=1}^{N} \underline{A}_{ii_2\cdots i_m} x_{i_2} \cdots x_{i_m} \right)_{1\le i\le N}. \tag{3.2.1}$$

So we see that $\underline{\mathbf{A}}\mathbf{x}^{m-1} \in \mathbb{R}^N$. With this multiplication, we can define the eigenspace for tensors. For tensors, we can distinguish two kinds of eigenvectors. We base our definitions on those given by Li et al. [14].

Definition 3.2 (H-Eigenpairs) *Let $\underline{\mathbf{A}} \in \mathbb{R}^{N\times N\times\cdots\times N}$ be a tensor of order m and dimension N. Also, let $\underline{\mathbf{A}}\mathbf{x}^{m-1} \neq 0$. If a vector $\mathbf{x} \in \mathbb{R}^N\backslash\{0\}$ satisfies*

$$\underline{\mathbf{A}}\mathbf{x}^{m-1} = \lambda\mathbf{x}^{[m-1]}, \tag{3.2.2}$$

where $\mathbf{x}^{[m-1]} = (x_1^{m-1}, ..., x_n^{m-1})$. Then, we say $(\lambda, \mathbf{x}) \in \mathbb{R} \times (\mathbb{R}^N\backslash\{0\})$ is an H-eigenpair of $\underline{\mathbf{A}}$.

Definition 3.3 (Z-Eigenpairs) *Let $\underline{\mathbf{A}} \in \mathbb{R}^{N\times N\times\cdots\times N}$ be a tensor of order m and dimension N. Also, let $\underline{\mathbf{A}}\mathbf{x}^{m-1} \neq 0$. If a vector $\mathbf{x} \in \mathbb{R}^N\backslash\{0\}$ satisfies*

$$\underline{\mathbf{A}}\mathbf{x}^{m-1} = \lambda\mathbf{x}, \tag{3.2.3}$$

then we say $(\lambda, \mathbf{x}) \in \mathbb{R} \times (\mathbb{R}^N\backslash\{0\})$ is a Z-eigenpair of $\underline{\mathbf{A}}$.

Then, for supersymmetric tensors, we can even say something about the number of H- and Z-eigenvalues and -eigenvectors. A supersymmetric tensor is defined according to definition

Definition 3.4 (Supersymmetry [10]) *A tensor $\underline{\mathbf{A}} \in \mathbb{R}^{N\times N\times\ldots\times N}$ of order m and dimensions N is called supersymmetric if its elements remain the same under any permutation.*

Let us illustrate this for a third order tensor. The tensor $\underline{\mathbf{A}}_3 \in \mathbb{R}^{N\times N\times N}$ is supersymmetric if

$$x_{ijk} = x_{ikj} = x_{jik} = x_{jki} = x_{kij} = x_{kji} \quad \forall i, j, k = 1, ..., N. \tag{3.2.4}$$

For matrices, i.e. tensors of second order, we can see that an eigenpair is both an H-eigenpair and a Z-eigenvalue. For both H- and Z-eigenvalues and -eigenvectors, Qi has proven that a supersymmetric tensor of order m, with m even, and dimension N has $N(m-1)^{N-1}$ eigenvalues and eigenvectors [19]. What is more, of thes $N(m-1)^{N-1}$ eigenvalues and eigenvectors, strictly less than $N(m-1)^{N-1}$ is real for $m \geq 4$.

3.3 Perron-Frobenius Theorem

The Perron-Frobenius theorem is a very pragmatical result for non-negative matrices, such as stochastic matrices. This theorem tells us something about the eigenpairs of stochastic matrices. A matrix often has several eigenvalues. We call this set the spectrum of a matrix.

Definition 3.5 (Spectrum) *Let* $\mathbf{A} \in \mathbb{R}^{N\times N}$. *We say* $\sigma(A) = \{\lambda_1, \lambda_2, ..., \lambda_N\}$, *where* λ_i *is an eigenvalue of* $\mathbf{A}$, *is the spectrum of* $\mathbf{A}$.

Definition 3.6 (Spectral Radius) *Let* $\mathbf{A} \in \mathbb{R}^{N\times N}$. *The spectral radius of* $\mathbf{A}$ *is defined by*

$$\rho(A) = \max\{|\lambda| \mid \lambda \in \sigma(A)\}, \tag{3.3.1}$$

where $\sigma(A)$ *is the spectrum of* $\mathbf{A}$ *as defined in definition 3.5.*

For a matrix to be irreducible, it is equivalent to say that the Markov chain is irreducible.

Theorem 3.7 (Perron-Frobenius Theorem for Matrices) *Let* $\mathbf{A} \in \mathbb{R}^{N\times N}$ *be irreducible and non-negative. Then,*

i. the spectral radius $\rho(A)$ *is an eigenvalue of* $\mathbf{A}$;

ii. there exists a non-negative vector $\mathbf{x} \in \mathbb{R}^N$, *such that* $\mathbf{Ax} = \rho(A)\mathbf{x}$;

iii. if λ *is an eigenvalue with a non-negative eigenvector, then* $\lambda = \rho(A)$;

iv. the eigenvalue $\lambda = \rho(A)$ *is a simple eigenvalue* $\mathbf{A}$;

v. if μ *is an eigenvalue of* $\mathbf{A}$, *then* $|\mu| \leq \lambda = \rho(A)$.

Recall that we call an eigenvector simple if its algebraic multiplicity is equal to one. In practice, this comes down to the eigenvalue λ only having one eigenvector, a unique eigenvector. The Perron-Frobenius theorem actually assigns many properties to an object with very few conditions. Since a stochastic matrix is also a positive matrix, these results are also applicable to our transition matrix. Then, the following lemma also comes in handy.

Lemma 3.8 [9] *If* $\mathbf{A}$ *is positive, i.e. all its elements are greater than zero, then* $\mathbf{A}$ *has a positive eigenpair* $(\lambda = \rho(A), \mathbf{x})$, *where* $\mathbf{x}$ *is unique up to a multiplicative constant.*

The Perron-Frobenius theorem is a very useful theorem for stochastic matrices. You can show that a stochastic matrix always has to have an eigenvalue $\lambda = 1$. Then, Perron-Frobenius tells us that the eigenvalue problem $\mathbf{Px} = \mathbf{x}$ has a solution for $\mathbf{x}$ where we can always normalize $\mathbf{x}$ to a stochastic vector. Translating to Markov language, this means that the first-order Markov chain has a unique stationary distribution indeed, which was already given in 2.7.

It turns out we can extend this result to higher-order tensors and generalise the theorem. Chang et al. [4] proved that the Perron-Frobenius theorem is also applicable to H-eigenvalues for tensors. We see that the Perron-Frobenius theorem places two conditions on the matrix: it has to be non-negative and irreducible. Then, we also expect the conditions for the tensor to be non-negativity and irreducibility. First, we need to define these features for a tensor.

Definition 3.9 (Non-negativity) *A tensor* $\underline{\mathbf{A}}$ *is called non-negative if all entries of the tensor are greater than or equal to zero. What is more, it is called positive if the tensor is non-negative and all its elements are non-zero.*

Definition 3.10 (Reducibility of a Tensor [4]) *A square tensor* $\underline{\mathbf{A}} \in \mathbb{R}^{N\times N\times\ldots\times N}$ *of order* m *and dimension* n *is called reducible if there exists a proper, non-empty index subset* $I \subset \{1,\ldots,N\}$ *such that*

$$a_{i_1,i_2,\ldots,i_m} = 0 \qquad \forall i_1 \in I \qquad \forall i_2,\ldots,i_m \notin I. \tag{3.3.2}$$

If $\underline{\mathbf{A}}$ *is not reducible, then* $\underline{\mathbf{A}}$ *is said to be irreducible.*

When a tensor is non-negative and irreducible, we can also say something about the properties of the H-eigenpairs.

Theorem 3.11 (Perron-Frobenius for Tensors with H-eigenpairs [4]) *Let* $\underline{\mathbf{A}} \in \mathbb{R}^{N\times N\times\ldots\times N}$ *be a tensor of order* m *and dimension* N*, which is irreducible (definition 3.10) and non-negative. Then, for the pair* $(\lambda, \bar{\mathbf{x}}) \in (\mathbb{R}, \mathbb{R}^N\backslash\{0\})$*,*

i. the tensor $\underline{\mathbf{A}}$ *has an* H*-eigenvalue* $\lambda > 0$*;*

ii. the tensor $\underline{\mathbf{A}}$ *has an* H*-eigenvector* $\bar{\mathbf{x}}$ *with every* $x_i > 0$*;*

iii. if $\bar{\mathbf{x}}$ *is an* H*-eigenvector with* $x_i > 0$*, then its* H*-eigenvalue is* λ*. This eigenvector is unique up to a multiplicative constant;*

iv. if μ *is an eigenvector of* $\underline{\mathbf{A}}$*, then* $|\mu| \leq \lambda$*.*

For a proof, see Chang et al. [4].

From Perron-Frobenius theory, we know two important things: stochastic matrices always have simple (or unique) eigenvectors and stochastic tensors have H-eigenvectors. Specifically, such an eigenvector is called a Perron vector. In other words, the Perron vector is the eigenvector corresponding to the highest algebraic eigenvalue. Unfortunately, we do not have such a nice theorem for Z-eigenpairs.

In this chapter, we have seen what a tensor is and how we can visualize it. We have seen that eigenpairs are also defined for tensors and we have discussed the Perron-Frobenius theorem, a very useful result for positive matrices and tensors. In the next chapter, we will see more of the applications of tensors and the theorems we encountered in this chapter.

Chapter 4

Higher-Order Markov Chains

In this chapter, we discuss higher-order Markov chains. A higher-order Markov chain depends on the present state and several past states. In this sense, it is actually a generalization of the Markov chain. We also explore the stationary distributions of the higher-order Markov chain.

4.1 The Definition of the Higher-Order Markov Chain

A second-order Markov chain is a stochastic process where the future state not only depends on the present state, but also on the past state just before the present state. We can define such a second-order Markov chain by looking at its transition probabilites.

Definition 4.1 (Second-Order Markov Chain) *Let $\{X(n) : n \geq 0\}$ be a stochastic process. We say $\{X(n) : n \geq 0\}$ is a second-order Markov chain when the future state not only depends on the present state, but also on the state before. The probability for a transition from state k to state j to state i then becomes:*

$$\begin{aligned} P\{X(n+1) = i | X(n) = j, X(n-1) = k, ..., X(1) = i_1, X(0) = i_0\} \\ = P\{X(n+1) = i | X(n) = j, X(n-1) = k\} = \underline{P}_{ijk}. \end{aligned} \tag{4.1.1}$$

We still think of $\mathbf{x}_n \in \mathbb{R}^N$ as the probability distribution at the nth transition. Also, $\mathbf{x}_{n,n-1}$ is the probability distribution for the $n-1$th to nth transition. However, we do not quite know what this looks like. This is just notation, it may be very hard to compute those probability distributions. If you want to make a transition, we denote

$$\mathbf{x}_{n+1} = \underline{\mathbf{P}}\mathbf{x}_{n,n-1}. \tag{4.1.2}$$

Since a second-order Markov chain is a stochastic process, we have

$$\underline{P}_{ijk} \geq 0, \qquad \sum_{i=0}^{\infty} \underline{P}_{ijk} = 1.$$

EXAMPLE 4.1 Consider the three cities R = Rome, M = Madrid and V = Vienna. Let R be the first state, M the second and V the third. An aircraft moves between those cities, but always makes two transitions at the time. For instance, it first flies to Rome, starting in Madrid, then it flies back to Rome.

The unfolding of the transition tensor of this second-order Markov chain can for instance be given by

$$\underline{\mathbf{P}}_{(1)} = \left[\underline{\mathbf{P}}_{ij0} \,|\, \underline{\mathbf{P}}_{ij1} \,|\, \underline{\mathbf{P}}_{ij2}\right] = \left[\begin{array}{ccc|ccc|ccc} 2/5 & 0 & 0 & 1 & 1/3 & 0 & 1 & 1/5 & 1/4 \\ 2/5 & 1 & 1/2 & 0 & 1/3 & 0 & 0 & 0 & 1/2 \\ 1/5 & 0 & 1/2 & 0 & 1/3 & 1 & 0 & 4/5 & 1/4 \end{array}\right]. \tag{4.1.3}$$

Now, we see that we have 27 entries instead of 9. Every entry is the probability that we go from the pair of states (k, j) to the pair (j, i). This is illustrated more intuitively in table 4.1.

Second last state	R			M			V		
Last state	R	M	V	R	M	V	R	M	V
Prob. next state is R	2/5	0	0	1	1/3	0	1	1/5	1/4
Prob. next state is M	2/5	1	1/2	0	1/3	0	0	0	1/2
Prob. next state is V	1/5	0	1/2	0	1/3	1	0	4/5	1/4

Table 4.1: The transition probabilities for the second-order Markov chain.

For instance, given that we are in Madrid and came from Vienna, the pair of states (V, M), the probability is 1/5 that we will be in Rome, i.e. the pair of states (M, R), at the subsequent time. This transition is shown in red in figure 4.1.

