Utrecht University

Rijksinstituut voor Volksgezondheid en Milieu
Ministerie van Volksgezondheid, Welzijn en Sport

Forecasting SARS-CoV-2 Virus Load in Sewage

Using Autoregression Models for Time Series Data

Master Thesis Mathematical Sciences

Fleur Slegers (5605407)

First supervisor	Dr. I. Kryven
Second supervisor	Dr. C. Spitoni
Daily supervisor	Dr. W. A. Hetebrij

March 17, 2023

Abstract

Wastewater is a reservoir of human excretion and contains virus particles shed by people. Its analysis can provide information on the prevalence of infectious diseases. In the Netherlands, sewage surveillance has been used as a tool for monitoring the COVID-19 pandemic by detecting and measuring SARS-CoV-2 virus particles in sewage at over 300 sewage treatment plants (STPs) multiple times a week. The result thereof is an extensive data set of multivariate time series. In this thesis, linear regression models are used to model and forecast virus load time series for each variable (STP). We compare vector autoregressive (VAR) models which were enriched with different variable selection methods, based on K-Nearest Neighbours, correlations between time series and principal component analysis. How much the inclusion of multiple variables improves predictions is strongly dependent on the number and choice of variables, on the smoothness of the time series, on the length of the training set and on time between training and testing. A remarkable result from this research is that intermediate number of variables used in the models resulted in largest test errors. We found that performance of the models was worse for STPs with small catchment areas. With this research, we shed light on how relationships between STPs can be incorporated in multivariate linear time series models and introduce three novel variable selection methods for VAR.

Contents

1 Introduction **1**

2 Time Series and Hilbert Spaces **4**
2.1 Univariate time series 4
2.2 Multivariate time series 6
2.3 Hilbert Spaces and linear predictions 6

3 Linear Regression **17**
3.1 Singular Value Decomposition 18
3.2 Principal Component Analysis 22

4 Models **27**
4.1 Autoregressive model (AR) 27
4.2 Vector autoregressive model (VAR) 28
4.2.1 VAR with principal variables (PCA) 28
4.2.2 VAR with K-Nearest Neighbours (KNN) 30
4.2.3 VAR with correlation-based neighbours (CORR) 31
4.3 Baseline model 32

5 Data analysis **33**
5.1 Training, validating and testing 33
5.2 Preprocessing 34

6 Application: SARS-CoV-2 virus loads in sewage **35**
6.1 Preprocessing steps 35
6.2 Additional constraints 36

7 Results **37**
7.1 7-step forecasts 37
7.2 Inspection of the validation errors 45
7.3 Predictive power per STP 46

8 Discussion **56**

Appendices **61**

A Implementation **62**

Chapter 1

Introduction

Wastewater-based epidemiology (WBE) is a relatively new and popular method within the field of infectious disease surveillance. As wastewater is a reservoir of human excretion, it contains endogenous biological compounds such as virus particles, and so its analysis can provide information on the spread of infectious diseases [19]. WBE has been used successfully for creating local illicit drug consumption profiles [33] and containing a polio outbreak by timely detection of poliovirus in Israeli sewage [6].

Recently, WBE has also been used for monitoring of the COVID-19 pandemic. Since March 2020, The National Institute for Public Health and the Environment (RIVM) has been collecting wastewater samples three to four times a week from sewage treatment plants (STPs) in the Netherlands. Currently, 317 STPs are sampled. Each STP serves a geographically well-defined population, together covering 99.7% of Dutch households. In figure (1.1), a map of the sampled STPs is given. STPs are distributed fairly evenly over the Netherlands, with larger STPs and a higher density of STPs in the west of the Netherlands. The number of inhabitants per STP ranges from 947 to $827,802$. SARS-CoV-2 virus particles are present in the faeces of about half of infected people, and flow with sewage to an STP. At the STPs, 24-hour samples of sewage are taken and sent to the RIVM, where genetic material (RNA) is extracted from the samples. Then, a PCR test is conducted to detect SARS-CoV-2 virus particles. The number of virus particles are corrected for daily sewage volume and normalized for number of inhabitants in the catchment area of the STP. The result is an extensive data set of time series of virus loads per 100,000 inhabitants for each STP. These time series contain information on the spatial and temporal trends of the disease [18].

Besides sewage surveillance, COVID-19 is additionally monitored through: clinical testing at public health services (GGDs) and general practitioner practices, hospital and ICU admissions, and deaths caused by COVID-19. Advantages of sewage surveillance with respect to these tools are its anonymity and its ability to measure virus particles in non-symptomatic cases. The method also does not rely on testing policy or preparedness of patients to get tested. It provides information on almost the whole population [18]. As infected individuals typically excrete virus particles before showing symptoms, sewage prevalence of SARS-CoV-2 precedes positive tests and hospital admissions, and could serve as an early-warning system for disease outbreaks [19].

WBE as a monitoring tool is not only used in the Netherlands. In 2021, at least 55 countries and 200 universities were conducting a wastewater monitoring program for SARS-CoV-2. Besides the

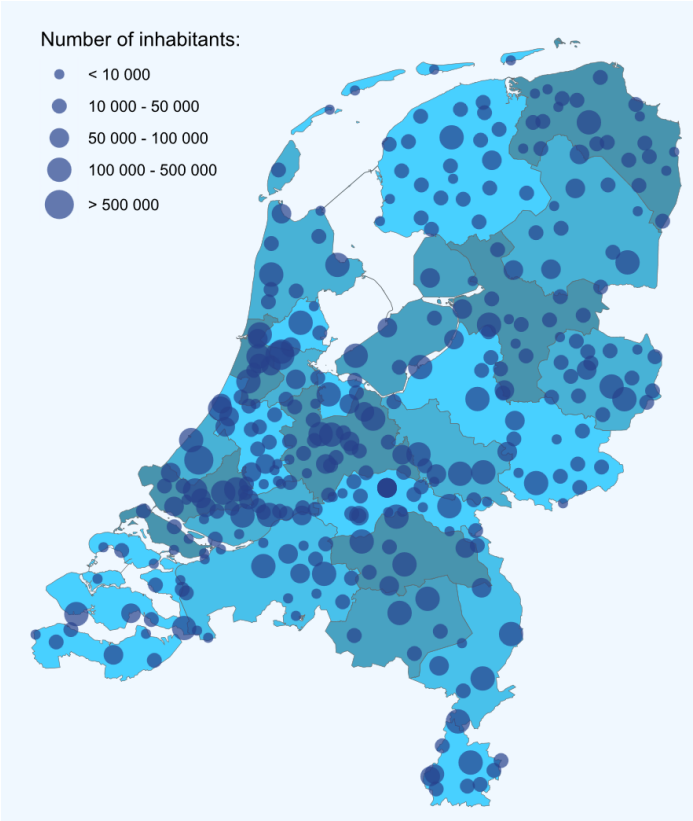


Figure 1.1: Map of sampled sewage treatment plants in the Netherlands (blue dots). The size of each dot represents the number of inhabitants in the catchment area of the STP.

Netherlands, also Finland, France, Hungary, Luxembourg, Spain, and Turkey have nationalized programs [16]. Much research has been done into the relationship between SARS-CoV-2 virus loads in sewage and COVID-19 case numbers at the level of catchment areas of STPs [22]. However, to our knowledge, the prediction of virus loads at the level of STPs, utilizing the extensive network of STPs, is still an underexplored problem. The spatial samples potentially contain information on where disease hot spots occur and on patterns of disease spread [11]. For example, spatial analysis has brought to light geographical patterns in the distributions of two other viral diseases [15, 20].

Wastewater surveillance is most useful when data can be interpreted in real time. Unfortunately, there is a delay of approximately three days between the sampling of STPs and publication of the measurements. To facilitate early warnings at the level of STP catchment areas, predictive models should be developed. Compartmental disease transmission dynamic models are popular tools for modelling and forecasting infectious diseases [29]. An example of such a model is the SIR (Susceptible, Infectious, Recovered) model, in which dynamics between compartments S, I and R are based described by a system of differential equations. A study by Rahimini, Chen and Gandomi (2021) found that the SIR model was one of the models that was most often applied to estimate the COVID-19 outbreak [17]. However, to be able to use the SIR model on wastewater data, virus loads need to be translated to number of infected people, which has proven to be difficult [2].

Within econometric modeling, vector autoregressive models are often used for time series forecasting. Sparse vector autoregressive models, in which many components are assumed to be zero or negligibly small, have been proven to be effective in large-scale econometric applications [9]. Autoregressive time series models are also suited for infectious disease modelling. For example, multivariate time-series autoregression models were used to predict daily and weekly new COVID-19 cases for different age groups [29]. Also for the wastewater data, multivariate time series models are well suited. With these models, the virus load at a given STP can be written as a linear combination of previous virus loads of (a subset of) all STPs. The incorporation of multiple sampling locations in these methods has shown to be beneficial in predicting new COVID-19 cases [13].

In this thesis, linear regression models are implemented to predict virus loads at the level of STPs. This research focuses on detecting and incorporating dependencies and correlations between STPs to improve forecasting performance. We focus on relatively short forecasting horizons (7 days). The main research question is whether the predictions of virus loads at a given STP, acquired using linear models, can be improved by incorporating historical data of other STPs. We want to detect and investigate (possible latent) relationships between time series of STPs. Can we identify STPs whose (lagged) virus loads are strong predictors of future virus loads at other STPs, and can we identify STPs for which the opposite is true? The ultimate goal is to create a multivariate linear regression model that has an underlying network structure with which virus loads at each STP can be predicted. Ideally, we would like to find a network structure such that a subset of STPs predicts all STPs.

This thesis is structured as follows. In chapter (2), we discuss the required mathematical background on time series analysis. In chapter (3), we explain the method of linear regression, which is used to determine model coefficients for linear models. In chapter (4), the models that are implemented are introduced. Then, in chapter (5), data analysis procedures are explained to show how historical data can be used to train models and choose optimal parameters. In chapter (6), we discuss the WBE data set that is used in this thesis and the data preprocessing steps that are performed to make the data set suitable for the time series models. In chapter (7), we show the main results, which are further discussed in chapter (8).

Chapter 2

Time Series and Hilbert Spaces

In this chapter, we discuss the basic concepts of time series analysis on which the models in chapter (4) are based. We only consider real-valued time series, although many notions extend to complex-valued time series. This chapter is mainly based on lecture notes by A.W. van der Vaart [23] and books on time series analysis and forecasting by P. J. Brockwell and R. A. Davis [4, 5].

2.1 Univariate time series

Observations of a variable at different points in time gives us information about how this variable evolves. The set of these observations is called a time series.

Definition 2.1: Discrete Time Series [30, Chapter 1]

A **discrete time series** $\{x_t\}$ is a set of observations x_t of random variables X_t, recorded at observation times t. The observation times $t = 1, 2, \ldots$ form a discrete and linearly ordered set.

An observation x_t is a realization of a random variable X_t and the time series $\{x_t\}$ is a realization of an underlying stochastic process $\{X_t\}$ [30]. We assume that the state space S of $\{X_t\}$ is $\mathbb{R}$. When analyzing the time series, we are interested in the joint distributions of the random vectors $(X_1, X_2, \ldots, X_n)^T$ for $n = 1, 2, \ldots$. Specifying these distributions is equivalent to specifying for all $n = 1, 2, \ldots$ and $-\infty < x_1, \ldots, x_n < \infty$ the following probability:

$$\mathbb{P}\left(X_1 \leq x_1, \ldots, X_n \leq x_n\right).$$

In practice, we often do not know these probabilities and the estimation from available data is complex and often involves too many parameters [4].

Time series models for observed data $\{x_t\}$ are models that predict future values $x_t, x_{t+1}, x_{t+2}, \ldots$ based on historical observations $\{x_1, x_2, \ldots, x_{t-1}\}$ of the time series. The models for this purpose are of the form

$$\hat{x}_t := g\left(x_{t-1}, x_{t-2}, \ldots, x_1\right),$$

where $\hat{x}_t$ denotes the predicted value at time t. These models try to specify the function $g(\cdot)$ based on available observations and an optimization function.

In order to be able to learn from observations and make informed predictions, the time series needs to possess at least some structure. Many time series models assume weak stationarity. A time series is called *weakly stationary* if $\mathbb{E}[X_t]$ and $\mathbb{E}[X_{t+h}X_t]$ exist and do not change over time [23]. The first and second moments are completely determined by its mean, the auto-covariance and auto-correlation functions. The *auto-covariance* $\gamma_X(h)$ of a weakly stationary time series at lag $h \in \mathbb{Z}$ is defined as

$$\gamma_X(h) = \operatorname{cov}(X_{t+h}, X_t).$$

The *auto-correlation* $\rho_X(h)$ of a weakly stationary time series at lag h is

$$\rho_X(h) = \rho_X(X_{t+h}, X_t) = \frac{\gamma_X(h)}{\gamma_X(0)}$$

[23]. The auto-covariance and auto-correlation are measures of linear dependencies between observations of a single variable. An example of a univariate time series is the Gaussian white noise series.

Example 2.1: Gaussian white noise series [5]

A process $\{Z_t\}$ is a **Gaussian white noise series** with mean 0 and variance σ^2, $\left(\{Z_t\} \sim \mathrm{WN}\left(0, \sigma^2\right)\right)$, if and only if $\{Z_t\}$ has zero mean and its covariance function $\gamma_Z(\cdot)$ is given by

$$\gamma_Z(h) = \begin{cases} \sigma^2 & \text{if } h = 0, \\ 0 & \text{if } h \neq 0. \end{cases}$$

In general, when working with time series data, a number of preprocessing steps might be needed to be able to implement certain models. For instance, time series data is often not weakly stationary in practice. Variables can steadily increase or decrease over a long period of time. This is called a *trend* [23]. A trend causes the mean to change over time, thus resulting in a non-stationary time series. *Seasonality* is when a variable shows a cyclic pattern in observation values [23]. If a time series contains seasonality, the variance changes over time, resulting in non-stationarity. Trends and seasonality can be detected by visually inspecting the time series or by inspecting the mean and auto-covariance function over time. Before applying the models to the data, trends and seasonality should be removed and can later be added to the model. There are various techniques to transform a time series into a weakly stationary one. Perhaps the easiest one are the *difference filter*

$$\nabla X_t = X_t - X_{t-1}$$

and the *seasonal difference filter*

$$\nabla_k X_t = X_t - X_{t-k},$$

where k is a positive integer, representing the period of the season. These filters can be applied multiple times to remove polynomial trends [4].