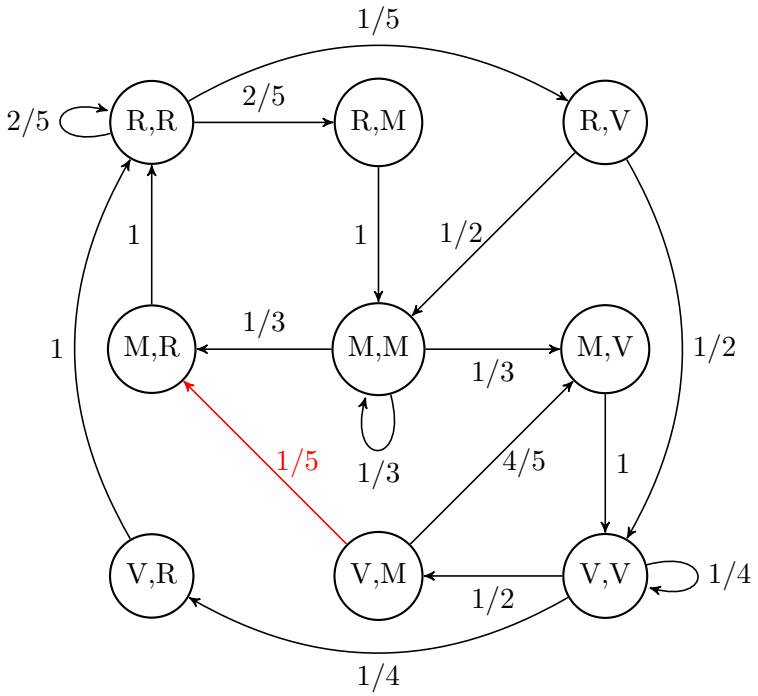


Figure 4.1: Visualization of the second-order Markov chain with transition probabilities given by (4.1.3).

With the second-order Markov chain, we can thus look at transitions from $A \to B \to C$, whereas we can only look at the transition $A \to C$ when considering a first-order Markov chain.

We have just said that a Markov chain can depend on the two former states: now, to make it more general, we can also say that it can depend on more than two states.

Definition 4.2 (Higher-Order Markov Chain) *Let $\{X(n) : n \geq 0\}$ be a stochastic process. We say $\{X(n) : n \geq 0\}$ is an $(m-1)$th-order Markov chain when the future state not only depends on the present state, but also on the past states. The probability for a transition from state $i_m, ..., i_2$ to state i_1 then becomes:*

$$P\{X(n+1) = i_1 | X(n) = i_2,\, X(n-1) = i_3, ..., X(0) = i_m\} = \underline{P}_{i_1 i_2 \cdots i_m}. \tag{4.1.4}$$

Now, let $\underline{\mathbf{P}} \in \mathbb{R}^{N \times N \times \ldots \times N}$ be an $(m-1)$th-order N-dimensional tensor. This means the tensor $\underline{\mathbf{P}}$ has N^{m-1} entries. We can also think of $\mathbf{x}_n \in \mathbb{R}^N$ as the probability distribution at the nth transition, equivalent to the case for the second-order Markov chain. Still, this is just notation. For the $(m-1)$th-order Markov Chain, this means we can express a transition as

$$\mathbf{x}_{n+1} = \underline{\mathbf{P}} \mathbf{x}_{n,n-1,\ldots,n-m+2}. \tag{4.1.5}$$

Since we are still talking about probabilities, the tensor should satisfy

$$\underline{P}_{i_1,i_2,\ldots,i_m} \geq 0, \qquad \sum_{i_1=1}^{\infty} \underline{P}_{i_1,i_2,\ldots,i_m} = 1, \tag{4.1.6}$$

for every $i_1, i_2, ..., i_m \in \langle N \rangle$, the state space. Since all the elements in a column sum to one, we call this transition tensor a column-stochastic tensor.

4.2 The Stationary Distribution of a Higher-Order Markov Chain

As we have seen, the first-order Markov chain has stationary distributions: distributions to which the chain converges when we let the number of transitions go to infinity. If the first-order Markov chain has stationary distributions, a higher-order Markov chain might have them as well. In this section, we will discuss whether these stationary distributions exist and what they look like.
The stationary distributions of a first-order Markov chain are the distributions that arise when we let the number of transitions go to infinity. In other words, when we translate this to an $(m-1)$th-order Markov chain, we are interested in

$$\bar{\mathbf{x}} = \lim_{n\to\infty} \mathbf{x}_n = \underline{\mathbf{P}} \lim_{n\to\infty} \mathbf{x}_{n-1,n-2,\ldots,n-m+1} \text{ with } \bar{x}_i = \lim_{n\to\infty} P\{X(n) = i\}. \tag{4.2.1}$$

The first-order Markov chain has a transition matrix with a stationary distribution vector. Since a second-order Markov chain has a transition tensor, we assume it to have a matrix as limiting distribution, where all the entries represent the stationary probabilities of pairs of states (i, j) [3]. We can compute the matrix by

$$X_{ij} = \sum_{k=0}^{\infty} \underline{P}_{ijk} X_{jk}, \qquad \sum_{i,j=0}^{\infty} X_{ij} = 1, \qquad X_{ij} \geq 0. \tag{4.2.2}$$

However, for big matrices, this method demands loads of storage. Imagine that a fourth-order tensor will have a third-order tensor as a stationary distribution, a fifth-order tensor will have a fourth-order tensor and so on. Li & Ng showed that another method may be more efficient, while

still giving sufficient values [14]. We assume that the joint probability distribution is the Kronecker product of its limiting probability distribution.

$$\lim_{n\to\infty} \mathbf{x}_{n-1,n-2,\ldots,n-m+1} = \lim_{n\to\infty} \mathbf{x}_{n-1} \otimes \lim_{n\to\infty} \mathbf{x}_{n-2} \otimes \ldots \otimes \lim_{n\to\infty} \mathbf{x}_{n-m+1} \tag{4.2.3}$$

Recall that the Kronecker product $(\mathbf{A} \otimes \mathbf{B}) \in \mathbb{R}^{(MP)\times(NQ)}$ of two matrices $\mathbf{A} \in \mathbb{R}^{M\times N}$, $\mathbf{B} \in \mathbb{R}^{P\times Q}$ is given by

$$\mathbf{A}\otimes\mathbf{B} = \begin{bmatrix} A_{11}B_{11} & A_{11}B_{12} & \cdots & A_{11}B_{1Q} & \cdots & \cdots & A_{1N}B_{11} & A_{1N}B_{12} & \cdots & A_{1N}B_{1Q} \\ A_{11}B_{21} & A_{11}B_{22} & \cdots & A_{11}B_{2Q} & \cdots & \cdots & A_{1N}B_{21} & A_{1N}B_{22} & \cdots & A_{1N}B_{2Q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ A_{11}B_{P1} & A_{11}B_{P2} & \cdots & A_{11}B_{PQ} & \cdots & \cdots & A_{1N}B_{P1} & A_{1N}B_{P2} & \cdots & A_{1N}B_{PQ} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ A_{M1}B_{11} & A_{M1}B_{12} & \cdots & A_{M1}B_{1Q} & \cdots & \cdots & A_{MN}B_{11} & A_{MN}B_{12} & \cdots & A_{MN}B_{1Q} \\ A_{M2}B_{21} & A_{M1}B_{22} & \cdots & A_{M1}B_{2Q} & \cdots & \cdots & A_{MN}B_{21} & A_{MN}B_{22} & \cdots & A_{MN}B_{2Q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ A_{M1}B_{P1} & A_{M1}B_{P2} & \cdots & A_{M1}B_{PQ} & \cdots & \cdots & A_{MN}B_{P1} & A_{MN}B_{P2} & \cdots & A_{MN}B_{PQ} \end{bmatrix}.$$

Heuristically seen, we observe that in the case of an $(m-1)$th order Markov chain, we would have an mth order tensor, thus an $(m-1)$th order tensor that gives its stationary distribution. If the mth order tensor is N-dimensional, the $(m-1)$th order tensor will also be N-dimensional. This tensor has N^{m-1} entries. When we look at the Kronecker product, a Kronecker product of two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^N$ will be $\mathbf{a} \otimes \mathbf{b} \in \mathbb{R}^{N\times N}$. The Kronecker product in (4.2.3) thus is a vector in $\mathbb{R}^{N\times N\times\ldots\times N}$ space. This means that the resulting vector in (4.2.3) also has N^m entries.

The use of the Kronecker product is, for instance, also illustrated in the higher-order singular value decomposition (HO-SVD) developed by De Lathauwer et al. [13]. They built this HO-SVD from the idea of a matrix SVD. They also remark there exists a relation between tensors and tensor space:

> In the same way as there exists an isomorphic link between matrix algebra and the algebra of linear vector mappings, a higher-order tensor can be regarded as formal representation of a linear mapping between a matrix and a vector space, a matrix and a matrix space, a matrix and a higher-order tensor space, etc. [13, p. 1274]

Since a tensor SVD can also be expressed by means of a Kronecker product, we could maybe think that a Kronecker product is the key to extending matrix computations to higher dimensions. This motivated Li & Ng to assume the joint probability distribution could be written as a Kronecker product.

Assuming that the joint probability distribution can be written as a Kronecker product, we can

compute the stationary distribution by

$$\begin{aligned}
\bar{x}_{i_1} &= \lim_{n\to\infty} P\{X(n) = i_1\} \\
&= \lim_{n\to\infty} \sum_{i_2=1}^{N}\sum_{i_3=1}^{N}\cdots\sum_{i_m=1}^{N} \underline{P}_{i_1,i_2,\ldots,i_m} P\{X(n-1)=i_2, X(n-2)=i_3, \ldots, X(n-m+1)=i_m\} \\
&= \sum_{i_2=1}^{N}\sum_{i_3=1}^{N}\cdots\sum_{i_m=1}^{N} \underline{P}_{i_1,i_2,\ldots,i_m} \prod_{k=2}^{m} \lim_{n\to\infty} P\{X(n)=i_k\} \\
&= \sum_{i_2=1}^{N}\sum_{i_3=1}^{N}\cdots\sum_{i_m=1}^{N} \underline{P}_{i_1,i_2\ldots,i_m} \bar{x}_{i_2}\ldots\bar{x}_{i_m} \\
&= (\underline{\mathbf{P}}\bar{\mathbf{x}}^{m-1})_{i_1}
\end{aligned}$$

If, then, we want to determine the stationary distribution of a higher-order Markov chain, we should solve the following tensor equation:

$$\mathbf{x} = \underline{\mathbf{P}}\mathbf{x}^{m-1}. \tag{4.2.4}$$

Now, the problem of solving for a stationary distribution has turned into a Z-eigenvalue problem. In the next section, we will see more clearly why this stationary distribution always exists. We will also see that this stationary distribution is unique under certain conditions.

4.3 Existence and Uniqueness of the Stationary Distribution

In this section, we primarily look at the existence of the stationary distribution of a higher-order Markov chain. We also briefly see under which conditions this stationary distribution is unique.

In the previous section, we have seen that we can regard the problem of finding the stationary distribution of a higher-order Markov chain as a Z-eigenvalue problem. In the following propositions and theorems, we will see why a solution to this Z-eigenvalue problem exists. First, I have proved the propositions needed for the understanding of theorem 4.6.

Proposition 4.3 *Let $\underline{\mathbf{P}}$ be a non-negative tensor of order m and dimension N that satisfies the conditions in (4.1.6), and let $\mathbf{x} \in \mathbb{R}^N$. When $x_i \geq 0$ and $\sum_i x_i = 1$, then $(\underline{\mathbf{P}}\mathbf{x}^{m-1})_i$ must be non-negative and $\sum_i (\underline{\mathbf{P}}\mathbf{x}^{m-1})_i = 1$.*

Proof. For $(\underline{\mathbf{P}}\mathbf{x}^{m-1})_i$, we have

$$(\underline{\mathbf{P}}\mathbf{x}^{m-1})_i = \sum_{i_2=1}^{N}\sum_{i_3=1}^{N}\cdots\sum_{i_m=1}^{N} \underline{P}_{ii_2\cdots i_m} x_{i_2} x_{i_3} \cdots x_{i_m}.$$

From (4.1.6), we know that every $\underline{P}_{ii_2\cdots i_m} \geq 0$. We also know that every $x_i \geq 0$, thus that $x_{i_2}\cdots x_{i_m} \geq 0$. Since both terms are greater than or equal to zero, we conclude that $p_{ii_2\cdots i_m} x_{i_2}\cdots x_{i_m} \geq 0$. Summing over all elements, we still have that $(\underline{\mathbf{P}}\mathbf{x}^{m-1})_i \geq 0$.

Summing over all i, we get

$$\begin{aligned}\sum_i (\underline{\mathbf{P}}\mathbf{x}^{m-1})_i &= \sum_i \left(\sum_{i_2=1}^{N} \sum_{i_3=1}^{N} \cdots \sum_{i_m=1}^{N} \underline{P}_{ii_2\cdots i_m} x_{i_2} x_{i_3} \cdots x_{i_m} \right) \\ &= \sum_{i_2=1}^{N} \sum_{i_3=1}^{N} \cdots \sum_{i_m=1}^{N} \left(\sum_i \underline{P}_{ii_2\cdots i_m} \right) x_{i_2} x_{i_3} \cdots x_{i_m} \\ &= \sum_{i_2=1}^{N} x_{i_2} \sum_{i_3=1}^{N} x_{i_3} \cdots \sum_{i_m=1}^{N} x_{i_m} = 1 \cdot 1 \cdot \ldots \cdot 1 = 1.\end{aligned}$$

We conclude that $(\underline{\mathbf{P}}\mathbf{x}^{m-1})_i$ is non-negative and that $\sum_i (\underline{\mathbf{P}}\mathbf{x}^{m-1})_i = 1$. □

Proposition 4.4 *Let*

$$\Delta_N = \{\mathbf{x} = (x_1, x_2, .., x_N)^T \in \mathbb{R}^n | x_i \geq 0, 1 \leq i \leq N, \sum_{i=1}^{N} x_i = 1\}.$$

Then Δ_N is a compact convex set. We also call Δ_N the unit or probability N-simplex.

PROOF. For compactness, we should only prove that Δ_N is a closed and bounded set, since $\Delta_N \subset \mathbb{R}^n$. For boundedness, we have $\sum_{i=1}^{N} x_i = 1$ and $x_i \geq 0$. This means that $x_i \leq 1$ for every $0 \leq 1 \leq N$, thus every $\mathbf{x} \in \Delta_N$ is bounded, therefore Δ_N is bounded. For closedness, define $f : \mathbb{R}_+^N \to \mathbb{R}, \mathbf{x} \mapsto \sum_{i=1}^{N} x_i$, where we note that f is continuous. Then, we note that $f^{-1}(\{1\}) = \Delta_N$ is closed since $\{1\}$ is closed.
For convexity, we have to show that $\lambda\mathbf{x} + (1-\lambda)\mathbf{y} \in \Delta_N$ for every $\lambda \in [0,1]$ and $\mathbf{x}, \mathbf{y} \in \Delta_N$. For $\lambda\mathbf{x} + (1-\lambda)\mathbf{y}$, we get

$$\begin{aligned}\lambda\mathbf{x} + (1-\lambda)\mathbf{y} &= \big(\lambda x_1, \lambda x_2, ..., \lambda x_N\big)^T + \big((1-\lambda)y_1, (1-\lambda)y_2, .., (1-\lambda)y_N\big)^T \\ &= \big(\lambda x_1 + (1-\lambda)y_1, \lambda x_2 + (1-\lambda)y_2, ..., \lambda x_N + (1-\lambda)y_N\big)^T.\end{aligned}$$

Since $\lambda \in [0,1]$ and $x_i, y_i \geq 0$, we have $\lambda x_i + (1-\lambda)y_i \geq 0$. What is more, we also know that $\sum_{i=1}^{N} x_i = 1$ and $\sum_{i=1}^{N} y_i = 1$, so

$$\sum_{i=1}^{N} \lambda x_i + (1-\lambda)y_1 = \lambda \sum_{i=1}^{N} x_i + (1-\lambda) \sum_{i=1}^{N} y_i = \lambda + (1-\lambda) = 1.$$

This means that $\lambda\mathbf{x} + (1-\lambda)\mathbf{y} \in \Delta_N$. Thus, Δ_N is a convex set.
With compactness and convexity combined, we conclude that Δ_N is a compact convex set. □

Proposition 4.5 *Let $\underline{\mathbf{P}}$ be a non-negative tensor of order m and dimension n that satisfies the conditions in* (4.1.6)*. We define the map $\Phi : \Delta_N \to \Delta_N$,*

$$(\Phi(\mathbf{x}))_i = (\underline{\mathbf{P}}\mathbf{x}^{m-1})_i. \tag{4.3.1}$$

Then Φ is well-defined and continuous.

PROOF. We consider $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$. Now, we want to prove that $\Phi(\mathbf{x}) = \Phi(\mathbf{y})$ if $\mathbf{x} = \mathbf{y}$. Assume that $\mathbf{x} \neq \mathbf{y}$. Then, we also have that $\underline{\mathbf{P}}\mathbf{x}^{m-1} \neq \underline{\mathbf{P}}\mathbf{y}^{m-1}$, so $\Phi(\mathbf{x}) \neq \Phi(\mathbf{y})$. We conclude that Φ is well-defined.
Since $(\underline{\mathbf{P}}\mathbf{x}^{m-1})_i$ is just a multiplication of elements of scalars, every element of Φ is continuous, thus Φ is continuous. □

Theorem 4.6 *If $\underline{\mathbf{P}}$ is a non-negative tensor of order m and dimension n that satisfies the conditions in 4.1.6, then there exists a non-zero non-negative vector $\bar{\mathbf{x}}$ such that $\underline{\mathbf{P}}\bar{\mathbf{x}}^{m-1} = \bar{\mathbf{x}}$ and $\sum_i x_i = 1$. In particular, if $\underline{\mathbf{P}}$ is irreducible, then $\bar{\mathbf{x}}$ must be positive.*

We follow Li and Ng here [14].
PROOF. According to the Brouwer Fixed Point Theorem, every continuous function f mapping a compact convex set to itself has a fixed point. Since Φ is continuous according to proposition 4.3 and Δ is compact and convex according to proposition 4.2, there is an $\bar{\mathbf{x}} \in \Delta$ such that $\Phi(\bar{\mathbf{x}}) = \bar{\mathbf{x}}$. Therefore, we can conclude that there must exist a point $\bar{\mathbf{x}}$ such that $\Phi(\bar{\mathbf{x}}) = \underline{\mathbf{P}}\bar{\mathbf{x}}^{m-1} = \bar{\mathbf{x}}$.
We have also seen in proposition 4.1 that every $\mathbf{x}$ is positive. □

Under certain conditions, we can even say that this limiting probability distribution $\bar{\mathbf{x}}$ is unique [14]. Let S be a subset of the state space, $\langle N \rangle$. Furthermore, let S' be the complement, i.e. $S' = \langle N \rangle \backslash S$. Define

$$\delta_m := \min_S \left\{ \min_{i_1,\ldots,i_{m-1} \in \langle N \rangle} \sum_{i \in S'} \underline{P}_{i,i_1,\ldots,i_{m-1}} + \min_{i_1,\ldots,i_{m-1} \in \langle N \rangle} \sum_{i \in S} \underline{P}_{i,i_1,\ldots,i_{m-1}} \right\}, \tag{4.3.2}$$

where $\underline{P}_{i,i_1,\ldots,i_{m-1}}$ are elements of the transition tensor $\underline{\mathbf{P}}$ of order m.

Theorem 4.7 *Let $\underline{\mathbf{P}}$ be an mth-order, n-dimensional, non-negative tensor that satisfies 4.1.6. If $\delta_m > \frac{m-2}{m-1}$, then the limiting distribution $\bar{\mathbf{x}}$ from theorem 4.6 is unique.*

Li and Ng [14] proved this particular theorem. This means we can indeed say that the limiting probability distribution is unique under certain conditions.

EXAMPLE 4.2 Recall the second-order Markov chain from figure 4.1 with transition matrix given by (4.1.3). In this case, the state space is $\langle N \rangle = \{\text{Rome, Madrid, Vienna}\}$.

For a third-order Markov chain, the condition from theorem 4.7 translates into

$$\delta_3 = \min_S \left\{ \min_{j,k \in \langle N \rangle} \sum_{i \in S'} \underline{P}_{ijk} + \min_{j,k \in \langle N \rangle} \sum_{i \in S} \underline{P}_{ijk} \right\} > \frac{1}{2}. \tag{4.3.3}$$

Define $S = \{\text{Rome, Madrid}\}$ and $S' = \{\text{Vienna}\}$. Then, for (4.3.3), we have

$$\begin{aligned} \delta_3 &= \min_{R,V} \left\{ \min_{j,k \in \{R,M,V\}} \sum_{i \in \{V\}} \underline{P}_{ijk} + \min_{j,k \in \{R,M,V\}} \sum_{i \in \{R,M\}} \underline{P}_{ijk} \right\} \\ &= \min_{R,V} \left\{ \min_{j,k \in \{R,M,V\}} \underline{P}_{Vjk} + \min_{j,k \in \{R,M,V\}} \left(\underline{P}_{Rjk} + \underline{P}_{Mjk} \right) \right\}. \end{aligned}$$

We know the transition probabilities from (4.1.4). We see that $\min_{j,k \in \{R,M,V\}} \underline{P}_{Vjk} = 0$, $\min_{j,k \in \{R,M,V\}} \underline{P}_{Rjk} = 0$ and $\min_{j,k \in \{R,M,V\}} \underline{P}_{Mjk} = 0$. This implies that $\delta_3 = 0 \not> \frac{1}{2}$. For this Markov chain, there does not exist a unique stationary distribution. We should be able to find

several Z-eigenvectors associated with a Z-eigenvalue of 1.

In this chapter, we have seen what a higher-order Markov chain is and how we can describe it mathematically. We have also seen that the higher-order Markov chain also has statonary distributions, which are unique in certain cases. In the next chapter, we will look at another kind of stochastic process, which resembles a second-order Markov chain: a vertex-reinforced random walk. This, we will study, and in particular, the spacey random walk.

Chapter 5

Vertex-Reinforced Random Walks

In this section, we discuss vertex-reinforced random walks. First, we give the definition of a vertex-reinforced random walk. After this, we discuss an example of a vertex-reinforced random walk: the spacey random walk. Furthermore, we explore the convergence of the spacey random walk.

5.1 The Definition of a Vertex-Reinforced Random Walk

A vertex-reinforced random walk is in itself quite abstract. In this section, we look at the definition of a vertex-reinforced random walk. In the next section, we will see how we can use it practically. The definition of the vertex-reinforced random walk was first introduced by Pemantle, but we follow the definition from Benaïm [1] and Gleich et al. [3] here.

Definition 5.1 *Let*

$$\begin{aligned} M : \Delta_N &\to \left\{\mathbf{P} \in \mathbb{R}^{N\times N} \middle| P_{ij} \geq 0,\, e^T\mathbf{P} = e^T\right\} \\ \mathbf{x} &\mapsto \mathbf{M}(\mathbf{x}) = \{M_{ij}(\mathbf{x})\} \end{aligned} \tag{5.1.1}$$

be a C^k $(k \geq 1)$ map. Let $\{X(n) : n \geq 0\}$ be a stochastic process with

$$X(0) = x(0), \tag{5.1.2}$$

$$s_i(n) = 1 + \sum_{s=1}^{n} Ind\{X(s) = i\}, \qquad \mathbf{w}(n) = \frac{\mathbf{s}(n)}{N+n}, \tag{5.1.3}$$

$$P\{X(n+1) = i|\mathcal{F}_n\} = [\mathbf{M}(\mathbf{w}(n))]_{i,X(n)}. \tag{5.1.4}$$

Here, $x(0)$ is the initial state, $\mathcal{F}_n$ is the σ-algebra generated by the random variables $X(i), i = 1, ..., n$ and $\mathbf{M}(\mathbf{w}(n))$ is an $N \times N$ column-stochastic matrix. We call the stochastic process $\{X(n) : n \geq 0\}$ that satisfies these conditions a vertex-reinforced random walk.

Here, Ind$\{\cdot\}$ denotes the indicator function, which is defined as

$$\text{Ind}\{X = x\} = \begin{cases} 1 \text{ if } X = x \\ 0 \text{ if } X \neq x. \end{cases} \tag{5.1.5}$$

The vertex-reinforced random walk has a feature that is completely different from the Markov chain: it involves the vector $\mathbf{w}(n)$, called the occupation vector, which changes every transition. It is actually dependent on the data. This means we cannot regard the vertex-reinforced random walk as

a Markov chain.

There is a special feature of a vertex-reinforced random walk that we can exploit: a vertex-reinforced random walk is related to a dynamical system. Define

$$\begin{aligned} \pi \,:\, &\left\{\mathbf{P} \in \mathbb{R}^{N\times N} \middle| P_{ij} \geq 0,\, e^T\mathbf{P} = e^T\right\} \to \Delta_N \\ &\mathbf{M}(\mathbf{x}) \mapsto \bar{\mathbf{x}}, \end{aligned} \tag{5.1.6}$$

where π maps a transition matrix to its stationary distribution. Now, there exists [1, 3] a relationship between the discrete reinforced-random walk and the dynamical system

$$\frac{d\mathbf{x}}{dt} = \pi(\mathbf{M}(\mathbf{x})) - \mathbf{x}. \tag{5.1.7}$$

This dynamical system involves π, the function which maps the transition matrix to its stationary distribution. The fact that this dynamical system exists, also says something about the existence of stationary distributions of the vertex-reinforced random walk, which we will see in section 5.4.

5.2 The Spacey Random Walk

A special case of the vertex-reinforced random walk is a spacey random walk. The spacey random walk resembles a second-order Markov chain. The second-order Markov chain depends on the two last states where both states are determined. The spacey random walk also depends on the two last states, just like the Markov chain. However, when the spacey random walker has visited its first state and is transitioning to the second, it completely *spaces out* and chooses its second last state at random. This makes it a very different process.
The transition probabilities for the spacey random walk are given in definition 5.2.