In the next sections, we will encounter time series that are shifted in time. To improve readability, we will make use of the *lag operator* (or backward shift operator), which is defined as follows:

Definition 2.2: Lag Operator [4]

For a time series $\{x_t\} = (x_1, x_2, \dots x_T)$, the **lag operator** L is the operator such that

$$L^k x_t = x_{t-k} \quad \text{for all } t > k \text{ and all } k = 1, 2, \dots.$$

2.2 Multivariate time series

When at each observation time multiple variables are observed, we refer to the series as a *multivariate time series*. Multivariate time series are of interest when the observed variables are assumed to be dependent.

Definition 2.3: Multivariate Time Series

A ***n*-dimensional multivariate time series** $\{\boldsymbol{x}_t\}$ is a set of observations $\boldsymbol{x}_t = (x_{1,t}, x_{2,t}, \dots, x_{n,t})^T$ of random variables $\boldsymbol{X}_t = (X_{1,t}, X_{2,t}, \dots, X_{n,t})^T$ recorded at observation times $t = 1, 2, \dots$.

A multivariate time series is called *weakly stationary* if the vector of means $(\mathbb{E}X_{1,t}, \dots, \mathbb{E}X_{n,t})^T$ and the covariance matrices of $X_{1,t}, \dots, X_{n,t}$ are time-invariant. The *auto-covariance* at lag h is the matrix

$$\gamma_{\boldsymbol{X}}(h) = \left(\operatorname{cov}\left(X_{i,t+h}, X_{j,t}\right)\right)_{i,j=1,\dots,n}$$

and the *auto-correlation* at lag h is

$$\rho_{\boldsymbol{X}}(h) = \left(\rho\left(X_{i,t+h}, X_{j,t}\right)\right)_{i,j=1,\dots,n}$$

[23].

2.3 Hilbert Spaces and linear predictions

Assuming weak stationarity of a variable, in particular a finite second moment, allows us to use Hilbert Space theory to solve the problem of finding the best predictor of $\boldsymbol{x}_t$ based on observations $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{t-1}$. By formulating the prediction problem in terms of Hilbert Spaces, we can use concepts from Euclidean geometry that are very intuitive in two- or three-dimensional space, for instance orthogonality and orthogonal projections.

We restrict ourselves to linear predictions and univariate time series. The *linear prediction problem* is defined as the problem of finding the linear combination (or model) $\varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \dots + \varphi_p x_{t-p}$ that minimizes the *squared error*

$$\sum_{i=p+1}^{m} \left(x_i - \varphi_1 x_{i-1} + \varphi_2 x_{i-2} + \dots + \varphi_p x_{i-p}\right)^2,$$

where p is a parameter that is called the *order* of the model and m is the number of observations [5].

For general Hilbert spaces, we can use the projection theorem to find a linear model that minimizes the squared prediction error for a stationary process. Before discussing the projection theorem, we give the requisite definitions and theorems from Hilbert space theory.

Definition 2.4: Real Vector Space [21]

A **real vector space** is a set $\mathscr{H}$ together with an addition operation $+ : \mathbb{H} \times \mathbb{H} \mapsto \mathbb{H}$ and a multiplication operation $\cdot : \mathbb{R} \times \mathbb{H} \mapsto \mathbb{H}$ that satisfy for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathscr{H}$ and all $\alpha, \beta \in \mathbb{R}$ the following conditions:

(a) $\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{y} + \boldsymbol{x}$,

(b) $(\boldsymbol{x} + \boldsymbol{y}) + \boldsymbol{z} = \boldsymbol{x} + (\boldsymbol{y} + \boldsymbol{z})$,

(c) There exists a vector $\mathbf{0} \in \mathscr{H}$ such that if $\boldsymbol{x} \in \mathscr{H}$, then $\boldsymbol{x} + \mathbf{0} = \boldsymbol{x}$,

(d) For each vector $\boldsymbol{x} \in \mathscr{H}$, there exists a vector $-\boldsymbol{x} \in \mathscr{H}$ such that $\boldsymbol{x} + (-\boldsymbol{x}) = \mathbf{0}$,

(e) $\alpha(\boldsymbol{x} + \boldsymbol{y}) = \alpha\boldsymbol{x} + \alpha\boldsymbol{y}$,

(f) $(\alpha + \beta)\boldsymbol{x} = \alpha\boldsymbol{x} + \beta\boldsymbol{x}$,

(g) $\alpha(\beta\boldsymbol{x}) = (\alpha\beta)\boldsymbol{x}$,

(h) $1\boldsymbol{x} = \boldsymbol{x}$ and $0\boldsymbol{x} = \mathbf{0}$.

Definition 2.5: Inner-Product Space [5]

A real vector space $\mathscr{H}$ is an **inner-product space** if for all $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathscr{H}$ and all $\alpha \in \mathbb{R}$ there exists a real number $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ such that

(a) $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{y}, \boldsymbol{x} \rangle$,

(b) $\langle \boldsymbol{x} + \boldsymbol{y}, \boldsymbol{z} \rangle = \langle \boldsymbol{x}, \boldsymbol{z} \rangle + \langle \boldsymbol{y} + \boldsymbol{z} \rangle$,

(c) $\langle \alpha\boldsymbol{x}, \boldsymbol{y} \rangle = \alpha \langle \boldsymbol{x}, \boldsymbol{y} \rangle$,

(d) $\langle \boldsymbol{x}, \boldsymbol{x} \rangle \geq 0$,

(e) $\langle \boldsymbol{x}, \boldsymbol{x} \rangle = 0$ if and only if $\boldsymbol{x} = \mathbf{0}$.

$\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ is called the **inner product** of $\boldsymbol{x}$ and $\boldsymbol{y}$.

For an element $\boldsymbol{x}$ of an inner-product space, we define its *norm* to be

$$\|\boldsymbol{x}\| = \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}.$$

For inner-product spaces, the *parallelogram law* and the *Cauchy-Schwarz inequality* hold.

Lemma 2.1: (Parallelogram law).

If $\mathscr{H}$ is an inner-product space, then for all $\boldsymbol{x}, \boldsymbol{y} \in \mathscr{H}$

$$\|\boldsymbol{x}+\boldsymbol{y}\|^2 + \|\boldsymbol{x}-\boldsymbol{y}\|^2 = 2\|\boldsymbol{x}\|^2 + 2\|\boldsymbol{y}\|^2. \tag{2.1}$$

Proof. The equation follows from repeatedly applying conditions (a) and (b) from definition (2.3.2). □

Lemma 2.2: (Cauchy-Schwarz inequality).

If $\mathscr{H}$ is an inner-product space, then the *Cauchy-Schwarz inequality* holds:

$$|\langle \boldsymbol{x}, \boldsymbol{y}\rangle| \leq \|\boldsymbol{x}\|\|\boldsymbol{y}\| \quad \text{for all } \boldsymbol{x}, \boldsymbol{y} \in \mathscr{H}$$

and

$$|\langle \boldsymbol{x}, \boldsymbol{y}\rangle| = \|\boldsymbol{x}\|\|\boldsymbol{y}\| \quad \text{if and only if } \boldsymbol{x} = \frac{\boldsymbol{y}\langle \boldsymbol{x}, \boldsymbol{y}\rangle}{\langle \boldsymbol{y}, \boldsymbol{y}\rangle}.$$

Proof. Let $\boldsymbol{x}, \boldsymbol{y}$ be elements of an inner-product space $\mathscr{H}$. We define the function $p : \mathbb{R} \mapsto \mathbb{R}$ as

$$p(t) = \langle t\boldsymbol{x} + \boldsymbol{y}, t\boldsymbol{x} + \boldsymbol{y}\rangle.$$

Using the conditions of an inner product space, we can rewrite p as

$$\begin{aligned} p(t) &= \langle t\boldsymbol{x} + \boldsymbol{y}, t\boldsymbol{x} + \boldsymbol{y}\rangle \\ &= \langle t\boldsymbol{x}, t\boldsymbol{x}\rangle + \langle t\boldsymbol{x}, \boldsymbol{y}\rangle + \langle \boldsymbol{y}, t\boldsymbol{x}\rangle + \langle \boldsymbol{y}, \boldsymbol{y}\rangle \\ &= t^2\langle \boldsymbol{x}, \boldsymbol{x}\rangle + 2t\langle \boldsymbol{x}, \boldsymbol{y}\rangle + \langle \boldsymbol{y}, \boldsymbol{y}\rangle \\ &= t^2\|\boldsymbol{x}\|^2 + 2t\langle \boldsymbol{x}, \boldsymbol{y}\rangle + \|\boldsymbol{y}\|^2, \end{aligned}$$

from which follows that if $\|\boldsymbol{x}\|^2 \neq 0$, then p is a polynomial of degree 2 in t. If $\|\boldsymbol{x}\|^2 = 0$, the Cauchy-inequality is trivial. Assume that $\|\boldsymbol{x}\|^2 > 0$, then p is a parabola opening upward, with its minimum at $t_{\min} = -\langle \boldsymbol{x}, \boldsymbol{y}\rangle\|\boldsymbol{x}\|^{-2}$. By condition (d) of definition (2.3), we know that $p(t) \geq 0$ for all $t \in \mathbb{R}$, so in particular, we have that

$$p\left(t_{\min}\right) = \frac{-\langle \boldsymbol{x}, \boldsymbol{y}\rangle^2 + \|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2}{\|\boldsymbol{x}\|^2} \geq 0,$$

from which follows that $\langle \boldsymbol{x}, \boldsymbol{y}\rangle^2 \leq \|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2$, and thus $|\langle \boldsymbol{x}, \boldsymbol{y}\rangle| \leq \|\boldsymbol{x}\|\|\boldsymbol{y}\|$.

Note that

$$p\left(t_{\min}\right) = 0 \iff \langle \boldsymbol{x}, \boldsymbol{y}\rangle^2 = \|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2.$$

By applying condition (e) of definition (2.5) to

$$p\left(t_{\min}\right) = \langle t_{\min}\boldsymbol{x} + \boldsymbol{y}, t_{\min}\boldsymbol{x} + \boldsymbol{y}\rangle = \left\langle -\langle \boldsymbol{x}, \boldsymbol{y}\rangle\|\boldsymbol{x}\|^{-2}\boldsymbol{x} + \boldsymbol{y}, -\langle \boldsymbol{x}, \boldsymbol{y}\rangle\|\boldsymbol{x}\|^{-2}\boldsymbol{x} + \boldsymbol{y}\right\rangle,$$

we see that

$$|\langle \boldsymbol{x}, \boldsymbol{y} \rangle| = \|\boldsymbol{x}\|\|\boldsymbol{y}\| \quad \text{if and only if } \boldsymbol{x} = \frac{\boldsymbol{y}\|\boldsymbol{x}\|^2}{\langle \boldsymbol{x}, \boldsymbol{y} \rangle} = \frac{\boldsymbol{y}\langle \boldsymbol{x}, \boldsymbol{y} \rangle}{\langle \boldsymbol{y}, \boldsymbol{y} \rangle}.$$

□

We will make use of the following proposition. For its proof, we refer to Chapter 2 of [5].

Proposition 2.1: (Continuity of the Inner Product).

Let $\mathscr{H}$ be an inner-product space. If $(\boldsymbol{x}_n)$ and $(\boldsymbol{y}_n)$ are sequences of elements in $\mathscr{H}$ such that $\|\boldsymbol{x}_n - \boldsymbol{x}\| \to 0$ and $\|\boldsymbol{y}_n - \boldsymbol{y}\| \to 0$ as $n \to \infty$, with $\boldsymbol{x}, \boldsymbol{y} \in \mathscr{H}$, then, as $n \to \infty$, $\|\boldsymbol{x}_n\| \to \|\boldsymbol{x}\|$ and $\langle \boldsymbol{x}_n, \boldsymbol{y}_n \rangle \to \langle \boldsymbol{x}, \boldsymbol{y} \rangle$.

Definition 2.6: Cauchy Sequence [5]

A sequence $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ of elements of an inner-product space $\mathscr{H}$ is a **Cauchy sequence** if

$$\|\boldsymbol{x}_n - \boldsymbol{x}_m\| \to 0 \text{ as } m, n \to \infty,$$

which is the case if for every $\epsilon > 0$ there exists a positive integer $N(\epsilon)$ such that

$$\|\boldsymbol{x}_n - \boldsymbol{x}_m\| < \epsilon \text{ for all } m, n > N(\epsilon).$$

An inner-product space $\mathscr{H}$ is called *complete* if every Cauchy sequence converges in norm to an element $x \in \mathscr{H}$. We are now ready to define *Hilbert Spaces*.

Definition 2.7: Hilbert Space [5]

An inner-product space which is complete is called a **Hilbert Space** $\mathscr{H}$.

Perhaps the easiest example of a real Hilbert Space is Euclidean Space. In the next example, we will show that Euclidean space satisfies the conditions of a Hilbert Space.

Example 2.2: (Euclidean Space).

If we consider the space consisting of all column vectors $\boldsymbol{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ for some $n \in \mathbb{N}$, and we define its inner product as

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{i=1}^{n} x_i y_i,$$

then this space (Euclidean space) is a real inner-product space. The norm of a vector in $\mathbb{R}^n$ is its length. Suppose we have a Cauchy sequence $(\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots)$ with $\boldsymbol{x_i} \in \mathbb{R}^n$ for $i = 1, 2, \ldots$. Then

$$\|\boldsymbol{x_n} - \boldsymbol{x_m}\| \to 0 \text{ as } m, n \to \infty. \tag{2.2}$$

As

$$\|\boldsymbol{x_n} - \boldsymbol{x_m}\|^2 = \sum_{i=1}^{n} |(x_n)_i - (x_m)_i|^2,$$

equation (2.2) holds if and only if

$$|(x_n)_i - (x_m)_i| \to 0 \text{ as } m, n \to \infty \text{ for all } i = 1, \ldots, n. \tag{2.3}$$

We can interpret each equation in (2.3) as an individual Cauchy sequence in $\mathbb{R}$. As $\mathbb{R}$ is a complete metric space, every Cauchy sequence in $\mathbb{R}$ has a limit that also lies in $\mathbb{R}$. So, for $i = 1, \ldots, n$, there exists some $x_i \in \mathbb{R}$ such that $|x_{ni} - x_i| \to 0$ as $n \to \infty$. As a result, there exists a $\boldsymbol{x} \in \mathbb{R}^n$ such that

$$|\boldsymbol{x_n} - \boldsymbol{x}| \to 0 \text{ as } n \to \infty.$$

It follows that the Euclidean space is complete, and thus a Hilbert space.