Definition 5.2 (Spacey Random Walk) *Let $(\Omega, \mathcal{F}_n, P)$ be a probability space, where $\mathcal{F}_i$ is the σ-algebra generated by the random variables $X(i), i = 1, ..., n$, and $\{X(n) : n \geq 0\}$ a stochastic process. The transition probabilities for a spacey random walk are defined by*

$$P\{Y(n) = k \,|\, \mathcal{F}_n\} = \frac{1}{n+N}(1 + \sum_{s=1}^{n} Ind\{X(s) = k\}), \tag{5.2.1}$$

$$\underline{P}_{ijk} = P\{X(n+1) = i | X(n) = j, Y(n) = k\}. \tag{5.2.2}$$

Here, Ind$\{\cdot\}$ denotes the indicator function, which is defined as in (5.1.5).
If we compare the transition probabilities of the spacey random walk with definition 5.1, the definition of a vertex-reinforced random walk, we can conclude that the spacey random walk is indeed an example of a vertex-reinforced random walk. The occupation vector of the spacey random walk is given by $\mathbf{w}_k = P\{Y(n) = k \,|\, \mathcal{F}_n\}$.

The manner in which we choose $Y(n)$ is dependent on the history. The indicator function counts how many times a state occurs in the history of the stochastic process. So, if we have visited state k many times before we arrive at n, the probability is also higher that we randomly choose k.

The transitions of the spacey random walk can also be expressed in a tensoric way, given in the following proposition.

Proposition 5.3 [3] *Let* $\mathbf{R} \in \mathbb{R}^{N \times N^2}$ *denote the unfolding of the tensor* $\underline{\mathbf{P}} \in \mathbb{R}^{N \times N \times N}$ *along the first index.*

$$\mathbf{R} := \left[\underline{P}_{\times\times 1} \,\middle|\, \underline{P}_{\times\times 2} \,\middle|\, \ldots \,\middle|\, \underline{P}_{\times\times N} \right] \tag{5.2.3}$$

Then, the Spacey random walk is a vertex-reinforced random walk defined by

$$\mathbf{x} \mapsto \sum_{k=1}^{N} \underline{P}_{\times\times k} \mathbf{x}_k = \mathbf{R} \cdot (\mathbf{x} \otimes \mathbf{I}). \tag{5.2.4}$$

PROOF. When we condition on $Y(n)$, we get

$$\begin{aligned} P\{X(n+1) = i | X(n) = j, \mathbf{w}(n)\} &= \sum_{k=1}^{N} P\{X(n) = i | X(n) = j, Y(n) = k\} P\{Y(n) = k | \mathbf{w}(n)\} \\ &= \sum_{k=1}^{N} \underline{P}_{ijk} \mathbf{w}_k(n) = [\mathbf{R} \cdot (\mathbf{w}(n) \otimes \mathbf{I})]_{ij} \end{aligned}$$

So indeed, the transition of a spacey random walker is defined by (5.2.4). □

Since a vertex-reinforced random walk is actually defined as a map $\mathbf{x} \mapsto \mathbf{M}(\mathbf{x})$, we can infer that $\mathbf{M}(\mathbf{x}) = \mathbf{R} \cdot (\mathbf{x} \otimes \mathbf{I})$. For the interested reader, we can even extend this notion. Consider a higher-order Markov chain where we forget more than one past state. Then, we have to choose more than one state at random. For a transition tensor of order m we will forget $m - 2$ states. Then, the spacey random walk is actually defined by

$$\mathbf{x} \mapsto \mathbf{R}(\underbrace{\mathbf{x} \otimes \ldots \otimes \mathbf{x}}_{m-2} \otimes I). \tag{5.2.5}$$

So we can indeed extend the spacey random walk to even higher-order data. However, we will only focus on the version of the spacey random walk where we choose one state at random.
Since we have now seen what a spacey random walk actually is, we will explore one of the simplest versions of it in the next section.

5.3 The $2 \times 2 \times 2$ case

In this section, we consider the simplest form of a spacey random walk: one with the minimum number of states, namely two. Let $\underline{\mathbf{P}} \in \mathbb{R}^{2 \times 2 \times 2}$ be a third order hypermatrix, where we denote the flattening $\mathbf{R}$ with

$$\mathbf{R} = \left[\begin{array}{cc|cc} a & b & c & d \\ 1-a & 1-b & 1-c & 1-d \end{array}\right], \tag{5.3.1}$$

where $a, b, c, d \in [0, 1]$. This is the most general hypermatrix $\underline{\mathbf{P}}$ for the $2 \times 2 \times 2$ case.
The transition matrix of the spacey random walked is linked to a dynamical system. The equilibrium of the dynamical system for the $2 \times 2 \times 2$ case is exactly solvable. This solution was first computed by Benson et al. [3] and is given in proposition 5.4.

Proposition 5.4 (The dynamical system of the $2 \times 2 \times 2$ case [3]) *The dynamical system of the general $2 \times 2 \times 2$ spacey random walk given in equation (5.3.1) is at equilibrium (i.e. $dx/dt = 0$) if and only if*

$$x = \begin{cases} \frac{1+2d-b-c \pm \sqrt{(b+c-2d-1)^2-4(a-b-c+d)d}}{2(a-b-c+d)} & \text{for } a+d \neq b+c \\ \frac{d}{1-b-c+2d} & \text{for } a+d = b+c,\ b+c \neq 1+2d \end{cases} \tag{5.3.2}$$

for any $x \in [0,1]$, if $a+d = b+c$ and $b+c = 1+2d$. In the case that $b+c = 1+2d$, the transitions look like

$$\mathbf{R} = \left[\begin{array}{cc|cc} 1 & b & 1-b & 0 \\ 0 & 1-b & b & 1 \end{array}\right]. \tag{5.3.3}$$

To compute the solutions for x for which the dynamical system is at equilibrium, we have to study the dynamical system of the spacey random walk. We can define a function $\mathbf{z}$ that maps a single probability to the probability simplex on two points. For the $2 \times 2 \times 2$ case, we can also introduce an explicit function for π.
Define the map $\mathbf{z}$ as

$$\mathbf{z} : [0,1] \to \Delta_1,\ x \mapsto \begin{bmatrix} x \\ 1-x \end{bmatrix}. \tag{5.3.4}$$

As we metioned, we can state π explicitly, which maps the transition matrix to its stationary distributions. We know that a 2×2 stochastic matrix has a unique eigenvector with eigenvalue $\lambda = 1$. This eigenvector can also be regarded as the stationary distribution of the first state. This function π maps the stochastic matrix to the first coordinate of this specific eigenvector. For π, this means

$$\pi\left(\begin{bmatrix} p & 1-q \\ 1-p & q \end{bmatrix}\right) = \mathbf{z}\left(\frac{1-q}{2-p-q}\right). \tag{5.3.5}$$

EXAMPLE 5.1 Let us consider one of the simplest transition tensors for the spacey random walk, the case where $a = b = c = d = 1/2$. In that case, we get the following flattening of the hypermatrix

$$\mathbf{R} = \left[\begin{array}{cc|cc} 1/2 & 1/2 & 1/2 & 1/2 \\ 1/2 & 1/2 & 1/2 & 1/2 \end{array}\right] \tag{5.3.6}$$

So here we have $a+d = 1/2+1/2 = 1$ and $b+c = 1/2+1/2 = 1$. We thus see that $a+d = b+c$. Then, we see that $b+c = 1/2+1/2 = 1$ and $1+2d = 1+2 \cdot 1/2 = 2$, thus $b+c \neq 2d$. Our system is at equilibrium at

$$x = \frac{d}{1+2d-b-c} = \frac{1/2}{1+2 \cdot 1/2 - 1/2 - 1/2} = 1/2$$

Since $\pi(A_{ij1}) = \mathbf{z}_1$, we conclude that $\mathbf{z}(x)$ must be the stationary distribution of the spacey random walk. The dynamical system of the spacey random walk is at equilibrium at $x = 1/2$ and its stationary distribution is given by $\mathbf{z}(x) = [1/2\ \ 1/2]^T$.

5.4 The Stationary Distribution of a Spacey Random Walk

The spacey random walk is a vertex-reinforced random walk and in that sense differs from a second-order Markov chain. However, as we have seen, the spacey random walk resembles the second-order

Markov chain. It turns out that the spacey random walk has stationary distributions, just like the second-order Markov chain. In this chapter, we will investigate whether the spacey random walk also converges to stationary distributions.

We first give a crude outline of when we think the spacey random walk is going to converge. Consider the occupation vector $\mathbf{w}(n)$ with $n \gg 1$. Now, we look at the spacey random walk at the nth step. Recall that the occupation vector changes with every step. Considering a stationary distribution, we expect a stationary occupation vector as well. Vice versa, the spacey random walk must be stationary when the occupation vector is stationary. Consider the spacey random walk at step $n+L$ with occupation vector $\mathbf{w}(n+L)$. Then, as we said, we expect that $\mathbf{w}(n) \approx \mathbf{w}(n+L)$. Furthermore, let $\mathbf{x}(n)$ be the stationary distribution of the spacey random walk, which should satisfy $\mathbf{x}(n) = \mathbf{w}(n+L)$. Heuristically, we have

$$\mathbf{w}(n+L) \approx \frac{n\mathbf{w}(n) + L\mathbf{x}(n)}{n+L} = \mathbf{w}(n) + \frac{L}{n+L}\big(\mathbf{x}(n) - \mathbf{w}(n)\big). \tag{5.4.1}$$

We can use (5.4.1) to obtain

$$\begin{aligned} \frac{d\mathbf{w}(n+L)}{dL} &\approx \lim_{L\to 0} \frac{\mathbf{w}(n+L) - \mathbf{w}(n)}{L} \\ &= \lim_{L\to 0} \frac{\mathbf{w}(n) + \frac{L}{n+L}\big(\mathbf{x}(n) - \mathbf{w}(n)\big) - \mathbf{w}(n)}{L} \\ &= \lim_{L\to 0} \frac{1}{n+L}\big(\mathbf{x}(n) - \mathbf{w}(n)\big) = \frac{1}{n}\big(\mathbf{x}(n) - \mathbf{w}(n)\big). \end{aligned} \tag{5.4.2}$$

So, in order for $\mathbf{w}(n) \approx \mathbf{w}(n+L)$, or $d\mathbf{w}(n+L)/dL$, we already see that we should end up with $\mathbf{x}(n) = \mathbf{w}(n)$.

We have some requirements for our stationary distribution: for instance, it should be unique. In other words, we want our transition tensor to have a unique Perron vector.

Definition 5.5 (Property B [3]) *A transition tensor $\underline{\mathbf{P}}$ of a spacey random walk satisfies Property B if the corresponding matrix $\mathbf{M}(\mathbf{w})$ has a unique Perron vector when $\mathbf{w} \in int(\Delta_N)$.*

The interior of the probability simplex is given by

$$\text{int}(\Delta_N) = \{\mathbf{x} = (x_1, x_2, .., x_N)^T \in \mathbb{R}^n \,|\, x_i > 0,\, 1 \le i \le N, \sum_{i=1}^{N} x_i = 1\}. \tag{5.4.3}$$

In other words, the interior of the probability simplex only contains positive, non-zero elements. The probability simplex itself can contain elements that have value zero.
If the transition tensor $\underline{\mathbf{P}}$ is strictly positive, we can infer from the definition of the vertex-reinforced random walk, definition, 5.1, that $P\{X(n+1) = i|\mathcal{F}_n\} = \mathbf{M}(\mathbf{w}(n))_{i,X(n)} > 0$ for every i. If the matrix $\mathbf{M}(\mathbf{w}(n))$ is thus strictly positive, we know from the Perron-Frobenius theorem (theorem 3.7) that $\mathbf{M}(\mathbf{w}(n))$ should have a unique positive Perron vector.
However, the transition tensor $\underline{\mathbf{P}}$ need not be strictly positive. More generally, a transition tensor satisfies property B if the corresponding matrix has a single recurrent class, a typical Markov property.

Theorem 5.6 [3] *A transition tensor $\underline{\mathbf{P}}$ of a spacey random walk satisfies Property B if and only if $\mathbf{M}(\mathbf{w})$ has a single recurrent class for some $\mathbf{w} \in int(\Delta_N)$.*

We give an outline of the proof, but details can be found in Benson et al. [3]. Remember from the properties of a Markov chain that it has a single class when it is irreducible. In order for this class to be recurrent, all the states in the class, in this case the whole Markov chain, should be recurrent. We give an outline of the proof [3]. The fact that $\mathbf{M}(\mathbf{w}(n))$ has a single recurrent class means that it only depends on its non-zero elements. In other words, Benson et al. say that $\mathbf{M}(\mathbf{w}(n))$ is completely determined by the graph of the non-zero elements. The resulting graph structure is given by the union of the graphs of every submatrix of $\mathbf{R}$. This graph structure is the same as that of $\mathbf{M}(\mathbf{w})$ for every $\mathbf{w} \in \text{int}(\Delta_N)$. Then, the the corresponding matrix $\mathbf{M}(\mathbf{w})$ should have a unique Perron vector.

If the dynamical system of a spacey random walk converges to a stationary point, we can also say that the occupation vector converges to a z-eigenvector of the transition tensor $\underline{\mathbf{P}}$. In order to prove that the dynamical system converges, we first need to refresh the notion of a flow and introduce the notion of an asymptotic pseudotrajectory. First, we introduce the definition of a flow.