For a probability space $(\Omega, \mathscr{F}, P)$, the space $L_2(\Omega, \mathscr{F}, P)$ is exactly the set of all real random variables X defined on Ω with finite second moment $\mathbb{E}\left[X^2\right]$ [23], where the second moment of a variable X is given by

$$\mathbb{E}\left[X^2\right] = \int_{\Omega} X^2 \mathrm{d}P = \int_{\mathbb{R}} x^2 p(x) \mathrm{d}x, \tag{2.4}$$

with p the probability density function of X. In the next example, we will see that $L_2(\Omega, \mathscr{F}, P)$, together with specific equivalence classes, is also a Hilbert space. To do this, we will make use of the following proposition:

Proposition 2.2

If x_n is a sequence in $L_2(\Omega, \mathscr{F}, P)$ and $\|x_{n+1} - x_n\| \leq 2^{-n}$ for $n = 1, 2, \ldots$, then there exists a random variable X on $(\Omega, \mathscr{F}, P)$ such that $x_n \to X$ with probability one. [5]

Proof. Suppose that $X_0 = 0$. Then $X_n = \sum_{j=1}^{n} (X_j - X_{j-1}) \leq \sum_{j=1}^{n} |X_j - X_{j-1}|$. By linearity of

expectation, it holds that

$$\mathbb{E}\sum_{j=1}^{\infty}|X_j - X_{j-1}| = \sum_{j=1}^{\infty}\mathbb{E}\,|X_j - X_{j-1}|\,. \tag{2.5}$$

Note that we can write $\mathbb{E}\,|X_j - X_{j-1}|$ as $\big|\langle X_j - X_{j-1}, 1\rangle\big|$. Applying the Cauchy-Schwarz inequality then gives the following inequality:

$$\mathbb{E}\,|X_j - X_{j-1}| \leq \|X_j - X_{j-1}\|\|1\| = \|X_j - X_{j-1}\|.$$

Filling in above inequality in equation (2.5), together with the assumption that $\|X_{n+1} - X_n\| \leq 2^{-n}$ for $n = 1, 2, \ldots$ gives

$$\mathbb{E}\sum_{j=1}^{\infty}|X_j - X_{j-1}| \leq \sum_{j=1}^{\infty}\|X_j - X_{j-1}\| \leq \|X_1\| + \sum_{j=1}^{\infty}2^{-j} = \|X_1\| + 1 < \infty.$$

From this, we conclude that with probability one, $\sum_{j=1}^{\infty}|X_j - X_{j-1}| < \infty$. So, $\sum_{j=1}^{n}|X_j - X_{j-1}|$ is a non-decreasing sequence that is bounded above that converges to its supremum. As a result, $\lim_{n\to\infty} X_n$ exists and is finite with probability one. Note that for $X_0 \neq 0$ the proof still holds. □

We will also use *Fatou's lemma*:

Lemma 2.3: Fatou's lemma [14]

For a measure space (Ω, Σ, μ) and a measurable function f on Ω, let $\{f_n\}$ be a sequence of extended real-valued measurable functions which is bound from below by an integrable function. Then

$$\int_{\Omega}\liminf_{n\to\infty} f_n \;\mathrm{d}\mu \leq \liminf_{n\to\infty}\int_{\Omega} f_n \;\mathrm{d}\mu.$$

Example 2.3: ($L_2(\Omega, \mathscr{F}, P)$).

This example is based on Chapter 2 of [5]. It is easy to see that $L_2(\Omega, \mathscr{F}, P)$ is a vector space. If we define the following inner product on $L_2(\Omega, \mathscr{F}, P)$:

$$\langle X, Y \rangle = \mathbb{E}\left[XY\right] = \int_{\mathbb{R}} \int_{\mathbb{R}} xy f_{X,Y}(x, y) \, \mathrm{d}x \, \mathrm{d}y,$$

where $f_{X,Y}$ is the joint probability distribution of X and Y, then $L_2(\Omega, \mathscr{F}, P)$ is also an inner-product space. We will now show that above inner product indeed satisfies the conditions in (2.5). Conditions (a) and (b) follow from L_2 being a vector space, and condition (d) is trivial. For condition (c), we have that

$$\langle \alpha X, Y \rangle = \int_{\mathbb{R}} \int_{\mathbb{R}} \alpha \; xy f_{X,Y}(x, y) \, \mathrm{d}x \, \mathrm{d}y = \alpha \int_{\mathbb{R}} \int_{\mathbb{R}} xy f_{X,Y}(x, y) \, \mathrm{d}x \, \mathrm{d}y = \alpha \langle X, Y \rangle.$$

From $X = 0$, it follows that $\mathbb{E}[X^2] = 0$. Then, we only have to show that if $\langle X, X \rangle = 0$, then $X = 0$. Assume that $\langle X, X \rangle = 0$. Then

$$\mathbb{E}\left[X^2\right] = \mathbb{E}\left[(x - \mathbb{E}[X])^2\right] + (\mathbb{E}[X])^2 = 0.$$

As both terms in the middle of this expression are always positive, it follows that $\mathbb{E}\left[(x - \mathbb{E}[X])^2\right] = 0$ and $\mathbb{E}[X] = 0$. So, it holds that $\mathbb{P}(X = 0) = 1$. By introducing equivalence classes to $L_2(\Omega, \mathscr{F}, P)$, we are able to take the last step from $\mathbb{P}(X = 0) = 1$ to $X = 0$. Random variables X and Y are said to be equivalent if $P(X = Y) = 1$.

We will now show that $L_2(\Omega, \mathscr{F}, P)$ is complete. To do this, we use the Cauchy-Schwarz equation, which for the space $L_2(\Omega, \mathscr{F}, P)$ is given by

$$\left|\mathbb{E}[UV]\right| \leq \sqrt{\mathbb{E}\left[U^2\right] \mathbb{E}\left[V^2\right]}.$$

We need to prove that if $\|X_m - X_n\|^2 \to 0$ as $m, n \to \infty$, then there exists $X \in L_2(\Omega, \mathscr{F}, P)$ such that $\|X_n - X\|^2 = \mathbb{E}\left|X_n - X\right|^2 \to 0$ as $n \to \infty$. Hereto we make use of proposition (2.3).

To prove completeness, we need to show that if $\|x_m - x_n\|^2 \to 0$ as $m, n \to \infty$, then there exists an element X in $L_2(\Omega, \mathscr{F}, P)$ such that $\|x_n - X\| \to 0$ as $n \to \infty$. Suppose that $\{x_n\}$ is a Cauchy sequence in $L_2(\Omega, \mathscr{F}, P)$. Then, or all $\epsilon > 0$, there exists an integer $N(\epsilon)$ such that $\|x_n - x_m\| < \epsilon$ for all $m, n > N(\epsilon)$. Choose integers $n_1 < n_2 < \ldots$ such that $\|x_n - x_m\| \leq 2^{-k}$ for all $m, n > n_k$.

By applying proposition (2.3) to the sequence $x_{n_1}, x_{n_2}, \ldots$, it follows that there exists a random variable X such that $x_{n_k} \to X$ as $k \to \infty$ with probability one. Now, we make use of Fatou's lemma (2.3). As

$$\begin{aligned} \|x_n - X\|^2 = \langle x_n - X, x_n - X \rangle = \mathbb{E}\left[(x_n - X)^2\right] &= \int |x_n - X|^2 \; \mathrm{d}P \\ &= \int \liminf_{k \to \infty} |x_n - x_{n_k}|^2 \; \mathrm{d}P, \end{aligned}$$

1: Example 2.3: extended

by Fatou's lemma it holds that

$$\|x_n - X\|^2 \leq \liminf_{k\to\infty} \|x_n - x_{n_k}\|^2.$$

As $\{x_n\}$ is a Cauchy sequence, we can choose n large enough to make the right side arbitrarily small. So, $\|x_n - X\|^2 \to 0$ as $n \to \infty$. By the triangle inequality,

$$\|X\| = \|x_n - x_n + X\| \leq \|x_n - X\| + \|x_n\|,$$

for which the right-hand side is finite for large enough n. Thus, $\mathbb{E}\left[X^2\right] < \infty$, thus $X \in L_2(\Omega, \mathscr{F}, P)$.

We can view observations of a stationary process as elements of a probability space $L_2(\Omega, \mathscr{F}, P)$. To solve the linear prediction problem, we first need to define orthogonality for the space $L_2(\Omega, \mathscr{F}, P)$.

Definition 2.8: Orthogonality [23]

Two elements $f, g \in L_2(\Omega, \mathscr{F}, P)$ are **orthogonal** ($f \perp g$) if $\langle f, g\rangle = 0$. Two subsets F, G are orthogonal ($F \perp G$) if $f \perp g$ for all $f \in F$ and all $g \in G$.

Theorem 2.1: Projection Theorem [23]

Let $L \subset L_2(\Omega, \mathscr{U}, \mu)$ be a closed vector subspace (i.e. a vector subspace that contains all its limit points). For every $f \in L_2(\Omega, \mathscr{U}, \mu)$, there exists a unique element $\Pi f \in L$ such that

$$\Pi f = \min_{l\in L} \|f - l\|^2.$$

This element is uniquely determined by the requirements $\Pi f \in L$ and $f - \Pi f \perp L$. The function Πf is called the **orthogonal projection** of f onto L.

Proof. Let $L \subset L_2(\Omega, \mathscr{U}, \mu)$ be a closed vector subspace and $f \in L_2(\Omega, \mathscr{U}, \mu)$. First, we show that there exists an element $\Pi f \in L$ such that $\Pi f = \min_{l\in L}\|f - l\|^2$. As $0 \in L$ and $\|f\| < \infty$, the minimal distance $d = \inf_{l\in L}\|f - l\|$ of f to L is finite. Let l_n be a sequence in L such that $\|f - l_n\|^2 \to d$. We can rewrite $\|l_m - l_n\|^2$ as follows:

$$\begin{aligned}
\|l_m - l_n\|^2 &= \|l_m - f + f - l_n\|^2 \\
&= \langle l_m - f, l_m - f\rangle + \langle f - l_n, f - l_n\rangle + 2\langle l_m - f, f - l_n\rangle \\
&= 2\|l_m - f\|^2 + 2\|f - l_n\|^2 + \langle l_m - f - (f - l_n), l_m - f - (f - l_n)\rangle \\
&= 2\|l_m - f\|^2 + 2\|f - l_n\|^2 - \|l_m - f - (f - l_n)\|^2 \\
&= 2\|l_m - f\|^2 + 2\|f - l_n\|^2 - \|l_m - f - (f - l_n)\|^2 \\
&= 2\|l_m - f\|^2 + 2\|f - l_n\|^2 - 4\left\|\frac{1}{2}(l_m + l_n) - f\right\|^2.
\end{aligned}$$

Note that $(l_m + l_n)/2 \in L$, and thus $-4\left\|\frac{1}{2}(l_m + l_n) - f\right\|^2 \leq -4d^2$. Also, it holds that $2\|l_m - f\|^2 \to 2d^2$ and $2\|f - l_n\|^2 \to 2d^2$ when $m, n \to \infty$. Thus, $\|l_m - l_n\|^2$ is bounded above by $4d^2 + o(1) - 4d^2$.

Together with the non-negativeness of $\|l_m - l_n\|^2$, it follows that this term converges to zero. By the completeness of $\mathscr{L}_2(\Omega, \mathscr{U}, \mu)$, it follows that l_n is a Cauchy sequence which converges to some element $x \in \mathscr{L}_2(\Omega, \mathscr{U}, \mu)$. As L is closed, we must have that $x \in L$. By the continuity of the norm, we have that $\|f - l\|^2 = \lim_{n\to\infty}\|f - l_n\|^2 \to d$, from which follows that we can choose x as the minimizing element Πf.

We will now show that the element $\Pi f = \min_{l\in L}\|f - l\|^2$ is unique. Suppose that

$$\Pi_1 f = \min_{l\in L}\|f - l\|^2 \quad \text{and} \quad \Pi_2 f = \min_{l\in L}\|f - l\|^2.$$

Then we can take the sequence l_n with $l_i = \Pi_1 f$ for $i = 1, 3, 5, \ldots$ and $l_i = \Pi_2 f$ for $i = 2, 4, 6, \ldots$. This sequence satisfies $\|f - l_n\| \to \infty$ as $n \to \infty$, and so we have already shown that l_n is a Cauchy-sequence and converges to a limit, from which follows that $\Pi_1 f = \Pi_2 f$.

It rests to show the orthogonality relation. Suppose that $\hat{x} \in L$ and $(f - \hat{x}) \perp L$. Let $y \in L$.

$$\begin{aligned}\|f - y\|^2 &= \langle f - \hat{x} + \hat{x} - y, f - \hat{x} + \hat{x} - y\rangle \\ &= \|f - \hat{x}\|^2 + \|\hat{x} - y\|^2 + 2\langle f - \hat{x}, \hat{x} - y\rangle \\ &\geq \|f - \hat{x}\|^2,\end{aligned}$$

with equality if and only if $y = \hat{x}$.

Now suppose that $\hat{x} \in L$ and $(f - \hat{x}) \not\perp L$. Take $\tilde{x} = \hat{x} + ay\|y\|^{-2}$, with y an element in L such that $\langle f - \hat{x}, y\rangle \neq 0$ and $a = \langle f - \hat{x}, y\rangle$. Then

$$\begin{aligned}\|f - \tilde{x}\|^2 &= \langle f - \hat{x} + \hat{x} - \tilde{x}, f - \hat{x} + \hat{x} - \tilde{x}\rangle \\ &= \|f - \hat{x}\|^2 + \|\hat{x} - \tilde{x}\|^2 + 2\langle f - \hat{x}, \hat{x} - \tilde{x}\rangle \\ &= \|f - \hat{x}\|^2 + \|-ay\|y\|^{-2}\|^2 + 2\langle f - \hat{x}, -ay\|y\|^{-2}\rangle \\ &= \|f - \hat{x}\|^2 + \frac{a^2}{\|y\|^2} - 2\frac{a^2}{\|y\|^2} \\ &< \|f - \hat{x}\|^2.\end{aligned}$$

So, if $(f - \hat{x}) \not\perp L$, there exists an element $\tilde{x} \in L$ such that $\|f - \tilde{x}\|^2 < \|f - \hat{x}\|^2$, so $\hat{x} \neq \Pi f$.

□

We can view Πf as the projection of f onto the linear space L, and $f - \Pi f$ is orthogonal to L. In the following example, we will give a geometric interpretation of the Projection Theorem, but first we need to define the root mean squared error.

Definition 2.9: Root Mean Squared Error

The root mean squared error $RMSE$ of predictions $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m$ with actual values $y_1, y_2, \ldots, y_m$ is

$$RMSE = \left(\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2\right)^{1/2}. \tag{2.6}$$

Example 2.4: (Minimum Root Mean Squared Error Linear Prediction of a Stationary Process)

Let $\{X_t, t = 0, 1, \dots\}$ be a stationary process on $(\Omega, \mathscr{F}, P)$ with mean zero and autocovariance function $\gamma_X(\cdot)$. We want to find the linear combination $\hat{X}_n = \sum_{i=1}^{p} \varphi_i X_{n-i}$ that best approximates X_n, so the linear combination that minimizes $\mathbb{E}\,|X_n - \sum_i^p \varphi_i X_{n-i}|$. Hereto we take $\mathscr{H} = L_2(\Omega, \mathscr{F}, P)$.

Note that by minimizing $\|X_n - \hat{X}_n\|^2$, we minimize $\mathbb{E}\,|X_n - \sum_i^p \varphi_i X_{n-i}|$.

Note that X_{n-i}, $i = 0, 1, \dots, p$ are all $(m-p)$-dimensional vectors, where m is the number of observations of the time series. As an example, consider the time series $X_t = (1, 4, 3, 2, 7)$. In this case, we are interested in linear prediction of X_n from it's two previous values. We have $\boldsymbol{X}_n = (3, 2, 7)^T$, $\boldsymbol{X}_{n-1} = (4, 3, 2)^T$ and $\boldsymbol{X}_{n-2} = (1, 4, 3)^T$. These vectors are drawn in figure (2.1).

As the prediction of $\boldsymbol{X}_n$ is a linear combination of $\boldsymbol{X}_{n-1}$ and $\boldsymbol{X}_{n-2}$, it must lie on the plane spanned by these vectors. This plane is depicted in light blue in figure (2.1). We call this plane V. As $\|\boldsymbol{X}_n - \boldsymbol{Y}\|^2$ is the distance from $\boldsymbol{X}_n$ to $\boldsymbol{Y}$, we see that taking $\boldsymbol{Y}$ as the intersection of V and the line through $\boldsymbol{X}_n$ that is perpendicular to V minimizes this distance.

In the next section, we will discuss the linear prediction problem in more detail.

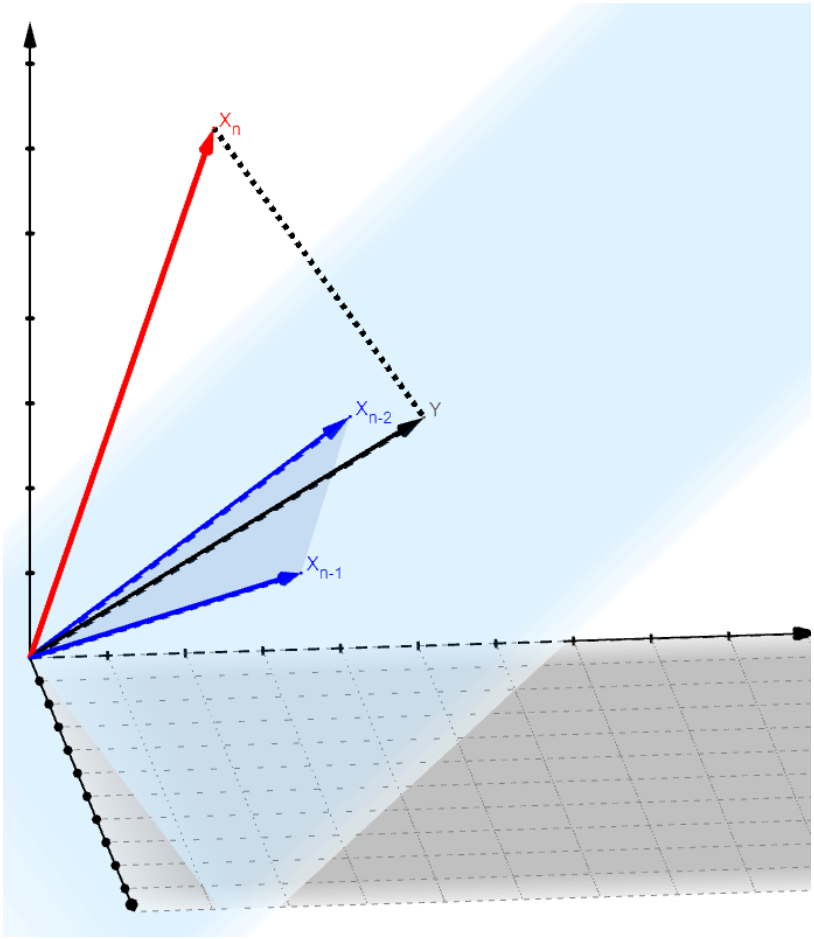


Figure 2.1: Geometric interpretation of linear prediction of $\boldsymbol{x}_n$ of order 2. The blue vectors $\boldsymbol{x}_{n-1}$ and $\boldsymbol{x}_{n-2}$ contain the observations shifted back one and two time steps, so the vectors $L\boldsymbol{x}_n$ and $L^2\boldsymbol{x}_n$, respectively. The red vector $\boldsymbol{x}_n$ is the target vector, which we want to approximate. The black vector $\boldsymbol{y}$ is the linear combination of $L\boldsymbol{x}_n$ and $L^2\boldsymbol{x}_{n-2}$ that minimizes $\|\boldsymbol{x}_n - \alpha_1 L\boldsymbol{x}_n + \alpha_2 L^2\boldsymbol{x}_n\|$ for $\alpha_1, \alpha_2 \in \mathbb{R}$. The solution $\boldsymbol{y}$ is found by projecting $\boldsymbol{X}_n$ onto the space spanned by $\boldsymbol{X}_{n-1}$ and $\boldsymbol{X}_{n-2}$.

Chapter 3

Linear Regression

With the term *linear regression*, we refer to statistical models that represent an outcome variable as a linear combination of indicator variables. If $\boldsymbol{A}$ is a matrix with observation times as rows and indicator variables as columns, and $\boldsymbol{b}$ is a vector of observations of the predictor variable, then the linear regression problem is finding the vector $\boldsymbol{x}$ such that

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}. \tag{3.1}$$

In the previous chapter, we have seen that linear time series models are examples of linear regression problems. In this chapter, we discuss how linear regression problems can be solved analytically given a data matrix of observation of variables.

Definition 3.1: Linear Regression Problem for Univariate Time Series

Let $\{x_t\}, t = 1, 2, \ldots T$ be a univariate time series. Suppose that we want to represent an observation x_t in terms of the linear combination of its p previous values $x_{t-1}, \ldots, x_{t-p}$. Then

$$\boldsymbol{A} = \begin{pmatrix} Lx_t & L^2x_t & \ldots & L^px_t \end{pmatrix}$$

(the matrix with as columns the lagged time series of $\{x_t\}$) and

$$\boldsymbol{b} = (x_{p+1}, x_{p+2}, \ldots, x_T)^T.$$

Note that in the definition above, the columns of $\boldsymbol{A}$ are not exactly L^ix_t, but actually contain the results of applying the operator L^i only on the observations of x_t at observation times $p+1, p+2, \ldots, T$. In the next example, we will clarify how this is done.

Example 3.1

Suppose we have the following sequence of observations of some variable: $(5, 6, 4, 7, 3)$. Suppose that $p = 2$. Then

$$\boldsymbol{A} = \begin{pmatrix} 6 & 5 \\ 4 & 6 \\ 7 & 4 \end{pmatrix} \quad \text{and} \quad \boldsymbol{b} = \begin{pmatrix} 4 \\ 7 \\ 3 \end{pmatrix}.$$

If $\boldsymbol{A}$ is a square matrix of full rank, then the solution to (3.1) is given by

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{b}.$$

However, when $\boldsymbol{A}$ is not square or not of full rank, it is not invertible.

In the general case that $\boldsymbol{A}$ is *underdetermined* (i.e. the number of observations is smaller than the number of coefficients to estimate), there are infinitely many solutions to equation (3.1), and another criterium is needed to select only one. One option is to select the *mimimum-norm solution*, i.e. the solution $\boldsymbol{x}$ to (3.1) such that $\boldsymbol{x} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|$.

If $\boldsymbol{A}$ is *overdetermined* (i.e. the number of observations is larger than the number of coefficients to estimate), it is possible that no exact solution exists. In this case, we want to select a solution that best fits the data, i.e. that minimizes the error between predicted values and actual values. The *least-squares solution* is the solution that minimizes the squared sum $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|^2$. [7].

To find solutions $\boldsymbol{x}$, we use a substitute for the inverse of $\boldsymbol{A}$ using Singular Value Decomposition (SVD).

3.1 Singular Value Decomposition

Definition 3.2: Singular Value Decomposition (SVD) [7]

The **SVD** of a real-valued matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ is

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{3.2}$$

where $\boldsymbol{U} \in \mathbb{R}^{n \times n}$ and $\boldsymbol{V} \in \mathbb{R}^{m \times m}$ are unitary matrices (i.e. $\boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{U}^T\boldsymbol{U} = I_n$ and $\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{V}^T\boldsymbol{V} = I_m$) with orthonormal columns, and $\boldsymbol{\Sigma} \in \mathbb{R}^{n \times m}$ is a diagonal matrix with non-negative entries on the diagonal.

Let r denote the rank of matrix $\boldsymbol{A}$. The matrix $\boldsymbol{U}$ in 3.4 contains the r orthonormal eigenvectors of $\boldsymbol{A}\boldsymbol{A}^T$, $\boldsymbol{V}$ the r orthonormal eigenvectors of $\boldsymbol{A}^T\boldsymbol{A}$, and $\boldsymbol{\Sigma}$ has as first r diagonal entries the square roots of the eigenvalues λ_i of both $\boldsymbol{A}\boldsymbol{A}^T$ and $\boldsymbol{A}^T\boldsymbol{A}$, supplemented with zeros to form a square matrix [27]. We can thus write the SVD of $\boldsymbol{A}$ in equation (3.4) as

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{u}_1 & \dots & \boldsymbol{u}_n \end{pmatrix} \begin{pmatrix} \boldsymbol{S} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{v}_1{}^T \\ \vdots \\ \boldsymbol{v}_m{}^T \end{pmatrix}, \tag{3.3}$$

where

$$\boldsymbol{S} = \begin{pmatrix} \sigma_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sigma_r \end{pmatrix}.$$

If we arrange the diagonal entries of $\boldsymbol{S}$ in decreasing order, then we can also decompose the matrix $\boldsymbol{A}$ in the following way.

Definition 3.3: Alternative Singular Value Decomposition [1, Chapter 5]

Any $m \times n$ real-valued matrix $\boldsymbol{A}$ with rank r can be decomposed as

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{3.4}$$

where $\boldsymbol{U} \in \mathbb{R}^{m \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{n \times r}$ are unitary matrices and

$$\boldsymbol{\Sigma} = \begin{pmatrix} \sigma_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sigma_r \end{pmatrix}$$

is a $r \times r$ diagonal matrix with positive entries that satisfy

$$\sigma_1 \geq \cdots \geq \sigma_r > 0.$$

Note that a SVD of a matrix is not unique [27]. A SVD exists for any matrix and can be computed numerically [7]. Once we have calculated the SVD of $\boldsymbol{X}$, we can insert it into equation 3.1 to get a solution for $\boldsymbol{x}$:

$$\begin{aligned} \boldsymbol{A}\boldsymbol{x} &= \boldsymbol{b} \\ \boldsymbol{U}\Sigma\boldsymbol{V}^T\boldsymbol{x} &= \boldsymbol{b} \\ \boldsymbol{V}\Sigma^{-1}\boldsymbol{U}^T\boldsymbol{U}\Sigma\boldsymbol{V}^T\boldsymbol{x} &= \boldsymbol{V}\Sigma^{-1}\boldsymbol{U}^T\boldsymbol{b} \\ \boldsymbol{x} &= \boldsymbol{V}\Sigma^{-1}\boldsymbol{U}^T\boldsymbol{b}. \end{aligned}$$

The solution of $\boldsymbol{x}$ in above expression can be written in terms of the *Moore-Penrose left pseudo-inverse* of $\boldsymbol{A}$.

Definition 3.4: Moore-Penrose left pseudo-inverse

The Moore-Penrose left pseudo-inverse $\boldsymbol{A}^\dagger$ of a matrix $\boldsymbol{A}$ is $\boldsymbol{V}\Sigma^{-1}\boldsymbol{U}^T$.

This expression gives the minimum-norm solution for underdetermined systems, and the least-squares solution for overdetermined systems.

Theorem 3.1: Minimum Norm Solution

For an underdetermined system $\boldsymbol{Ax} = \boldsymbol{b}$, $\boldsymbol{x} = \boldsymbol{A}^\dagger \boldsymbol{b}$ is the minimum euclidean norm solution.

Proof. Let $\boldsymbol{A}$ be a $m \times n$-matrix with $m < n$. Let $\boldsymbol{x}$ and $\boldsymbol{z}$ be solutions such that $\boldsymbol{Az} = \boldsymbol{Ax} = \boldsymbol{b}$, and let $\boldsymbol{x} = \boldsymbol{A}^\dagger \boldsymbol{b}$. We can write the euclidean norm of $\boldsymbol{z}$ as follows:

$$\|\boldsymbol{z}\|_2 = \|\boldsymbol{z} - \boldsymbol{x} + \boldsymbol{x}\|_2 = \|\boldsymbol{z} - \boldsymbol{x}\|_2 + \|\boldsymbol{x}\|_2 + 2\boldsymbol{x}^T(\boldsymbol{z} - \boldsymbol{x}). \tag{3.5}$$

Note that

$$\begin{aligned}
\boldsymbol{x}^T(\boldsymbol{z} - \boldsymbol{x}) &= \left(\boldsymbol{A}^\dagger \boldsymbol{Ax}\right)^T (\boldsymbol{z} - \boldsymbol{x}) \\
&= \boldsymbol{x}^T \boldsymbol{A}^\dagger \boldsymbol{A}(\boldsymbol{z} - \boldsymbol{x}) \\
&= \boldsymbol{x}^T \left(\boldsymbol{A}^\dagger \boldsymbol{Az} - \boldsymbol{A}^\dagger \boldsymbol{Ax}\right) \\
&= \boldsymbol{x}^T \left(\boldsymbol{A}^\dagger b - \boldsymbol{A}^\dagger b\right) \\
&= 0.
\end{aligned}$$

Filling in in equation 3.5, we get that $\|\boldsymbol{z}\|_2 = \|\boldsymbol{z} - \boldsymbol{x}\|_2 + \|\boldsymbol{x}\|_2 \geq \|\boldsymbol{x}\|_2$. So, $\boldsymbol{x}$ is the minimum norm solution. □

In figure (3.1), a geometric interpretation of the minimum norm solution is depicted. The red line contains all solutions to the system of equations $\boldsymbol{Ax} = \boldsymbol{b}$. The green lines denote levels of the l_2 norm, i.e. vectors $\boldsymbol{x}$ for which $\|\boldsymbol{x}\|^2 = {x_1}^2 + {x_2}^2 = a$ for some $a > 0$. The minimum norm solution $\boldsymbol{x}^*$ is the first intersection point of $\boldsymbol{Ax} = \boldsymbol{b}$ and ${x_1}^2 + {x_2}^2 = a$ that arises when increasing a from zero. The solution is the vector that is perpendicular to all other solution vectors.