Definition 5.7 (Flow [12]) *Let $U \subset \mathbb{R}^N$ be an open set. A continuous map $\Phi : \mathbb{R} \times U \to U, (t, y) \mapsto \Phi_t(y)$ is called a flow if it satisifies*

i. $\Phi_0(y) = y$ for all $y \in U$;

ii. $\Phi_s(\Phi_t(y)) = \Phi_{s+t}(y)$ for all $y \in U$ and $s, t \in \mathbb{R}$.

In general, an autonomous dynamical system $d\mathbf{x}/dt = f(\mathbf{x})$, where $f : \mathbb{R}^N \to \mathbb{R}^N$ is a vector field, with initial value $\mathbf{x}(0) = \mathbf{x}_0$, generates a flow $\Phi(t, \mathbf{x}_0)$ of the vector field f. This is also why the following proposition is true; the spacey random walk is linked to an autonomous dynamical system, which generates a flow.

Proposition 5.8 *Let the spacey random walk be defined by the dynamical system*

$$\frac{d\mathbf{x}}{dt} = \pi(\mathbf{M}(\mathbf{x})) - \mathbf{x} = f(\mathbf{x}), \tag{5.4.4}$$

with initial value $\mathbf{x}(0) = \mathbf{u}$. In this case, f generates a flow $\Phi : \mathbb{R}_+ \times \Delta_N \to \Delta_N$, where $\Phi(t, \mathbf{u})$ is the value of the initial value problem at time t.

Now, we introduce the definition of an asymptotic pseudotrajectory. This definition was first introduced by Benaïm [1].

Definition 5.9 (Asymptotic Pseudotrajectory [1]) *A continuous function $X : \mathbb{R}_+ \to \Delta_N$ is called an asymptotic pseudotrajectory of Φ if for any $L \in \mathbb{R}_+$, locally uniformly,*

$$\lim_{t \to \infty} ||X(t + L) - \Phi(L, \mathbf{x}(t))|| = 0$$

There are also certain conditions under which an asymptotic pseudotrajectory exists almost surely. Benaïm gives such a condition in the next theorem. Interested readers can find a proof in Benaïm [1].

Theorem 5.10 [1] *Let $\tau_0 = 0$ and $\tau_n = \sum_{i=1}^{n} 1/(i + 1)$. Define the continuous function W by $W(\tau_n) = \mathbf{w}(n)$ and let W be an affine function, a function that preserves the points from one affine space to another, on $[\tau_n, \tau_{n+1}]$. Then, we can say that W linearly interpolates the occupation vector on decaying time intervals. In that case, W is almost surely an asymptotic pseudotrajectory of Φ.*

In the next theorem, we combine proposition 5.8 and theorem 5.10. In proposition 5.8, we have already seen in proposition 5.8 that the spacey random walk generates a flow. In the next theorem, we see that we can also define an asymptotic pseudotrajectory.

Theorem 5.11 [3] *Let* $\mathbf{R}$ *be the flattening of an mth-order, N-dimensional stochastic tensor* $\underline{\mathbf{P}}$ *and let* $\mathbf{w}$ *be the occupation vector. Now, suppose that* $\mathbf{R}$ *satisfies Property B. In that case we can define a flow* $\Phi : \mathbb{R}_+ \times \Delta_N \to \Delta_N$*, where* $\Phi(t, \mathbf{u})$ *is the solution to*

$$\frac{d\mathbf{x}}{dt} = \pi\big(\mathbf{R} \cdot (\mathbf{x} \otimes \mathbf{I})\big) - \mathbf{x}, \qquad \mathbf{x}(0) = \mathbf{u}.$$

Then, for any $L \in \mathbb{R}_+$*,* W *is an asymptotic pseudotrajectory fo* Φ *almost surely, which means that* $\lim_{t\to\infty} ||W(t + L) - \Phi(L, \mathbf{x}(t))|| = 0$ *locally uniformly, where* W *is defined as in theorem 5.10. Furthermore, if the dynamical system of the spacey random walk converges, then the occupation vector* $\mathbf{w}(n)$ *converges to a stochastic* Z*-eigenvector* $\mathbf{x}$ *of* $\underline{\mathbf{P}}$*.*

Theorem 5.11 indeed tells us that, in order for the spacey random walk to converge, the occupation vector converges to a stochastic z-eigenvector of $\underline{\mathbf{P}}$. From theorem 5.10, we know that the function W is given by the occupation vector $\mathbf{w}(n)$. If the dynamical system of the spacey random walk converges, it converges to a stochastic Z-eigenvector of $\underline{\mathbf{P}}$. This dynamical system is represented in the asymptotic pseudotrajectory by Φ. The function $W = \mathbf{w}(n)$ thus converges to Φ, when time goes to infinity. In the end, we can indeed conclude that the occupation vector $\mathbf{w}(n)$ converges to a stochastic Z-eigenvector.

Again, here we expect a state that is visited often, to have a high centrality. The randomly chosen state is dependent on the history of the stochastic process. The preferred states here are the states that are common in the history, have many nodes and have a high probability of transitioning to. The stationary distribution of a spacey random walk should thus be a good candidate for a centrality measure.

Example 5.2 Recall the transition tensor of the $2 \times 2 \times 2$-case in (5.3.6). Now, from proposition 5.4, we had derived that the stationary distribution of this spacey random walk was given by $\mathbf{z}(x) = [1/2 \;\; 1/2]^T$.
In this case, let us say $\mathbf{z}(x) = \mathbf{x}$, which is nothing more than renaming the stationary distribution. Now, from theorem 5.11, it follows that the occupation vector $\mathbf{w}(n)$ converges to a stochastic z-eigenvector of P. We thus expect $\underline{\mathbf{P}}\mathbf{x}^2 = \mathbf{x}$. Let us check if this is true for the $2 \times 2 \times 2$-case for which we have already found the stationary distribution:

$$\underline{\mathbf{P}}\mathbf{x}^2 = \begin{bmatrix} \underline{P}_{111}x_1x_1 + \underline{P}_{112}x_1x_2 + \underline{P}_{121}x_2x_1 + \underline{P}_{122}x_2x_2 \\ \underline{P}_{211}x_1x_1 + \underline{P}_{212}x_1x_2 + \underline{P}_{221}x_2x_1 + \underline{P}_{222}x_2x_2 \end{bmatrix} = \begin{bmatrix} 4 \cdot (1/2)^3 \\ 4 \cdot (1/2)^3 \end{bmatrix} = \begin{bmatrix} 4/8 \\ 4/8 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} = \mathbf{x}.$$

So we see that the stationary distribution of the spacey random walk with transition tensor given by (5.3.6) is indeed a z-eigenvector of its transition tensor.

Just as the first- and higher-order Markov chain, the spacey random walk also has stationary distributions. We have already illustrated why we can use the stationary distributions for a first-order Markov chain as a centrality measure. The same applies to the stationary distributions of a spacey random walk: in the long run, we can see which states are visited most and which states are not. Intuitively, this gives us a notion of centrality. In the next chapter, we will see how we can put theory into practice, and execute numerical experiments with the first-order Markov chain and the spacey random walk.

Chapter 6

Empirical Markov Chains & SRWs

In the next two chapters, we will analyse European air traffic. In the last chapters, we have seen what Markov chains are and what a spacey random walk is. When Benson et al. [3] did numerical experiments with Markov chains and the spacey random walk, they found out that the spacey random walk gave a better result than a second-order Markov chain. Also, the data was unavailable for the second-order Markov chain. This is why we will only consider the first-order Markov chain and spacey random walk.
In this section, we describe how we can use the first-order Markov chain and spacey random walk in numerical experiments. We discuss the techniques we use and how efficient our numerical implementation is.

6.1 First-Order Markov Chain

In this section, we discuss the numerical computation for the transition matrix of a first-order Markov chain. Also, we see how we can compute the stationary distribution.
Our transition matrix should satisfy

$$P_{ij} = \begin{cases} \hat{P}_{ij} & \text{if there is a flight from } i \text{ to } j \\ 0 & \text{if there is no flight from } i \text{ to } j, \end{cases} \tag{6.1.1}$$

where

$$\hat{P}_{ij} = \frac{\#\,\text{passengers from } j \text{ to } i}{\sum_l \#\,\text{passengers from } j \text{ to } l}, \tag{6.1.2}$$

which are the maximum likelihood estimators for the first-order Markov chain [3]. This way, we can build our transition matrix $\mathbf{P}$.

Listing 6.1: Constructing a transition matrix for a first-order Markov chain

```
function s = firstMC(i,j,k)
% Function of i and j, if there is a flight from i to j, and k, the number of
    passengers
N = max([max(i),max(j)]); s = sparse(i,j,k,N,N); c = sum(s,1);
for j = 1:N % Normalize to a column-stochastic matrix
        if c(j) > 0
                s(:,j) = s(:,j)./c(j);
        end
end
```

We can determine our stationary distributions according to theorem 2.2, which states that we have to determine the eigenvector of $\mathbf{P}$ with eigenvector $\lambda = 1$.

In order to obtain the eigenvector, we use a power iteration, also called the power method. The power method approximates the eigenvalue corresponding to the largest eigenvalue very quickly. The method is, for instance, described by Golub & Van Loan [8]. A single power iteration is given by

$$\mathbf{x}_k = \frac{\mathbf{P}^k \mathbf{x}_0}{||\mathbf{P}^k \mathbf{x}_0||}. \tag{6.1.3}$$

Further on in this chapter, we briefly discuss why this method converges to an eigenvector of $\mathbf{P}$.

We can implement the power method in Matlab with the following code.

Listing 6.2: Computing the stationary distribution of a first-order Markov chain

```
function x = stat1MC(P, tol)
% Function of P, the transition matrix, and tol, the tolerance level of the
    power method
[N,N] = size(P); % Determine the number of states
e = ones(N,1); x = e/N; oldx = zeros(N,1);
while norm(x-oldx) > tol
        oldx = x;
        x = P*x;
        x = x/sum(x)
end
```

With the power method, we can easily compute the stationary distributions of the Markov chain. These are the centrality measures that we aim to compute. In the next section, we will aim to obtain a similar result for the spacey random walk.

6.2 Spacey Random Walk

In this section, we introduce the numerical computation for the transition tensor of a spacey random walk. We used a different method to approximate our transition tensor than Benson et al [3] did. Benson et al. were working with trajectory data. Our method, however, is based on the transition matrix of the first-order Markov chain. The main reason for this is that, unlike trajectory data, we only have data that is suitable for a first-order Markov chain. In order to use our data, we made some more assumptions than in the original paper.

First, we consider our transition matrix $\mathbf{P}$ of the first-order Markov chain. We can establish a relationship between a first-order Markov chain and a spacey random walk, namely

$$P\{X(n+1) = i | X(n) = j\} = \sum_{k=1}^{N} P\{Y(n) = k\} \underline{P}_{ijk} = \sum_{k=1}^{N} \mathbf{w}_k(n) \underline{P}_{ijk}. \tag{6.2.1}$$

Then, we generate trajectories by simulating a random walk through the Markov chain. We generate trajectories where we choose the first two states. Note that the state we chose first has no consequence for the later trajectory, since we use a first-order Markov chain here. Now, we are actually creating history to work with.
We based the simulation algorithm on the `simulate` function from the Econometrics Toolbox.

Listing 6.3: Simulating Trajectories

```
function X = simulate2(P,numSteps,X0,X1)
% Function of P, the transition matrix of the first-order Markov chain,
    numsteps, the number of steps in a simulation, and X0 and X1, the two
    initial values
[N, N] = size(P); % Determine the number of states
X = zeros(1+numSteps,1);
simState = X0; simState2 = X1; % Set initial values
X(1,1) = simState; X(2,1) = simState2;
for i = 3:(1+numSteps) % Simulate states during the number of steps
        u = rand;
        cs = cumsum(P(:,simState2));
        if cs(N,1) == 0
                X(i,1) = simState2;
        else
                simState2 = find(u < cumsum(P(:,simState2)),1);
                X(i,1) = simState2;
        end
end
```

We simulated trajectories for every possible i and j. With this data, we can approximate $\mathbf{w}(n)$, the occupation vector. From (5.4.1) we know that $\mathbf{w}(n) \approx \mathbf{x}(n)$ for big n. We also know that we approximate the stationary distribution of the first-order Markov chain, which should be roughly equal to the spacey random walk for the same data, by transitioning a vast number of times. This is why we approximate the occupation vector by several large trajectories of the first-order Markov chain.
Then, we come up with the following log-likelihood problem:

$$\mathcal{L}(\underline{\mathbf{P}}) = -\log\left(\sum_{k=1}^{N} \mathbf{w}_k \underline{P}_{ijk}\right). \tag{6.2.2}$$

Note that $\mathcal{L}(\underline{\mathbf{P}})$ gives us a minimization problem with N^3 unknown variables. This means we have to minimize a vector function. Furthermore, we also want

$$\sum_{i=1}^{N} \underline{P}_{ijk} = 1, \qquad 0 \leq \underline{P}_{ijk} \leq 1. \tag{6.2.3}$$

For practical reasons, we use the vectorisation of the tensor $\underline{\mathbf{P}}$, i.e. $\underline{\mathbf{p}}_{vec} \in \mathbb{R}^{N^3}$. Then, we use the `fminsearchbnd` function [5] to find the minima in the interval $[0, 1]$. This function searches for minima in a bounded domain. Then, we normalize our tensor.