Theorem 3.2: Least Squares Solution

For an overdetermined system $\boldsymbol{Ax} = \boldsymbol{b}$, the the *Moore-Penrose left pseudo-inverse* is the least-squares solution.

Proof. For this proof, we use the first definition of SVD (3.1) and its form as in equation (3.3). We want to find the solution x which minimizes the sum of squares of errors r^2, i.e. the $\boldsymbol{x}$ for which

$$r^2(x) := \|\boldsymbol{Ax} - \boldsymbol{b}\|_2^2$$

is minimized. Hereto, we will use that unitary transformations are invariant under the euclidean norm.

$$r^2(x) = \|\boldsymbol{Ax} - \boldsymbol{b}\|_2^2 = \left\|\boldsymbol{U}^T(\boldsymbol{Ax} - \boldsymbol{b})\right\|_2^2 = \left\|\boldsymbol{U}^T(\boldsymbol{U\Sigma V}^T\boldsymbol{x} - \boldsymbol{b})\right\|_2^2 = \left\|\boldsymbol{\Sigma V}^T x - \boldsymbol{U}^T\boldsymbol{b}\right\|_2^2.$$

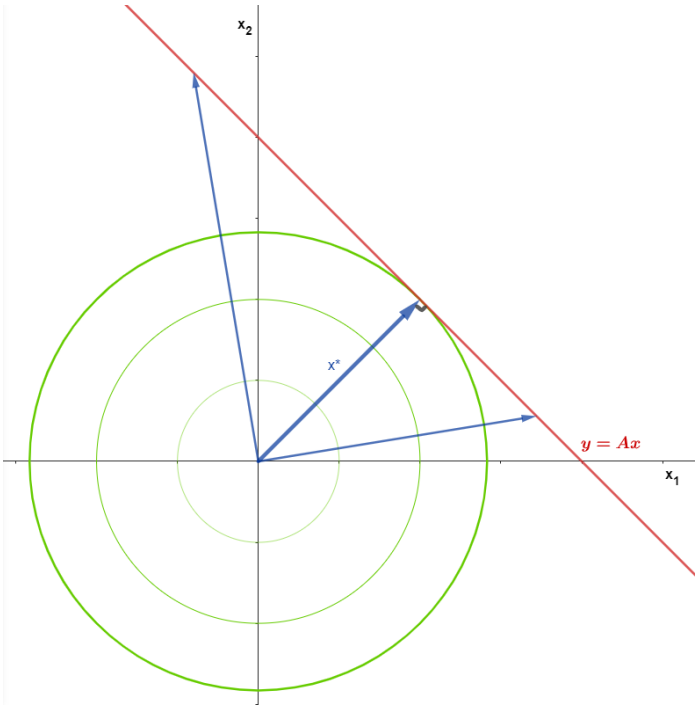


Figure 3.1: A geometric interpretation of the minimum norm solution. The red line contains all solutions $\boldsymbol{x}$ to $\boldsymbol{Ax} = \boldsymbol{y}$. The blue vectors are examples of such solutions $\boldsymbol{x}$. The green circles are level curves of the L2-norm. The solution $\boldsymbol{x}^* = \boldsymbol{A}^\dagger \boldsymbol{y}$ is the minimum norm solution, which is the first intersection that occurs between the level curves and the line of solution when increasing the norm.

$$
\begin{aligned}
r^2(x) &= \left\| \begin{pmatrix} \boldsymbol{S} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} - & {\boldsymbol{v}_1}^T & - \\ & \vdots & \\ - & {\boldsymbol{v}_m}^T & - \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} - \begin{pmatrix} - & {\boldsymbol{u}_1}^T & - \\ & \vdots & \\ - & {\boldsymbol{u}_n}^T & - \end{pmatrix} \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \right\|_2^2 \\
&= \left\| \begin{pmatrix} \sigma_1 \left(v_{11}x_1 + v_{12}x_2 + \cdots + v_{1m}x_m \right) \\ \vdots \\ \sigma_r \left(v_{r1}x_1 + v_{r2}x_2 + \cdots + v_{rm}x_m \right) \\ \mathbf{0} \end{pmatrix} - \begin{pmatrix} u_{11}b_1 + u_{12}b_2 + \cdots + u_{1m}b_m \\ \vdots \\ u_{n1}b_1 + u_{n2}b_2 + \cdots + u_{nm}b_m \end{pmatrix} \right\|_2^2 .
\end{aligned}
$$

Note that the 2-norm of a vector is a sum which we can simplify by omitting zero-terms. We then get the following expression for the error r^2:

$$
r^2(x) = \left\| \boldsymbol{S}{\boldsymbol{U}_r}^T \boldsymbol{x} - \boldsymbol{U_r}^T \boldsymbol{b} \right\|_2^2 + \left\| {\boldsymbol{U}_{-r}}^T \mathbf{b} \right\|_2^2, \tag{3.6}
$$

where $\boldsymbol{V}_r$ is the $n \times r$-matrix containing the first r column vectors of $\boldsymbol{V}$, $\boldsymbol{U}_r$ is the $m \times r$-matrix containing the first r column vectors of $\boldsymbol{U}$, and $\boldsymbol{U}_{-r}$ contains all other columns of $\boldsymbol{U}$. So, the solution $\boldsymbol{x}$ that minimizes $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2$ is the solution that minimizes equation 3.6. Because $\boldsymbol{b}$ and ${\boldsymbol{U}_{-r}}^T$ are independent on our choice of $\boldsymbol{x}$, this is the same as minimizing

$$
\left\| \boldsymbol{S}{\boldsymbol{U}_r}^T \boldsymbol{x} - \boldsymbol{U_r}^T \boldsymbol{b} \right\|_2^2 .
$$

Filling in $\boldsymbol{x} = \boldsymbol{V}_r \boldsymbol{S}^{-1} {\boldsymbol{U}_r}^T$ gives

$$
\boldsymbol{S}{\boldsymbol{U}_r}^T \boldsymbol{x} - \boldsymbol{U_r}^T \boldsymbol{b} = \mathbf{0},
$$

which is indeed the smallest possible 2-norm. □

3.2 Principal Component Analysis

Principal Component Analysis (PCA) is a popular technique for dimension reduction in data analysis. It assumes that high-dimensional data samples are drawn from a single low-dimensional subspace. PCA projects the D-dimensional data points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ onto a lower-dimensional space S of dimension $d \ll D$, in such a way that as much of the the initial variance as possible is preserved [28]. In figure (3.2), a geometric interpretation of PCA is given.

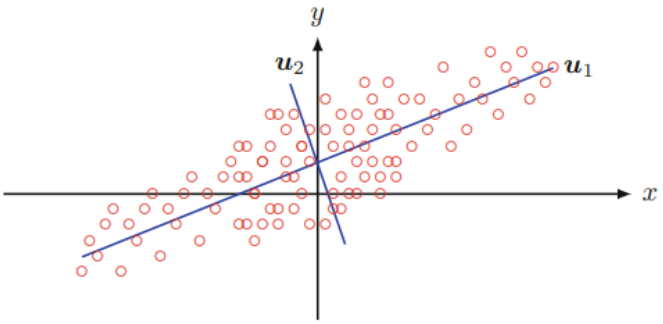


Figure 3.2: Example of a two-dimensional data set (red dots) with its two principal components $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$. From [28].

Definition 3.5: Principal Components [28, Chapter 2]

Let $\boldsymbol{x} \in \mathbb{R}^D$ be a multivariate random variable and $d < D$ the dimension of the subspace that $\boldsymbol{x}$ will be projected on. The d **principal components** of $\boldsymbol{x}$, $y_1, \ldots, y_d$, are the d uncorrelated linear combinations of $\boldsymbol{x}$, given by

$$y_i = \boldsymbol{u}_i{}^T \boldsymbol{x} + a_i \in \mathbb{R}, \quad \boldsymbol{u}_i \in \mathbb{R}^D, \quad i = 1, \ldots, d, \tag{3.7}$$

that maximize the variance of $y_1, \ldots, y_d$ consecutively and satisfy

$$\|\boldsymbol{u}_i\| = 1, \quad i = 1, \ldots d$$

and

$$\mathrm{Var}(y_1) \geq \mathrm{Var}(y_2) \geq \cdots \geq \mathrm{Var}(y_d) > 0.$$

When we consider a set $\{\boldsymbol{x}_j\}_{j=1}^N$ of realizations in $\mathbb{R}^D$ of the random variable $\boldsymbol{X}$, we can approach PCA from a geometric point of view. If we assume that there exists a subspace $S \subset \mathbb{R}^D$ of dimension d such that all points $\boldsymbol{x}_j, j = 1, \ldots, N$ lie on S, then each point $\boldsymbol{x}_j \in S$ can be represented as

$$\boldsymbol{x}_j = \boldsymbol{\mu} + \boldsymbol{U}\boldsymbol{y}_j, \quad j = 1, 2, \ldots, N,$$

where $\boldsymbol{\mu} \in S$ is a point in the subspace, $\boldsymbol{U}$ is a $D \times d$-matrix whose columns form a basis of S, and $\boldsymbol{y}_j \in \mathbb{R}^d$ is the vector of new coordinates of $\boldsymbol{x}_j$ in the subspace. As

$$\boldsymbol{\mu} + \boldsymbol{U}\boldsymbol{y}_j = (\boldsymbol{\mu} + \boldsymbol{U}\boldsymbol{z}) + \boldsymbol{U}\left(\boldsymbol{y}_j - \boldsymbol{z}\right)$$

for all $\boldsymbol{z} \in \mathbb{R}^d$, and

$$\boldsymbol{\mu} + \boldsymbol{U}\boldsymbol{y}_j = \boldsymbol{\mu} + (\boldsymbol{U}\boldsymbol{A})(\boldsymbol{A}^{-1}\boldsymbol{y}_j)$$

for every invertible matrix $\boldsymbol{A} \in \mathbb{R}^{d\times d}$, we see that the choice of $\boldsymbol{\mu}$ and $\boldsymbol{U}$ is arbitrary, and we will not be able to find a unique affine subspace for the data. By adding the condition that the columns of $\boldsymbol{U}$ are orthonormal (i.e. $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$) and

$$\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{y}_j = \boldsymbol{0},$$

a subspace S for projection of the data points can be found that is unique up to rotation [28].

Now suppose that the points $\boldsymbol{x}_j$ do not lie perfectly in S, but that

$$\boldsymbol{x}_j = \boldsymbol{\mu} + \boldsymbol{U}\boldsymbol{y}_j + \varepsilon_j, \quad j = 1, 2, \ldots, N.$$

In this case, we can write the data matrix $\boldsymbol{A}$ as

$$\boldsymbol{A} = \boldsymbol{P}\boldsymbol{L}^T + \boldsymbol{E},$$

where $\boldsymbol{P}$ is an $n \times k$-matrix whose columns contain the principal components, $\boldsymbol{L}$ is a $m \times k$-matrix whose entries are the *component loadings*, and $\boldsymbol{E}$ is an error-matrix. The matrix product $\boldsymbol{P}\boldsymbol{L}^T$ is an approximation of $\boldsymbol{A}$. The goal is to find $\boldsymbol{P}$ and $\boldsymbol{L}$ that minimize the approximation error:

$$\min_{\boldsymbol{P},\boldsymbol{L}} \|\boldsymbol{E}\|_F^2 = \left\|\boldsymbol{A} - \boldsymbol{P}\boldsymbol{L}^T\right\|_F^2,$$

where $\|\cdot\|_F$ is the Frobenius norm.

Definition 3.6: Frobenius norm [27]

The Frobenius norm of a $m \times n$-matrix $\boldsymbol{F}$ with entries $\boldsymbol{F}_{ij}$ is defined as

$$\|\boldsymbol{F}\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |\boldsymbol{F}_{ij}|^2\right)^{1/2}.$$

The solutions for above minimization problem are given through SVD [1].

Theorem 3.3: PCA through SVD [1]

Let $\boldsymbol{A}$ be a $N \times D$-matrix, with SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. Assume that $d \leq \operatorname{rank}(\boldsymbol{A}) = r$ and that the diagonal elements of $\boldsymbol{\Sigma}$ are positive and arranged in decreasing order. Let $\boldsymbol{U}_d$ denote the $N \times d$-matrix consisting of the first d columns of $\boldsymbol{U}$ and $\boldsymbol{V}_d$ the $D \times d$-matrix consisting of the first d columns of $\boldsymbol{V}$. Let $\boldsymbol{\Sigma}_d$ be the diagonal matrix with diagonal entries the first d entries of $\boldsymbol{\Sigma}$. Then,

$$\left\|\boldsymbol{A} - \boldsymbol{P}\boldsymbol{L}^T\right\|_F^2$$

with $\boldsymbol{P}$ a $N \times d$ matrix and $\boldsymbol{L}$ a $D \times d$ matrix is minimized for

$$\boldsymbol{P}\boldsymbol{L}^T = \boldsymbol{U}_d\boldsymbol{\Sigma}_d\boldsymbol{V}_d^T.$$

Proof. Let $\boldsymbol{M}$ be a real-valued $N \times D$-matrix with a SVD of $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$. We can rewrite the expression for $\|\boldsymbol{A} - \boldsymbol{M}\|_F^2$ in the following way:

$$\|\boldsymbol{A} - \boldsymbol{M}\|_F^2 = \left\|\boldsymbol{A} - \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right\|_F^2 \tag{3.8}$$

$$= \left\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T + \boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T - \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right\|_F^2 \tag{3.9}$$

$$= \left\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T\right\|_F^2 + \left\|\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T - \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right\|_F^2 + 2c, \tag{3.10}$$

with

$$\begin{aligned}
c &= \sum_{i=1}^{N}\sum_{j=1}^{D}\left(\boldsymbol{A}_{ij} - (\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T)_{ij}\right)\left((\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T)_{ij} - (\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T)_{ij}\right) \\
&= \sum_{i=1}^{N}\sum_{j=1}^{D}\left(\boldsymbol{A}_{ij}(\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T)_{ij} - \boldsymbol{A}_{ij}(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T)_{ij} - (\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T)_{ij}(\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T)_{ij} + (\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T)_{ij}(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T)_{ij}\right) \\
&= \operatorname{tr}\left(\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T\right) - \operatorname{tr}\left(\boldsymbol{A}^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right) - \operatorname{tr}\left(\left(\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T\right)^T\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T\right) + \operatorname{tr}\left(\left(\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T\right)^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right) \\
&= \operatorname{tr}\left(\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T\right) - \operatorname{tr}\left(\boldsymbol{A}^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right) - \operatorname{tr}\left(\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T\right) + \operatorname{tr}\left(\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{A}^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right).
\end{aligned}$$

Using that $\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}$ and the property of the trace that

$$\operatorname{tr}(\boldsymbol{A}\boldsymbol{B}\boldsymbol{C}) = \operatorname{tr}(\boldsymbol{C}\boldsymbol{A}\boldsymbol{B}),$$

we see that

$$c = \operatorname{tr}\left(\boldsymbol{V}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\right) - \operatorname{tr}\left(\boldsymbol{A}^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right) - \operatorname{tr}\left(\boldsymbol{V}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\right) + \operatorname{tr}\left(\boldsymbol{A}^T\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right) = 0.$$

Furthermore, we have that

$$\begin{aligned}
\left\|\boldsymbol{A} - \boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T\right\|^2 &= \|\boldsymbol{A}\|^2 - 2\operatorname{tr}\left(\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T\right) + \operatorname{tr}\left(\boldsymbol{V}\boldsymbol{V}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T\right) \\
&= \|\boldsymbol{A}\|^2 - \operatorname{tr}\left(\boldsymbol{V}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\right).
\end{aligned}$$

Filling in above expression and $c = 0$ into equation (3.10) gives us

$$\|\boldsymbol{A} - \boldsymbol{M}\|^2 = \|\boldsymbol{A}\|^2 - \operatorname{tr}\left(\boldsymbol{V}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\right) + \left\|\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T - \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right\|^2.$$

The above expression can be minimized if we can find $\boldsymbol{U}, \boldsymbol{\Sigma}$ and $\boldsymbol{V}$ that maximize $\operatorname{tr}\left(\boldsymbol{V}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\right)$ while minimizing $\left\|\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T - \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right\|^2$. The matrices

$$\boldsymbol{U} = \boldsymbol{U}_m, \quad \boldsymbol{\Sigma} = \boldsymbol{\Sigma}_m, \quad \boldsymbol{V} = \boldsymbol{V}_m$$

with $\boldsymbol{U}_m, \boldsymbol{\Sigma}_m$ and $\boldsymbol{V}_m$ as defined in (3.2) both maximize $\operatorname{tr}\left(\boldsymbol{V}^T\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{V}\right)$ and minimize $\left\|\boldsymbol{A}\boldsymbol{V}\boldsymbol{V}^T - \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\right\|^2$. For the proof of this last statement, we refer to [1, Chapter Appendix]. □

Note that the solution given by SVD does not give unique matrices $\boldsymbol{P}$ and $\boldsymbol{L}$. By adding the following constraints onto $\boldsymbol{P}$ and $\boldsymbol{L}$:

$$\frac{1}{N}\boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}$$

and adding the constraint that $\boldsymbol{L}^T\boldsymbol{L}$ is a diagonal matrix whose diagonal elements are arranged in decreasing order, the following unique solution is found:

$$\boldsymbol{P} = \sqrt{N}\boldsymbol{U}_m, \quad \boldsymbol{L} = \frac{1}{\sqrt{N}}\boldsymbol{V}_m\boldsymbol{\Sigma}_m.$$

The proportion of explained variance by the first m PCs PEV_m is given by

$$\text{PEV}_m = \frac{\text{tr}(\boldsymbol{P}\boldsymbol{L}^T)^T\boldsymbol{P}\boldsymbol{L}^T}{\text{tr}\left(N^{-1}\boldsymbol{A}^T\boldsymbol{A}\right)}$$

[1].

We can rewrite the matrix $\boldsymbol{P}$ as $\boldsymbol{A}\boldsymbol{W}$, where the matrix $\boldsymbol{W}$ is the weight matrix $\boldsymbol{W}$. Elements in $\boldsymbol{W}$ provide the weights by which variables are multiplied to form the PCs (columns of $\boldsymbol{P}$).

Chapter 4

Models

In this chapter, we explain the linear models that are implemented for this research. All models are deterministic and work on regular time series data (i.e. time series with regular intervals between observations). We start with AR(p), a simple autoregressive model for univariate time series. Then, we discuss VAR(p), the variant of AR(p) for multivariate time series. At last, we show some lower-dimensional variants of VAR(p), where the covariates are reduced to the principal components of the data, the nearest neighbours of a predictor in a geographical sense and most correlated variables.

4.1 Autoregressive model (AR)

The simplest model for time series analysis is the autoregressive (AR) model. This model assumes that an observation of a variable is a linear combination of it's previous values up to a certain lag p plus noise. Optionally, an intercept term can be added.

Definition 4.1: Autoregressive model (AR(p)) [4]

An **autoregressive model** of order p (AR(p)) of a variable X specifies a time series $\{x_t\}$ at time t as

$$x_t = \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \cdots + \varphi_p x_{t-p} + \varphi_{p+1} + \varepsilon_t,$$

where $\{x_t\}$ is stationary, $\varphi_1, \varphi_2, \ldots, \varphi_{p+1}$ are constant coefficients and $\varphi_p \neq 0$. The error term ε_t is assumed to be a Gaussian white noise series with mean zero.

We can use the AR(p) model to make predictions $\hat{x}_t$ of the underlying process $\{X_t\}$ using the following expression:

$$\hat{x}_t = \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \cdots + \varphi_p x_{t-p} + \varphi_{p+1}. \tag{4.1}$$

In order to make good predictions, the model coefficients $\varphi_1, \varphi_2, \ldots, \varphi_{p+1}$ need to be chosen so that the modeled values of $\hat{x}_t$ are as close to the observed values x_t as possible.

4.2 Vector autoregressive model (VAR)

The vector autoregressive model with order p (VAR(p)) is the multidimensional version of AR(p). In this model, it is assumed that an observation of any one variable is also dependent on past observations of other variables. The model writes observations of a variable as a linear combination of p previous observations of itself and all other variables, plus an intercept term.

Definition 4.2: Vector autoregressive model (VAR(p)) [4]

The **vector autoregressive model** of order p (VAR(p)) is of the form

$$\boldsymbol{x}_t = \left(\sum_{i=1}^{p} \boldsymbol{W}_i \boldsymbol{x}_{t-i} \right) + \boldsymbol{c} + \boldsymbol{\epsilon}_t$$

where $\boldsymbol{x}_{t-i}, i = 0, 1, \ldots$ are vectors of n stationary time series, $\boldsymbol{W}_i$ are $(n \times n)$ coefficient matrices, $\boldsymbol{c}$ is an intercept vector and $\boldsymbol{\varepsilon}_t$ is a vector of error terms, which are assumed to be Gaussian white noise series with mean zero.

Note that the VAR(p) can also be written as

$$\boldsymbol{x}_t = \begin{pmatrix} \boldsymbol{W}_1 & \ldots & \boldsymbol{W}_p & \boldsymbol{c} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_{t-1} \\ \vdots \\ \boldsymbol{x}_{t-p} \\ \mathbf{1} \end{pmatrix} + \boldsymbol{\epsilon}_t. \tag{4.2}$$

The left-most matrix in equation (4.2) is the $n \times (n \cdot p + 1)$ coefficient matrix $\boldsymbol{W}$.

The number of coefficients k is $n\,(n \cdot p + 1)$, where n is the number of variables. If k is larger or comparable to the number of observations, predictors can seem to be highly correlated, even when they are independent and identically distributed. As a result, the coefficients of important but spuriously correlated variables may become relatively small. Furthermore, high dimensional spaces have geometrical properties that are counter-intuitive, which can make the linear tools described in chapter (2) unsuitable [9]. Another problem that could arise by incorporating too many variables is overfitting, which will be discussed in the next chapter.

To prevent the problems described above, we propose to reduce the number of parameters by writing each variable as a linear combination of a restricted number of variables. In the next sections, we discuss different methods for this parameter reduction.

4.2.1 VAR with principal variables (PCA)

PCA is widely used as a dimension reduction tool. It can be applied to the VAR model by replacing the original variables by a smaller set of principal components (PCs), thereby reducing the number of coefficients to be determined. As we have seen in chapter (3), each PC explains a proportion of variance of the data. This proportion can be interpreted as a measure of importance of the PC. By selecting only the first few PCs, the dimension of the problem is reduced while trying to preserve as much variation in the data as possible. As the PCs are usually linear combinations of all the original variables, interpreting the model can be difficult. Ideally, we would like to reduce the number of

predictor variables instead of the number of PCs. We can use the PCs and their proportion of explained variance to select a subset of the variables, called the *principal variables*. We will now explain how principal variables are selected from PCs.

Let $\boldsymbol{A}$ be a $N \times D$-data matrix of D variables $X_1, \ldots, X_D$. Let V denote the set of variables. When we perform principal component analysis on $\boldsymbol{A}$, we get D principal components $y_1, \ldots, y_D$, sorted in decreasing order of proportion of explained variance PEV. The expression of the i-th PC is as follows:

$$y_i = w_{i1}X_1 + w_{i2}X_2 + \cdots + w_{iD}X_D + a_i, \quad w_{i1}, \ldots, w_{iD}, a_i \in \mathbb{R},$$

and so we can write the vector of PCs $(y_1, y_2, \ldots, y_D)^T$ as

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_D \end{pmatrix} = \begin{pmatrix} w_{11} & w_{12} & \ldots & w_{1D} \\ w_{21} & w_{22} & \ldots & w_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ w_{D1} & w_{D2} & \ldots & w_{DD} \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_D \end{pmatrix} = \boldsymbol{W} \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_D \end{pmatrix}.$$

If the variables in V all have mean-zero and are defined on a similar scale, the entries in $\boldsymbol{W}$ contain information on the importance of each variable for determining the PCs. Together with the proportions of explained variance for each PC, we define the importance measure $f : V \times \{1, 2, \ldots, D\} \mapsto \mathbb{R}$ for the first $d < D$ PCs:

$$f(X_i, d) := \mathrm{PEV}_1 \left|\boldsymbol{W}_{i1}\right| + \sum_{j=2}^{d} \left(\mathrm{PEV}_i - \mathrm{PEV}_{i-1}\right) \left|\boldsymbol{W}_{ij}\right|,$$

where d is the number of PCs that are used and PEV_i is the proportion of explained variance of the first i PCs. We use this measure to define the set $PV_{d,k}$ of k principal variables.

Definition 4.3: Principal Variables

The set $PV_{d,k}$ of k **principal variables**, based on the first d PCs, is the set

$$\mathrm{PV}_{d,k} \subseteq V$$

such that

$$|\mathrm{PV}_{d,k}| = k$$

and

$$f(X_i, d) \leq \min_{X_r \in \mathrm{PV}_{d,k}} f(X_r, d)$$

for all $X_i \notin \mathrm{PV}_{d,k}$.

With this set, we can define the vector autoregression model with principal variables ($PCA(p, d, k)$).

Definition 4.4: Vector Autoregression with Principal Variables ($PCA(p, d, k)$)

The **vector autoregression model with principal variables** ($PCA(p, d, k)$) of order p, based on the first d principal components of the data matrix, and with k principal variables is of the form

$$\boldsymbol{x}_t = \left(\sum_{i=1}^{p} \boldsymbol{W}_i \boldsymbol{x}_{t-i} \right) + \boldsymbol{c},$$

with $\boldsymbol{W}_i = (\boldsymbol{w}_{i1}\ \boldsymbol{w}_{i2}\ \ldots \boldsymbol{w}_{iD})$, where

$$\boldsymbol{w}_{ij} = \boldsymbol{0} \text{ if } X_j \notin PV_{d,k}$$

for all $i = 1, 2, \ldots, p$.

4.2.2 VAR with K-Nearest Neighbours (KNN)

Let $\{\boldsymbol{x}_t\}$ be a n-dimensional time series, with $V = \{X_1, X_2, \ldots, X_n\}$ the set of variables. If a distance measure $d : V \times V \mapsto \mathbb{R}$ between all pairs of variables is defined, for instance the euclidean distance, we can for each variable $X_i \in V$ define a set $\text{KNN}_{X_i,k}$ of its k nearest neighbors.

Definition 4.5: k-Nearest Neighbors

Let V be a set of variables with distance measure d. For any variable $X_i \in V$, the set of its k **nearest neighbors** is the set

$$\text{KNN}_{X_i,k} \subseteq V$$

such that

$$|\text{KNN}_{X_i,k}| = k$$

and

$$d(X_i, X_j) \geq \max_{X_r \in \text{KNN}_{X_i,k}} d(X_i, X_r)$$

for all $X_j \notin \text{KNN}_{X_i,k}$.

Definition 4.6: k-Nearest Neighbours model (KNN(p, k))

The **K-Nearest Neighbours model** (KNN(p, k)) of order p with k predictor variables is of the form

$$\boldsymbol{x}_t = \left(\sum_{i=1}^{p} \boldsymbol{W}_i \boldsymbol{x}_{t-i} \right) + \boldsymbol{c}$$

where $\boldsymbol{x}_t$ is a vector of n stationary time series $X_1, \ldots, X_n$, $\boldsymbol{W}_i$ are $n \times n$ coefficient matrices, $\boldsymbol{c}$ is an intercept vector and

$$(W_i)_{rs} = 0 \text{ if } X_s \notin \text{KNN}_{X_r,k}.$$

4.2.3 VAR with correlation-based neighbours (CORR)

The idea behind the correlation based model is to detect and leave out uninformative variables. Hereto, the correlations between time series of variables with lagged time series of all variables are calculated. We will use $\boldsymbol{x}_i$ to denote the set of observations of variable X_i, and $L^l\boldsymbol{x}_i$ to denote the set of observations of X_i that are shifted backwards l time steps. Correlations of time series of variables X_r with lagged time series of variable X_s are calculated. An example of a correlation function that can be used is the Bravais-Pearson Correlation Coefficient.