Listing 6.4: Constructing a transition tensor for a spacey random walk

```
function p = SRW(P, numSteps)
 % Function of P, the transition matrix of the first-order Markov chain and
     numSteps, the number of steps in a simulation
[N,N] = size(P); % Determine the number of states
X0 = repelem(1:N, N); X1 = [1:N]; X1 = repmat(X1, 1, N); % Generate initial
    values
for i = 1:N^2 % Simulate for all possible initial values
        X = simulate2(P, numSteps, X0(i), X1(i)); mat(i,:) = X;
end
```

```
 % Count the number of occurances of a state in the simulation
for i = 1:N^2
        w(i,:) = histcounts(mat(i,:), N, 'Normalization','probability');
end
w = reshape(w',[],1);
x0 = ones(N^3,1)/N^3; x1 = zeros(N^3,1); x2 = ones(N^3, 1);
 % Solve minimization problem
x = fminsearchbnd(@(p) -log(sum(w.*p)),x0,x1,x2);
 % Reshape to a tensor from the vectorization
p = reshape(x, [N, N, N]); sp = sum(p,1);
 % Normalize to a column-stochastic tensor
for k = 1:3
        for j = 1:3
                p(:,j,k) = p(:,j,k)./sp(1,j,k);
        end
end
```

We have already seen that $\mathbf{w}(n) \approx \mathbf{x}(n)$ for large n. Gleich et al. [7] heuristically supposed that we can approximate the transitions of the spacey random walk as a first-order Markov chain:

$$P\{X(n+1) = i | X(n) = j\} \approx \mathbf{w}_k \underline{P}_{ijk}. \tag{6.2.4}$$

Then, consider the frontal slices $\mathbf{R}_k$ of the transition hypermatrix $\underline{\mathbf{P}}$, then a transition of the first-order Markov chain becomes

$$\sum_{k=1}^{N} \mathbf{R}_k \mathbf{w}_k = \mathbf{R} \cdot (\mathbf{w} \otimes \mathbf{I}), \tag{6.2.5}$$

which is exactly (5.2.4), so that affirms our assumption that we can approximate a spacey random walk as a Markov chain. This directly implies that we can also approximate the stationary distributions of a spacey random walk by

$$\mathbf{x} = \underline{\mathbf{P}}(\mathbf{x} \otimes \mathbf{x}) = \underline{\mathbf{P}}\mathbf{x}^2, \tag{6.2.6}$$

the z-eigenvalues of $\underline{\mathbf{P}}$ with $\lambda = 1$. This is also in accordance with the more formal convergence to this stationary distribution given in theorem 5.11.
We compute these Z-eigenvalues of the probability tensor by the so-called classical higher-order power method [2]. This resembles the power method for matrices. This is also an iterative method, where we compute $\mathbf{x}$ by

$$\mathbf{x}_{k+1} = \frac{\underline{\mathbf{P}}\mathbf{x}_k^{m-1}}{||\underline{\mathbf{P}}\mathbf{x}_k^{m-1}||}. \tag{6.2.7}$$

An elaboration on the power method for tensors can be found in Benson & Gleich [2]. We will only use (6.2.7) here to find the stationary distribution quickly.
To implement this in Matlab, we first need a function for a multiplication of tensors according to (3.2.1).

Listing 6.5: Multiplication for a third-order tensor

```
function a = tensormltp(P,x)
% Function of P, a third-order tensor, and x, a vector
[ni,nj,nk] = size(P); %
for i = 1:ni
```

```
5           for j = 1:nj
6                   for k = 1:nk
7                           a(i) = P(i,j,k)*x(j)*x(k);
8                   end
9           end
10  end
```

Then, we can use the power method to compute our stationary distribution.

Listing 6.6: Computing the stationary distribution of a spacey random walk

```
1   function x = statSRW(P, tol)
2   % Function of P, the transition tensor and tol, the tolerance level of the
        power method
3   [N,N,N] = size(P); % Determine the number of states
4   e = ones(N,1); x = e/N; oldx = zeros(N,1);
5   while norm(x-oldx) > tol
6           oldx = x;
7           x = (tensormltp(P,x))/sum(tensormltp(P,x));
8   end
```

In this manner, we can compute the stationary distributions of the spacey random walk.

6.3 Convergence of the Power Method

In this section, we give some more backgrounds on the methods used to compute the eigenvector of the matrix respectively tensor. First, we give a proof of convergence of the power method for matrices and say something about its numerical complexity. Then, we explain some more of the higher-order power method and where it comes from.

Theorem 6.1 (Convergence of the Power Method) *Let $\mathbf{A} \in \mathbb{R}^N$ be a matrix with N linearly independent eigenvalues. Then, a power iteration is defined as*

$$\mathbf{x}_k = \frac{\mathbf{A^k x}_0}{||\mathbf{A}^k \mathbf{x}_0||}. \tag{6.3.1}$$

If k approaches infinity, this method approaches the largest eigenvalue of $\mathbf{A}$.

Proof. By assumption, we say that $|\lambda_1| \geq |\lambda_2| \geq ... \geq |\lambda_N|$. The corresponding eigenvectors are $\mathbf{v}_1, ..., \mathbf{v}_N$. Since these eigenvectors are linearly independent, they form a basis for $\mathbb{R}^N$. Then, we can write

$$\mathbf{x}_0 = c_1\mathbf{v}_1 + c_2\mathbf{v}_2 + ... + c_N\mathbf{v}_N, \tag{6.3.2}$$

for some constants $c_1, ..., c_N \in \mathbb{R}$ and $c_k \neq 0$ for all k. We can plug the expression from (6.3.2) into the expression given in (6.3.1), which gives us

$$\mathbf{x}_k = \frac{\mathbf{A}^k \mathbf{x}_0}{||\mathbf{A}^k \mathbf{x}_0||} = \frac{\mathbf{A}^k(c_1\mathbf{v}_1 + ... + c_N\mathbf{v}_N)}{||\mathbf{A}^k(c_1\mathbf{v}_1 + ... + c_N\mathbf{v}_N)||}. \tag{6.3.3}$$

Now, note that $\mathbf{A}^k\mathbf{v}_j = \lambda_j^k\mathbf{v}_j$ for all j. We can illustrate this with an example of $k = 2$:

$$\mathbf{A}^2\mathbf{v}_j = \mathbf{A}(\mathbf{A}\mathbf{v}_j) = \mathbf{A}\lambda_j\mathbf{v}_j = \lambda_j(\mathbf{A}\mathbf{v}_j) = \lambda_j^2\mathbf{v}_j.$$

Rewriting (6.3.3), we get

$$\mathbf{x}_k = \frac{c_1\lambda_1^k\mathbf{v}_1 + ... + c_N\lambda_N\mathbf{v}_N}{||c_1\lambda_1^k\mathbf{v}_1 + ... + c_N\lambda_N\mathbf{v}_N||} = \frac{\lambda_1^k}{||\lambda_1^k||}\frac{c_1\mathbf{v}_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^k\mathbf{v}_2 + ... + c_N\left(\frac{\lambda_N}{\lambda_1}\right)^k\mathbf{v}_N}{||c_1\mathbf{v}_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^k\mathbf{v}_2 + ... + c_N\left(\frac{\lambda_N}{\lambda_1}\right)^k\mathbf{v}_N||}.$$

Since $|\lambda_1| \geq |\lambda_i|$ for all $i = 2, .., N$, we know that $\left(\frac{\lambda_i}{\lambda_1}\right)^k \to 0$ as $k \to \infty$. So, as $k \to \infty$,

$$\lim_{k\to\infty} \mathbf{x}_k = \lim_{k\to\infty} \frac{\lambda_1^k}{|\lambda_1^k|}\frac{c_1\mathbf{v}_1}{||c_1\mathbf{v}_1||}. \tag{6.3.4}$$

So we can conclude that $\mathbf{x}_k$ indeed converges to the largest eigenvalue. □
We see that the power method for matrices converges to an eigenpair. However, we have seen another power method: a power method for higher-order tensors. For this higher-order method, we will not show a proof, but we will show where the results come from.

The Symmetric Shifted Higher-Order Power Method (SS-HOPM) was developed by Kolda & Mayo [11]. For a symmetric tensor $\underline{\mathbf{A}}$, this shifted higher-order power method is similar to the one given in (6.2.7), but the iterates are now given by

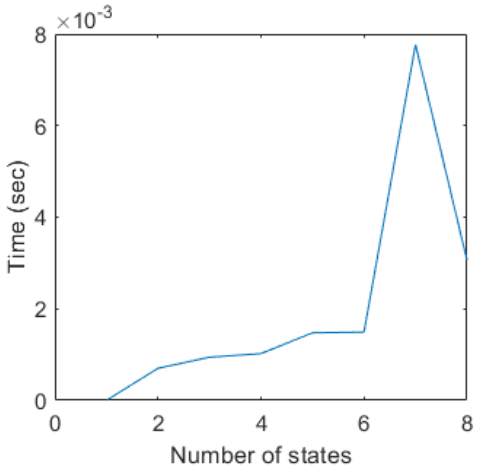
$$\mathbf{x}_{k+1} = \frac{\frac{1}{1+\gamma}(\underline{\mathbf{A}}\mathbf{x}_k^{m-1} - \gamma\mathbf{x}_k)}{||\frac{1}{1+\gamma}(\underline{\mathbf{A}}\mathbf{x}_k^{m-1} - \gamma\mathbf{x}_k)||_2}. \tag{6.3.5}$$

For a right choice of γ, the shifted higher-order power method converges. The starting points of the higher-order power method are of great importance, because they also make sure to which eigenpairs the power method converges. The computational complexity of SS-HOPM is $\mathcal{O}(n^m)$ for a tensor of order m. Compared to the regular power method for matrices, this is actually the same: a matrix, a tensor of the second order, has computational complexity of $\mathcal{O}(n^2)$.
Benson & Gleich showed that the shifted higher-order power method can be derived from the dynamical system of the spacey random walk [2]. Here, the iterates are computed by the expression given in (6.2.7). We only see a minor difference regarding the norm that is used.
The classical higher-order power method arises when $\gamma = 0$. The classical higher-order power method does not converge in all cases [11]. For convenience, we used the classical higher-order power method in this thesis.

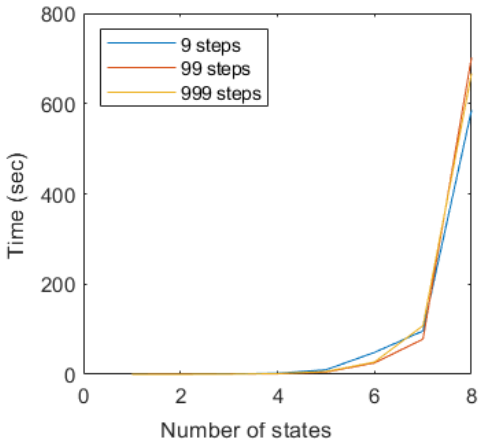
In the next section we compare the two different methods of computing the stationary distributions for the first-order Markov chain and the spacey random walk.

6.4 Efficiency

Since the spacey random walk is constructed from the first-order Markov chain, we expect the first-order Markov chain to be much more efficient than the spacey random walk. For the spacey random walk, we have to construct the transition matrix from the first-order Markov chain to be able to construct the transition tensor for the spacey random walk. This is reflected in the computation time of both the transition matrix of the first-order Markov chain and the transition tensor of the spacey random walk.



(a) Computation time for the first-order Markov chain



(b) Computation time for the spacey random walk with simulation of a varying number of steps

Figure 6.1: Computation time for the transition matrix or tensor for the first-order Markov chain and the spacey random walk

In the figures above, we show the computation time for computing the transition matrix for the first-order Markov chain and the transition tensor for the spacey random walk, respectively. The computation time for the spacey random walk transition tensor rapidly increases with the number of states, especially when we compare it to the first-order Markov chain transition matrix. When we come to just eight states, it is already a thousand times more time-efficient to compute the transition matrix of a first-order Markov chain than to compute the transition tensor for the spacey random walk.

In figure 6.1b, we can see that the computation time does not increase when increasing the number of steps used in the simulation. This implies that the function `fminsearch` takes up the most time. Another minimization method would thus heavily decrease the computation time, a method such as a projected gradient descent method posed in Benson et al. [3], but goes beyond the scope of this thesis.

Chapter 7

European Air Traffic

In this chapter, we discuss the centralities of European air traffic. In chapter 6, we have seen that the spacey random walk transition tensor takes very long to compute. This is why we have chosen to compare the first-order Markov chain and the spacey random walk with only eight states. We look at the first-order Markov chain alone with 447 states. In addition to this, we investigate what the effect of Lelystad Airport will be on European air traffic.

We got all our data from the database of Eurostat [6], where we looked at 'Air transport' > 'Air transport measurement - passengers' > 'Detailed air passenger transport by reporting country and routes'. We looked at all the passengers carried from the airports in countries from the European Union in 2017. For the selection of the 447 airports, we only picked the airports that were big enough to have the amount of passengers registrated on Eurostat. The locations of the airports come from the Open Flights database [17].