Definition 4.7: Bravais-Pearson Correlation Coefficient [8, Chapter 4]

The **Bravais-Pearson Correlation Coefficient** r of paired observations $(x_i, y_i), i = 1, 2, \ldots, n$ of two variables X and Y measures how linearly correlated X and Y are and is given by

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

where $\overline{x}$ and $\overline{y}$ are the mean of the observations of X and Y respectively. Note that $-1 \leq r \leq 1$. When $|r|$ is close to 1, the linear correlation between X and Y is considered strong. When $|r|$ is close to 0, this correlation is considered weak.

We can define a neighborhood of nearest neighbors in the sense of the correlation measure in a similar way as we did for k-Nearest Neighbors. However, we will now make this neighborhood dependent of the order p that is used in the model. A variable X_j is added to the neighborhood of X_i if for any lag $l = 0, 1, \ldots, p$, the correlation $\text{cor}(\boldsymbol{x}_i, L^s\boldsymbol{x}_j)$ is *small enough*. We will call the set of neighbors the k-Most Correlated Neighborhood. What is meant with "small enough" will become clear in the following definition.

Definition 4.8: k-Most Correlated Neighbors

Let V be a set of variables with correlation measure cor and let p be a positive integer. For any variable $X_i \in V$, the set of its k **most correlated neighbors** is the set

$$\text{KMCN}_{X_i,k,p} \subseteq V$$

such that

$$|\text{KMCN}_{X_i,k,p}| = k$$

and

$$\min_{l=0,1,\ldots,p} \left|\text{cor}\left(\boldsymbol{x}_i, L^l\boldsymbol{x}_j\right)\right| \leq \min_{\substack{X_r \in \text{KMCN}_{X_i,k,p} \\ s=0,1,\ldots,p}} |\text{cor}(\boldsymbol{x}_i, L^s\boldsymbol{x}_r)|$$

for all $X_j \notin \text{KMCN}_{X_i,k,p}$.

If for any X_i, X_j and $l = 0, 1, \ldots, p$ the observation times of variables X_i and $L^l X_j$ are not equal, only the observation times on which both variables are observed are used for the calculation of $\text{cor}\left(\boldsymbol{x}_i, L^l\boldsymbol{x}_j\right)$. We will now define the model that is based on the k-Most Correlated Neighborhoods.

Definition 4.9: k-Most Correlated Neighbours model (CORR(p,k))

The **k-Most Correlated Neighbours model** (CORR(p,k)) of order p with k predictor variables is of the form

$$\boldsymbol{x}_t = \left(\sum_{i=1}^{p} \boldsymbol{W}_i \boldsymbol{x}_{t-i} \right) + \boldsymbol{c}$$

where $\boldsymbol{x}_t$ is a vector of n stationary time series $X_1, \ldots, X_n$, $\boldsymbol{W}_i$ are $(n \times n)$ coefficient matrices, $\boldsymbol{c}$ is an intercept vector and

$$(W_i)_{rs} = 0 \text{ if } s \notin \text{KMCN}_{X_r,k,p}.$$

4.3 Baseline model

In our analyis, we will compare the models already mentioned with a baseline model which copies the last observed value of each STP. We will call this model LAST. The model is of the following form:

$$\boldsymbol{x}_t = \boldsymbol{x}_{t-i}.$$

In the next chapter we will see some results of the models on the SARS-CoV-2 data sets.

Chapter 5

Data analysis

In this thesis, we use algorithms to build regression models from data that approximate the true, unknown rules of the data. In this chapter, we describe the process of splitting the available data in separate sets, and show how models are trained and evaluated using these different sets. We also describe preprocessing steps that are applied to the data before training takes place.

5.1 Training, validating and testing

In the first phase of model building, called the *training phase*, a subset of the data (called the *training set*) is used to determine model coefficients. Let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ the output space. Then each element of the training set (or *training instance*) can be written as a tuple $(\boldsymbol{x}, \boldsymbol{y})$ where $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathcal{Y}$, and a mapping $f : \mathcal{X} \mapsto \mathcal{Y}$ is established [32]. We have already encountered an example, where we wanted to determine coefficients $\boldsymbol{\varphi}$ s.t. $f : \boldsymbol{A} \mapsto \boldsymbol{b}$ with $f(\boldsymbol{A}) = \boldsymbol{A}\boldsymbol{\varphi}$ for a data matrix $\boldsymbol{A}$ and a target vector $\boldsymbol{b}$.

Once a model (or mapping) has been established, it can be used to make predictions based on new input. To test how well the model generalizes to new samples, the model predicts outcomes on the *test set*, which consists of data instances that have not been used in the training phase. Each *test instance* is a tuple $(\boldsymbol{x}^*, \boldsymbol{y}^*)$. Using an error measure, outcomes $\boldsymbol{y}^*$ and $\hat{\boldsymbol{y}}^* = f(\boldsymbol{x}^*)$ are compared for each test instance, and so we get an indication of the generalizability of the model [32]. In this thesis, we will measure model performance with the *root mean squared error* (RMSE).

There are often multiple candidate algorithms to model the data, and each algorithm comes with parameters for which (infinitely) many values can be chosen. To choose between algorithms and parameter settings, a *validation set* is used. All algorithms are trained for a number of parameter combinations. The performance of these models is tested on the validation set. Then, the optimal model is selected, which is the model with parameters that minimize the error on the validation set. Training is then repeated, now using data points of both the training and validation set. The model performance on the test set gives an indication of how well the models performs on new data.

Testing the models performance on new data can prevent over- and underfitting. *Overfitting* is the phenomenon where the model is too complex, thereby fitting the training data very closely, with as result a loss of generalizability of the model [26]. This loss of generalizability occurs because noise in the training data is seen as part of the underlying rules of the data. In general, the higher the model

complexity, the smaller the error on the training set. Overfitted models will change significantly if a different training set is used. *Underfitting* occurs when the model is not flexible enough, and is therefore not able to capture the underlying rules governing the data, and does not learn the relationship between input and output properly. An underfit model depends very little on the training data. A good model should balance complexity and goodness of fit [28].

All available samples are divided over the 3 sets, where the training set consists of 60% of the data and the validation and test set both of 20%. We have chosen to fill the training, validation and test set chronologically, so that the training set contains the oldest observations and the test set the most recent observations.

5.2 Preprocessing

During data preprocessing, some techniques are applied to the original data, resulting in a data set that is better suited for model building.

Missing values and regularity

For the models discussed in the previous chapter, missing values are problematic. We have seen that the models require inputs $\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2}, \ldots$, and so all observations at all observation times should be available. Furthermore, we require the time series to be regular, so there should be regular time intervals between measurements. From the original data, a regular data set can be constructed using linear interpolation.

Definition 5.1: Linear Interpolation for Time Series Data

Given two consecutive observations x_t and x_{t+h} with $h \geq 1$ of a variable X, we define

$$x_{t+i} = x_t + i\left(\frac{x_{t+h} - x_t}{h}\right)$$

for all $i = 1, \ldots, h$.

If observations at the beginning or end of the time series are missing, the first and last observation respectively are copied to create regular data.

Moving Average Smoothing

Smoothing is a technique to reduce variation between time consecutive steps, which hopefully reduces noise and enables the model to fit the underlying rules of the data. From a time series $\{x_t\}$ of original observations, a new time series $\{y_t\}$ is created whose observations y_t are averages of a specified interval (window) around the original observation x_t. To determine $\{y_t\}$, first a window size w is chosen. Then, a window of this slides along the time series and averages of the window are calculated. When the window contains the original observations and $w-1$ previous values, the resulting time series is specified as follows:

$$y_t := \frac{1}{w}\left(x_t + x_{t-1} + \cdots + x_{t-w}\right), \quad t > w.$$

Chapter 6

Application: SARS-CoV-2 virus loads in sewage

In this chapter, we describe in more detail the data set that is used in this research, and the preprocessing steps that have been performed before the modelling phase. We use RIVM's open data set on wastewater measurements, which can be accessed through [18]. Wastewater measurements are created by taking 24-hour samples of wastewater at each STP. From each sample the number of SARS-CoV-2 virus particles present in the sample is determined. Then, this number is corrected for the daily flow of wastewater through the STP and is normalized for the number of inhabitants the STP serves. The resulting data set contains observations of the sampling date of the measurement, the name of the STP from which the sample was taken and the average concentration of SARS-CoV-2 RNA, per 100,000 inhabitants and corrected for daily flow. Sampling frequencies vary over time and between STPs. In 2022, each STP is in principle sampled four times a week. The number of virus particles in a sample can be too small to be measured. In this case, the value is recorded as zero [18]. In this research, we make use of observations from 01-01-2020 to 20-10-2022.

6.1 Preprocessing steps

First, all zero-values in the data sets were replaced by 10^{12}, which is approximately half of the cut-off point for SARS-CoV-2 RNA detection in sewage samples. Then, a $\log_{10}$-transformation was applied to all observations. The following STPs were removed: Aalst, Woerden, Katwoude, Eck-en-Wiel, Valburg, Lienden and Veenendaal. These STPs were left out due to consistently deviating values (Woerden, Katwoude), large periods without measurements (Aalst, Eck-en-Wiel, Valburg, Lienden) or the catchment size was unknown at the time of implementation (Veenendaal).

In the SARS-CoV-2 data set, irregular intervals are present due to irregular sampling intervals. We use different methods to construct a data set with regular time intervals, one of which is linear interpolation as described in the previous chapter. A moving average with window size 3 was applied to this data. The resulting data set will be referred to as the *linear data set*.

We also make use of a data set of estimated $\log_{10}$-transformed virus loads based on a multilevel Bayesian penalized spline model, of which a detailed description can be found in [3]. We will refer

to this data set as the *modelled data set* from here on.

6.2 Additional constraints

The following constraints on predictions $\hat{x}_t$ were added to the models:

$$12 \leq \hat{x}_t \leq 15.25,$$

and

$$0.76 \leq \hat{x}_t - \hat{x}_{t-1} \leq 0.76,$$

to decrease unrealistic behaviour of the models. These constraints are based on the maximum and minimum values observed, and the maximum difference between two consecutive observed values between 2020 and 2022.

Chapter 7

Results

In this chapter, we discuss our main results. We start by discussing the 7-step predictions of the various models for different training and test sets. Next, the model performances are compared to the baseline model and we indicate optimal parameter settings and show figures of the predictions. Then, in section (7.2), we further investigate the behaviour of the multivariate models by inspecting the coefficients matrices for different parameter settings. In section (7.3), we discuss how model performance differs from STP to STP.

7.1 7-step forecasts

In this section, we present the results of 7-day ahead predictions on the validation set and test set as described in chapter (6), both for the linear data set and the modelled data set. Training for AR and VAR was repeated for each $p \in \{1, 2, \ldots, 7\}$. For the models KNN, CORR and PCA, training was repeated for each combination of $p \in \{1, 2, \ldots, 7\}$ and at least values 17 values of k, where k is the number of covariates in the prediction for each STP. The number of principal components d that were used to determine the principal components were 41 and 4 for the linear data and the modelled data, respectively, with corresponding proportions of explained variance of 0.7559 and 0.9986.

For each model, the optimal p or optimal combination of p and k (i.e. the parameter(s) that minimized the total RMSE on the validation set) was selected. Then, training was repeated for the optimal parameter(s), using both the training and validation set. In table (7.1), the optimal parameters are given.

	data set			
	linear		modelled	
	p	k	p	k
AR	6		7	
VAR	7		7	
PCA	7	240	2	31
KNN	6	1	2	1
CORR	6	1	2	1

Table 7.1: Parameter combinations that were found to minimize the average RMSE of all STPs on the validation set for each model.

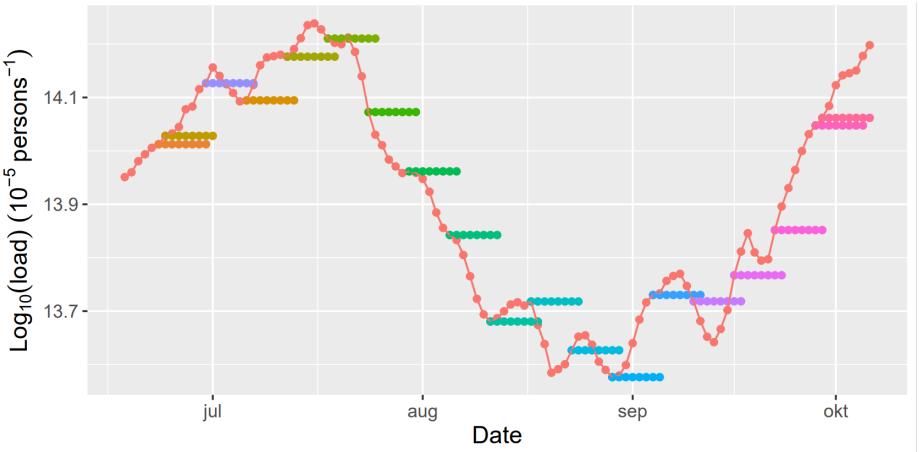
Table (7.2) contains the errors on the test set for each model with optimal parameters and each data set. Here, the error is the average error of all STPs and all 7-day predictions on the test data.

	data set	
	linear	modelled
LAST	0.2433	0.0596
AR	**0.2006**	0.0065
VAR	0.2431	**0.0010**
PCA	0.2427	0.0830
KNN	0.2191	0.0086
CORR	0.2191	0.0086

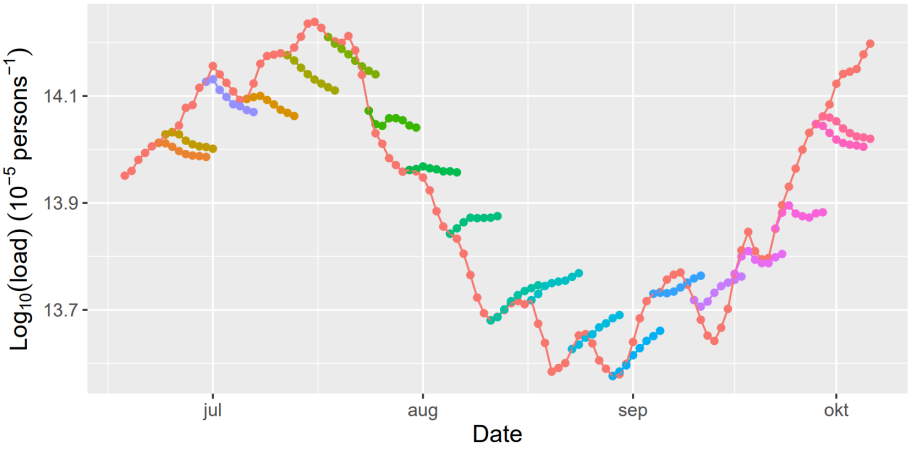
Table 7.2: Average RMSE of all STPs on the test set of the models with optimal parameters as found on the validation set. Smallest RMSEs found per data set are written in bold.