7.1 Centrality Measures in Comparison

For a dataset consisting of taxi trajectories in Manhattan, New York City, Benson et al. [3] simulated a first-order Markov chain, second-order Markov chain and a spacey random walk. They found out that the first-order Markov chain they simulated gave the biggest errors when computing the stationary distribution. The spacey random walk and second-order Markov chain, however, gave very accurate results for the stationary distribution.
Since we have data that is only suitable for a first-order Markov chain, we choose to construct our transition tensor for the spacey random walk by means of the method we described in section 6.2. However, in section 6.3, we encountered that it was very inefficient to compute the spacey random walk. This is why we chose to simulate both a first-order Markov chain and spacey random walk with only eight states. Then, we will compare them. In the next section, we will study the first-order Markov chain with 447 states.

We have chosen airports in eight different big cities in Europe to work with: Amsterdam, Athens, Frankfurt, London, Madrid, Oslo, Paris and Rome.

We used simulated trajectories of a length of 1000. For both the power method for matrices and tensors, we used a tolerance of 10^{-3}. Eventually we computed the stationary distributions for both the first-order Markov chain and the spacey random walk. These results are shown in both figure 7.1 and 7.2.

In 7.1, we can see the values of the vectors that represent the stationary distributions.

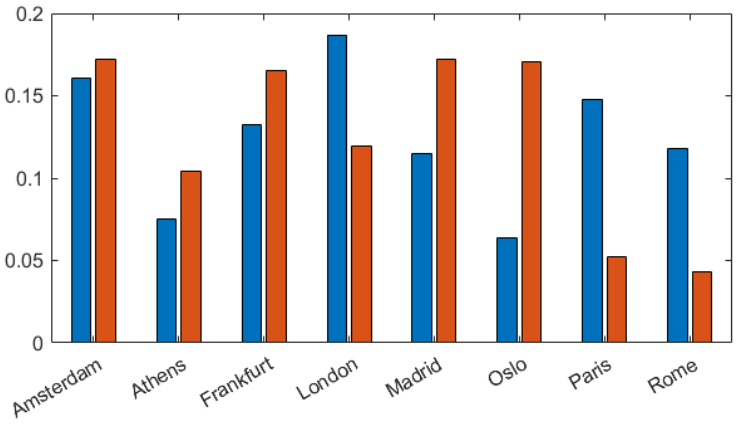


Figure 7.1: The stationary distributions for the first-order Markov chain (blue) and spacey random walk (red)

We see that the stationary distributions from the first-order Markov chain and spacey random walk show some discrepancy. At first sight, we would say that the first-order Markov chain actually tends to be give a more realistic view in this case: it is quite unlikely that the airports in Madrid and Oslo should be more central than than London and Paris.

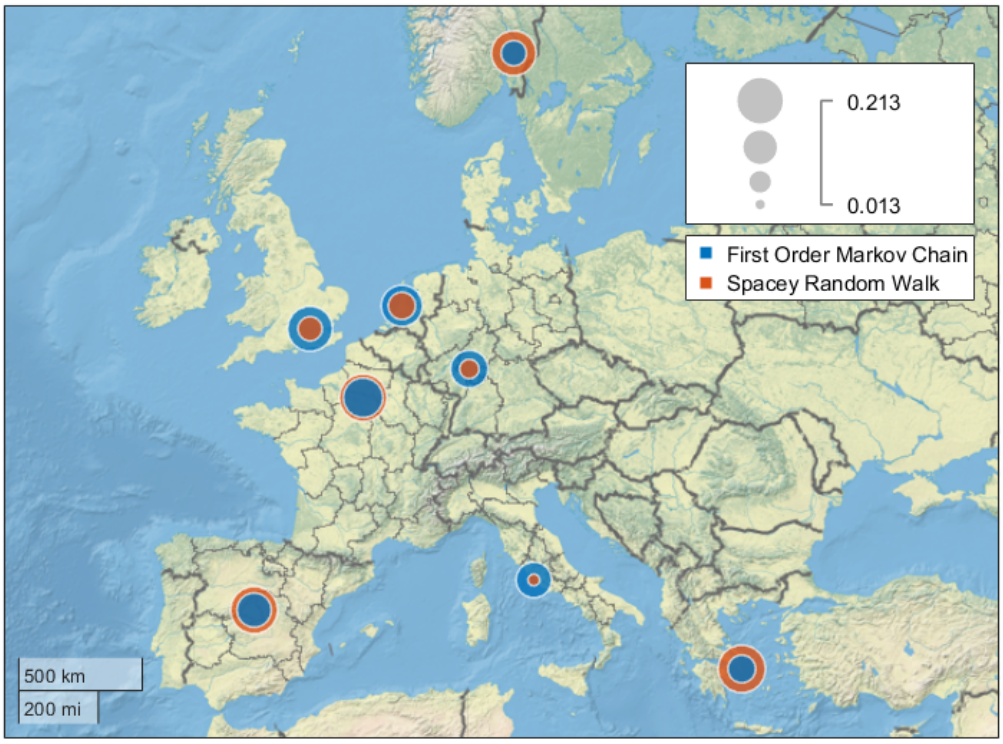


Figure 7.2: The stationary distributions for the first-order Markov chain (blue) and spacey random walk (red) shown in a map of Europe

Simulating the Markov chain could give us the discrepancies we see. Say we start in Oslo and Madrid when we simulate a Markov chain. Then, for example, since there are so many people usually flying from Madrid to Oslo, our simulation gives high probabilities from transitioning from Madrid to Oslo. This could alter our views, if it is really hard to reach Madrid. Also, since our data only contains the history of the first-order Markov chain, we do not have any additional information which could provide a more precise result.

These numerical experiments affirmed the assumption that the spacey random walk would be difficult to compute. In our case, the spacey random walk gave worse results than the first-order Markov chain did.
In the next section, we will only consider a first-order Markov chain to investigate a total number of 446 European airports to the air traffic above Europe.

7.2 Lelystad Airport

Schiphol Airport plays a major role in European air traffic: it is one of the biggest airports in Europe. This immediately implies that Schiphol Airport has an important economic position. It has always been a difficult political issue: can we further expand Schiphol Airport, even if it negatively affects local residents?

> Het netwerkverkeer op Schiphol is essentieel voor de economisch waardevolle functie van de nationale luchthaven als belangrijke hub in Europa. Gezien de schaarse capaciteit voor luchtvaart in Nederland en de aanzienlijke belasting van het milieu die de luchtvaart op dit moment en op middellange termijn nog met zich meebrengt, hecht het kabinet eraan beschikbare capaciteit in te zetten om de hubfunctie van Schiphol maximaal te ondersteunen. Conform het Aldersakkoord wil het kabinet Lelystad Airport ontwikkelen als overloopluchthaven voor vakantievluchten van Schiphol, zodat op Schiphol ruimte kan worden geboden om het (inter)continentale netwerk te versterken.
>
> *Traffic to and from Schiphol is essential for the economically valuable function of the national airport as important hub in Europe. Considering the scarce capacity of aviation in The Netherlands and the substantial environmental impact aviation has at this moment and still for a medium long period, the cabinet adheres to using the available capcity to maximally support the hubfunction of Schiphol. In conformity with the Alders agreement, the cabinet pledges to develop Lelystad Airport to relocate holiday flights from Schiphol to Lelystad, in order to give Schiphol space to strengthen their (inter)continental network.*
>
> C. Van Nieuwenhuizen-Wijbenga, Minister of Infrastructure and Water Management [16]

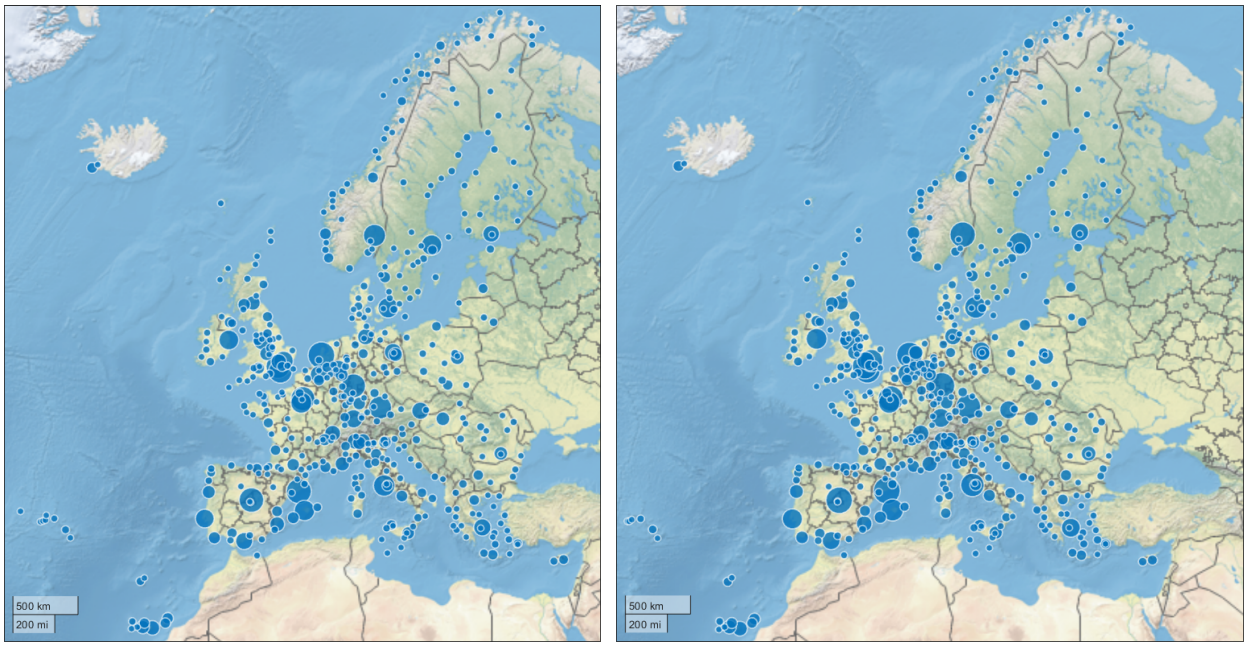
In mid 2019, the Minister of Infrastructure and Water Management will come with a report that investigates the perspectives of Lelystad Airport. It will independently investigate what destinations aircrafts will fly to from Lelystad Airport. Since I have started my data analysis before, I made an estimation. It is already known what the amount of flights that will take off from Lelystad Airport will be once it opens in 2020: 4000 flights per year. After that, Lelystad Airport will expand to a number of 10 000 flights per year in 2023. In 2023, Lelystad Airport will be evaluated. Then, it will be determined how the airport will expand to a total number of 25 000 flights per year.

Year	Estimated Number of Flights per Year
2020	4000
2021	7000
2023	10 000
After 2023	25 000

Table 7.1: The estimated growth of Lelystad Airport

The flights that will be transferred from Amsterdam Schiphol Airport to Lelystad airport will only be recreational flights. The destinations of these flights will especially comprise of destinations in the neighbourhood of the Mediterranean.
The European Union forbids Lelystad Airport to open for new destinations. Flights from Lelystad Airport are not allowed to go to new destinations, they can only transfer destinations from Amsterdam Schiphol Airport.

At the moment I am writing this thesis, Van Nieuwenhuizen has just appointed a research team to do a market research: which airlines are willing to fly from Lelystad Airport to which destinations? Since I had to wait for this, I made an estimation myself. I mainly chose destinations around the Mediterrenean, but also some of the bigger cities in Europe. These cities are very popular for city trips. A detailed overview of the flights that will be transferred from Schiphol to Lelystad airport can be found in Appendix B.



(a) Europe at present (b) Europe around 2030

Figure 7.3: A view of the centrality of the airports in Europe with recreational flights shifted from Schiphol Airport to Lelystad Airport, modelled using a first-order Markov chain.

In figure 7.3, we can see the difference in centrality for Europe at present and around 2030. The bigger the size of the bubble, the higher the centrality. If we look at the destinations in the Mediterranean area, we can see that the centrality of the airports there does not change. This is

actually logical, since we only transferred flights from Schiphol to Lelystad. There are no new flights which go to the Mediterranean.
If we look closely at the Netherlands, we see an airport of medium size arising in the middle, while Schiphol stays relatively the same size. Almost 500 000 flights depart from or arrive at Schiphol Airport every year, so 25 000 transferred flights apparently do not affect the centrality of Schiphol. We zoom in at the Netherlands to give this a closer look.

(a) At present

(b) In 2020

(c) In 2021-2022

(d) In 2023

(e) In the far future

Figure 7.4: A view of the airports in the Netherlands with recreational flights shifted from Schiphol Airport to Lelystad Airport, modelled using a first-order Markov chain.

If we zoom in at the Netherlands, we can see the growth of the centrality of Lelystad airport more clearly. In figure 7.4, we can see the centralities of the airports in the Netherlands. We can see how the centralities develop over the course of time: we see the centralities at present, in 2020, in 2021-2022, in 2023 and in the far future. In the figure, we still considered all 447 airports in Europe. We see that Lelystad Airport will eventually be the most central airport in the Netherlands after Schiphol when considering European air traffic. Also, comparing Lelystad with Brussels, which we can also see on figure 7.4, we see that Lelystad Airport can definitely not remain unnoticed in the future.

I think we can conclusively say that Lelystad Airport is certainly aiming on becoming an airport with a relatively high centrality. Still, the future will show if ever there will depart 25 000 flights per year from the capital of Flevoland.

Chapter 8

Conclusion and Discussion

Centrality Measure

Our aim in this thesis was to define a centrality measure using Markov chains and the spacey random walk. We have seen that the Markov chain and spacey random walk both have stationary distributions; distributions to which the stochastic process tends to go when transitioning an infinite number of times. Stationary distributions always exist and are unique in some cases. These stationary distributions all turn out to be very good candidates for a centrality measure.

Perron-Frobenius

In the beginning of the twentieth century, Perron proved the Perron-Frobenius theorem for matrices. In 2008, Chang, Pearson & Zhang proved the Perron-Frobenius theorem for H-eigenpairs. For the existence and uniqueness of the stationary distribution of the higher-order Markov chain, we could not use a result of the Perron-Frobenius theorem. The power of the Perron-Frobenius theorem mainly lies in the attribution of certain properties to positive matrices. However, we have seen, for stochastic tensors, the stationary distributions always exist as a consequence of Brouwer's fixed point theorem. Also, Li and Ng [14] proved that this stationary distribution is unique in some cases. With these results combined, we could set up a similar theorem for Z-eigenpairs of stochastic tensors, maybe even positive tensors.

Efficiency

Computing the transition tensor for a spacey random walk was more inefficient than computing the transition matrix for a first-order Markov chain. In the future, we could use two significant improvements to decrease the computation time for the spacey random walk.

The most difficult part in finding the transition tensor of the spacey random walk, was the finding of all the tensor elements by means of a minimization problem. We used a built-in Matlab function to find the elements. It is very probable that there are better minimization methods out there that could improve the efficiency of the computation.

The other difficult part lies in finding the stationary distribution of the spacey random walk. The higher-order power method has a higher computational complexity than the regular power method for matrices has, $\mathcal{O}(n^3)$ compared to $\mathcal{O}(n^2)$. Furthermore, for badly chosen γ, this power method does not even converge to an eigenpair. Also, when the initial values are badly chosen, the power method does not converge as well. Conclusively, the higher-order power method is quite tricky.

In any case, it would be beneficient for the computation time to explore other methods to find Z-eigenpairs of a tensor. It would also be profitable to search for methods in which convergence is ensured.

European Air Traffic

Since we had data that was suitable for a first-order Markov chain, we had to choose another method to be able to work with the data. Further research could focus on trajectory data of European aircrafts. Trajectory data comprises of more information than the data for the first-order Markov chain does. This is probably why we obtained worse results with the spacey random walk (which we based on the first-order Markov chain); we could not work with more information. Benson et al. found out that the spacey random walk gave better results than the second-order Markov chain when it was based on trajectory data [3]. If we could use trajectory data, we could obtain a better result.

Bibliography

[1] M. Benaïm. *Vertex-reinforced Random Walks and a Conjecture of Pemantle.* The Annals of Probability 25(1). 1997.

[2] A.R. Benson, D.F. Gleich. *Computing Tensor Z-eigenvectors with Dynamical Systems.* May 2018. Retrieved from `https://arxiv.org/pdf/1805.00903.pdf`.

[3] A.R. Benson, D.F. Gleich, L.-H. Lim. *The Spacey random walk: a stochastic process for higher-order data.* SIAM Review, 59(2). 2017.

[4] K.C. Chang, K. Pearson, T. Zhang. *Perron-Frobenius Theorem for Nonnegative Tensors.* Commun. Math. Sci. 6(2). 2008.

[5] J. D'Errico. *Bound constrained optimization using fminsearch.* Retrieved from `https://nl.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd-fminsearchcon`, May 10, 2019.

[6] Eurostat. *Transport Database.* Retrieved from `https://ec.europa.eu/eurostat/web/transport/data/database`, March 8, March 9, May 7, 2019.

[7] D.F. Gleich, L.-H. Lim, Y. Yu. *Multilinear PageRank.* SIAM J. Matrix Anal. Appl., 36(4), pp. 1507-1541, 2015.

[8] G.H. Golub, C.F. Van Loan. *Matrix Computations.* (Fourth Edition). Baltimore, Maryland: Johns Hopkins University Press.

[9] S. Hu, L. Qi. *A Necessary and Sufficient Condition for Existence of a Positive Perron Vector.* SIAM J. Matrix Anal. Appl., 37(4), 1747–1770, 2016.

[10] T.G. Kolda, B.W. Bader. *Tensor Decompositions and Applications.* SIAM Review, 51(3). 2009.

[11] T.G. Kolda, J.R. Mayo. *Shifted Power Method for Computing Tensor Eigenpairs.* SIAM J. Matrix Anal. Appl., 32(4), 1095-1124, 2011.

[12] Y. Kuznetsov, H. Hanssmann. *Basis Differentiaalvergelijkingen.* Utrecht University (January 2018).

[13] L. De Lathauwer, B. De Moor, J. Vandewalle. *A Multilinear Singular Value Decomposition.* SIAM J. Matrix Anal. Appl. 21(4). 2000.

[14] W. Li, M.K. Ng. *On the Limiting Probability Distribution of a Transition Probability Tensor.* Linear and Multilinear Algebra, 62(3). March 2014.

[15] E. Mengi. *Lecture 10: Computation of Eigenvalues and Eigenvectors.* Spring 2010, Koc University. Retrieved from `http://home.ku.edu.tr/~emengi/teaching/math304_s2010/Lecture10.pdf`, June 4, 2019.

[16] C. van Nieuwenhuizen. *Kamerbrief over selectiviteitsbeleid Schiphol – Lelystad Airport* (January 17, 2019). Retrieved from `https://www.rijksoverheid.nl/documenten/kamerstukken/2019/01/17/selectiviteitsbeleid-schiphol-lelystad-airport`.

[17] Open Flights. *Airport Database.* Retrieved from `https://openflights.org/data.html#airport`, March 17, 2019.

[18] R. Pemantle. *Vertex-reinforced random walk.* Probability Theory and Related Fields, 92(1), pp. 117-136.

[19] L. Qi. *Eigenvalues of a real supersymmetric tensor.* Journal of Symbolic Computation, 40. 2005.

[20] S.M. Ross. *Introduction to Probability Models* (Eleventh Edition, 2014). Amsterdam: Elsevier.

[21] M. Rosvall, A.V. Esquivel, A. Lancichinetti, J.D. West, R. Lambiotte. *Memory in network flows and its effects on spreading dynamics and community detection.* Nature Communications volume 5, 4630. 2014.

[22] Schiphol 2019. *Leuke feitjes in een factsheet.* Retrieved from `https://www.schiphol.nl/nl/jij-en-schiphol/pagina/wist-je-dat-factsheets/`, May 23, 2019.

[23] S.E. Shreve. *Stochastic Calculus for Finance II: Continuous-Time Models.* Springer Science + Business Media, New York 2004.

[24] F. Tudisco, F. Arrigo, A. Gautier. *Node and Layer Eigenvector Centralities for Multiplex Networks.* 22 November 2017. Retrieved from `https://arxiv.org/abs/1711.08448`.

Appendix A

Matlab Code

Here is an overview of all the code we used for this thesis. We used Matlab R2018b, Update 2, 64-bit.

```
% Make a transition matrix for a first-order Markov chain

function s = firstMC(i,j,k)
% Function of i and j, if there is a flight from i to j, and k, the number of
    passengers

N = max([max(i),max(j)]); s = sparse(i,j,k,N,N); c = sum(s,1);

for j = 1:N % Normalize to a column-stochastic matrix
        if c(j) > 0
                s(:,j) = s(:,j)./c(j);
        end
end
```

```
% Compute the stationary distribution for a first-order Markov chain

function x = stat1MC(P, tol)
% Function of P, the transition matrix, and tol, the tolerance level of the
    power method

[N,N] = size(P); % Determine the number of states
e = ones(N,1); x = e/N; oldx = zeros(N,1);

while norm(x-oldx) > tol
        oldx = x;
        x = P*x;
        x = x/sum(x)
end
```

```
% Simulate a first-order Markov chain with two initial values

function X = simulate2(P,numSteps,X0,X1)
% Function of P, the transition matrix of the first-order Markov chain,
    numsteps, the number of steps in a simulation, and X0 and X1, the two
    initial values

```

```
[N, N] = size(P); % Determine the number of states
X = zeros(1+numSteps,1);
simState = X0; simState2 = X1; % Set initial values
X(1,1) = simState; X(2,1) = simState2;

for i = 3:(1+numSteps) % Simulate states during the number of steps
        u = rand;
        cs = cumsum(P(:,simState2));
        if cs(N,1) == 0
                X(i,1) = simState2;
        else
                simState2 = find(u < cumsum(P(:,simState2)),1);
                X(i,1) = simState2;
        end
end
```

```
% Construct a transition tensor for a spacey random walk

function p = SRW(P, numSteps)
% Function of P, the transition matrix of the first-order Markov chain and
    numSteps, the number of steps in a simulation

[N,N] = size(P); % Determine the number of states
X0 = repelem(1:N, N); X1 = [1:N]; X1 = repmat(X1, 1, N); % Generate initial
    values

for i = 1:N^2 % Simulate for all possible initial values
        X = simulate2(P, numSteps, X0(i), X1(i)); mat(i,:) = X;
end

% Count the number of occurances of a state in the simulation
for i = 1:N^2
        w(i,:) = histcounts(mat(i,:), N,'Normalization','probability');
end
w = reshape(w',[],1);

x0 = ones(N^3,1)/N^3; x1 = zeros(N^3,1); x2 = ones(N^3, 1);

% Solve minimization problem
x = fminsearchbnd(@(p) -log(sum(w.*p)),x0,x1,x2);

% Reshape to a tensor from the vectorization
p = reshape(x, [N, N, N]); sp = sum(p,1);

% Normalize to a column-stochastic tensor
for k = 1:3
        for j = 1:3
                p(:,j,k) = p(:,j,k)./sp(1,j,k);
        end
end
```

```
% Multiplication for a third-order tensor
function a = tensormltp(P,x)

[ni,nj,nk] = size(P);
for i = 1:ni
        for j = 1:nj
                for k = 1:nk
                        a(i) = P(i,j,k)*x(j)*x(k);
                end
        end
end
```

```
% Compute the stationary distributions of a spacey random walk

function x = statSRW(P, tol)
% Function of P, the transition tensor and tol, the tolerance level of the
    power method

[N,N,N] = size(P); % Determine the number of states
e = ones(N,1); x = e/N; oldx = zeros(N,1);

while norm(x-oldx) > tol
        oldx = x;
        x = (tensormltp(P,x)-x)/sum(tensormltp(P,x)-x);
end
```

Appendix B

Lelystad Airport

In this appendix, you can find all the destinations that airplanes from Lelystad Airport will fly to. We chose these destinations ourselves, since they were not known at the time of the simulations. The destinations are mainly around the Mediterranean. The numbers of passengers are adapted to the number of passengers to and from the airport in 2017.

ICAO	**Name of Airport**	**City**	**Country**
LEAL	Alicante International Airport	Alicante	Spain
GCRR	Lanzarote Airport	Arrecife	Spain
LEBL	Barcelona International Airport	Barcelona	Spain
EDDT	Berlin-Tegel Airport	Berlin	Germany
LEBB	Bilbao Airport	Bilbao	Spain
LFBD	Bordeaux-Mérignac Airport	Bordeaux	France
LHBP	Budapest Liszt Ferenc International Airport	Budapest	Hungary
LIRQ	Peretola Airport	Florence	Italy
GCFV	Fuerteventura Airport	Fuerteventura	Spain
LEGE	Girona Airport	Gerona	Spain
GCLP	Gran Canaria Airport	Gran Canaria	Spain
LEIB	Ibiza Airport	Ibiza	Spain
LGKO	Kos Airport	Kos	Greece
EGLL	London Heathrow Airport	London	United Kingdom
LFLL	Lyon Saint-Exupéry Airport	Lyon	France
LEMD	Adolfo Suárez Madrid–Barajas Airport	Madrid	Spain
LEMG	Málaga Airport	Malaga	Spain
LFML	Marseille Provence Airport	Marseille	France
LIML	Milano Linate Airport	Milan	Italy
LFRS	Nantes Atlantique Airport	Nantes	France
LIRN	Naples International Airport	Naples	Italy
LFMN	Nice-Côte d'Azur Airport	Nice	France
LFPG	Charles de Gaulle International Airport	Paris	France
LIRP	Pisa International Airport	Pisa	Italy
LIRF	Leonardo da Vinci–Fiumicino Airport	Rome	Italy
LEZL	Sevilla Airport	Sevilla	Spain
GCTS	Tenerife South Airport	Tenerife	Spain
LGTS	Thessaloniki Macedonia International Airport	Thessaloniki	Greece
LFBO	Toulouse-Blagnac Airport	Toulouse	France
LEVC	Valencia Airport	Valencia	Spain
LIPZ	Venice Marco Polo Airport	Venice	Italy
LGZA	Zakynthos International Airport "Dionysios Solomos"	Zakynthos	Greece

Table B.1: Destinations for flights that will be transferred from Schiphol to Lelystad, based on my own estimations.