In figure (7.1), the weighted average of some 7-step predictions are plotted for the models LAST, AR(6), VAR(7) and PCA(7, 240) and the linear data set. The weighted average was calculated with as weights the sizes of the catchment areas of the STPs. The results for KNN and CORR are omitted, as the optimal parameter combination was found to be $(p = 6, k = 1)$ for both these models. The resulting predictions were very similar to those of AR($p = 6$), and are therefore not shown.

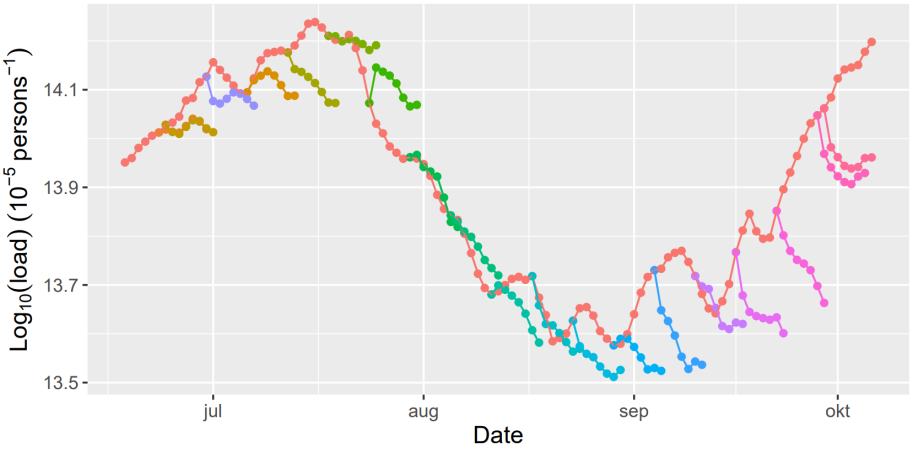
(a) Last



(b) AR(6)



(c) VAR(7)



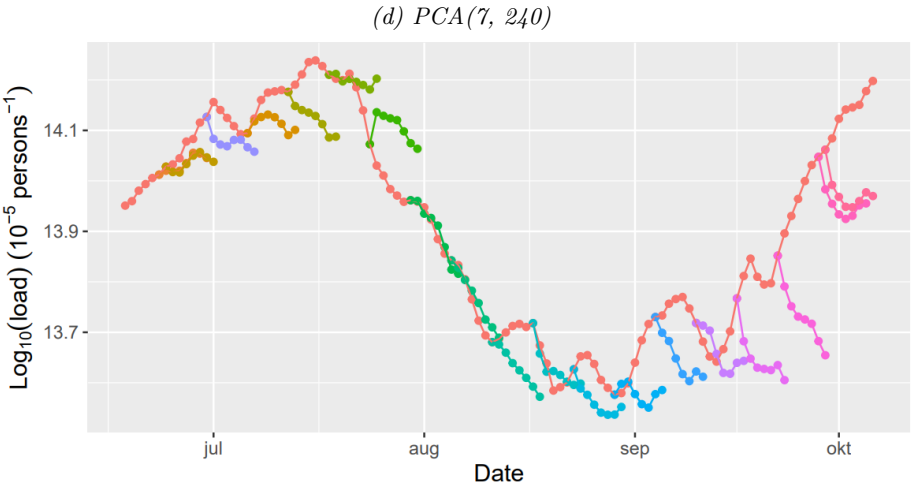
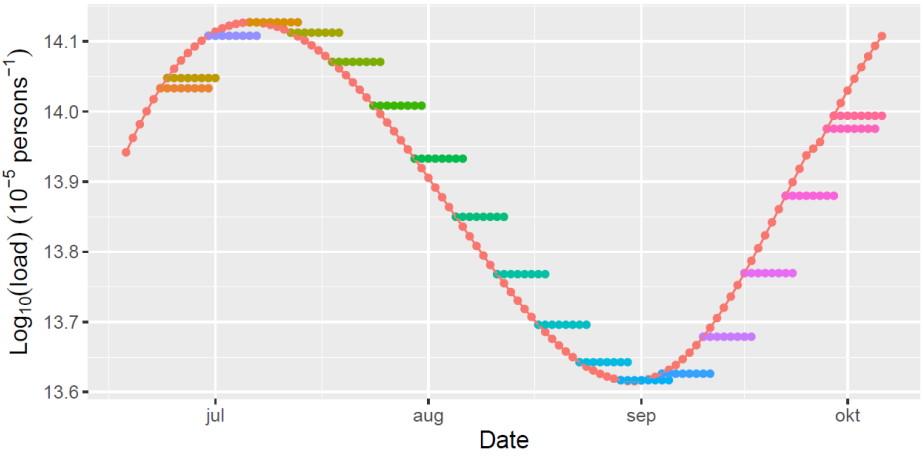


Figure 7.1: Predictions on the test set for the linear data set. The red points are actual values and the coloured lines are 7-step predictions, averaged over all STPs. Predictions of the models LAST, AR(p = 6), VAR(6) and PCA(p = 7, d = 41, k = 240) are shown.

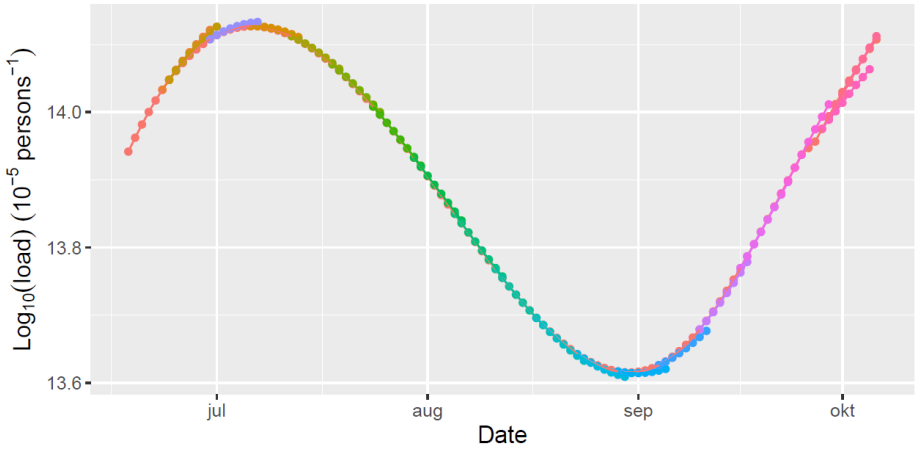
The RMSE on the test set for the linear data set is smallest for AR(6). For the linear data, all VAR models have a RMSE that is close to that of the baseline model. Looking at figure (7.1), we notice the following differences in predictions between models. For the linear data set, the AR predicts almost straight lines, which do not necessarily follow the trend of the data. The VAR and PCA predictions are more flexible, which explains the smaller training errors for these models. Both VAR and PCA predict well between mid July through mid August, when the virus loads go steeply downwards, but perform poorly when the trend rises steeply (mid September through October). The results of the VAR and PCA model look very similar, due to the large number of principal variables used in the PCA model.

In figure (7.2), the weighted average of some predictions are plotted for the models LAST, AR(7), PCA(2, 31) and KNN(2, 1) and the modelled data set.

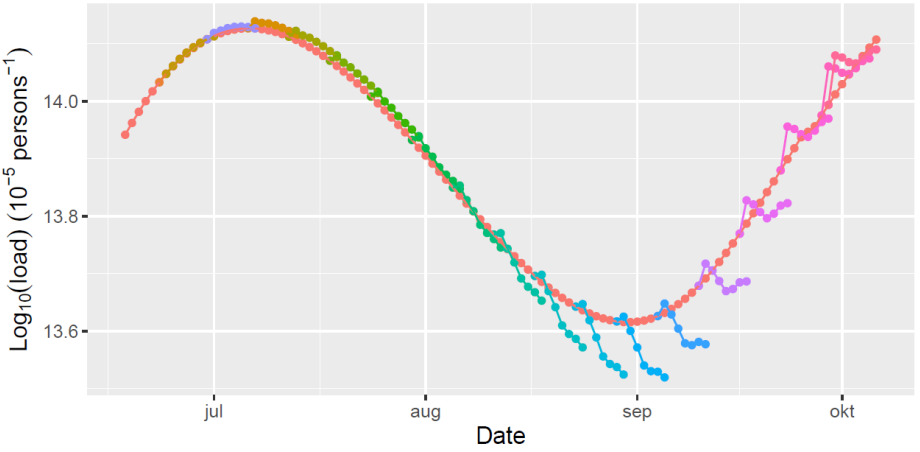
(a) Last



(b) AR(7)



(c) PCA(2, 31)



(d) KNN(2, 1)

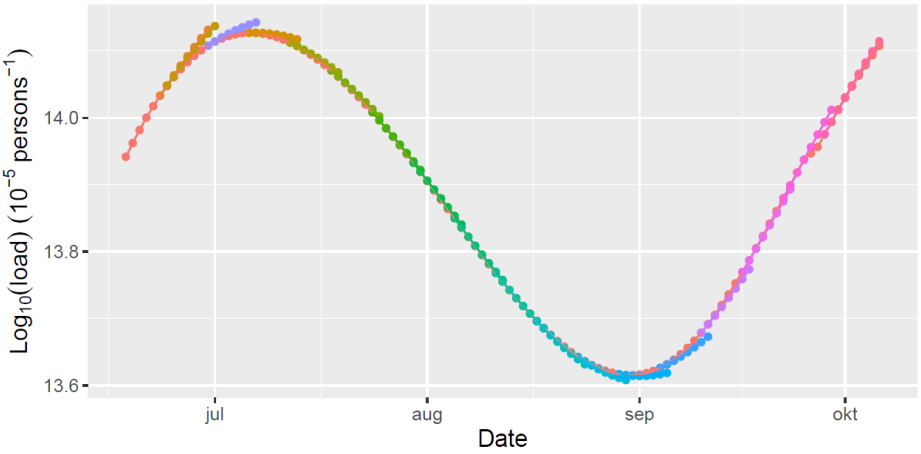


Figure 7.2: Predictions on the test set for the modelled data set. The red points are actual values and the coloured lines are 7-step predictions, averaged over all STPs. Predictions of the models LAST, AR(p = 7), PCA(p = 2, d = 4, k = 31) and KNN (p = 2, k = 1) are shown.

For the modelled data, all models outperform LAST on the first half of the test set. However, the RMSE for the PCA model was relatively large. When we look at the predictions, we see that the PCA model predicts well at the beginning of the test set, with very bad predictions from mid August on. For the KNN model, the predictions for June and July after 4 or 5 steps lie relatively far from the real curve.

For the linear data, it stands out that prediction lines seem to deviate more from the actual values when the starting date of the prediction lies further away from the date of the last training instance. For the modelled data, this phenomenon is observed for the PCA model. To check if this is indeed the case, we have plotted the average RMSE for 7-step predictions on the linear data set, starting at each date in the test set. The results are given in figure (7.3).

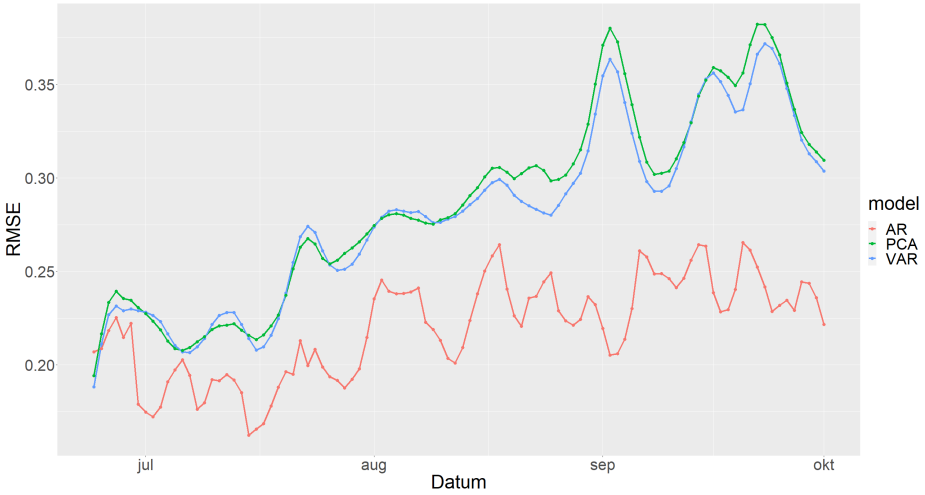


Figure 7.3: RMSE on 7-step predictions of each date in the test set.

We notice two large peaks in the RMSE for both the PCA and VAR model at the beginning and around the middle of September. When comparing figure (7.3) to the actual values of the test set in figure (7.1), we see that these peaks occur at dates where the virus loads are shaped like a parabola opening upward, so where the virus loads starts rising after it had been decreasing. For VAR, PCA and AR, a rising trend in the RMSE is observed, meaning that the RMSE grows when the start date of a prediction lies further away from the last training instance (which was 23-06-2022). This trend is present to a greater extent for the VAR and PCA models. We expect that the model performances benefit from training up to the point of prediction, so for each test instance $\boldsymbol{x}_t$, the model is trained on data from $t-h$ up to $t-1$ for some $h > 1$. In this case, training is repeated very often, and to keep computation time within limits, we need to choose h appropriately.

In figure (7.4), the RMSE of 7-step predictions starting from three different dates (24-06-2022, 01-08-2022, 24-09-2022) is plotted for different sizes of the training set. The model that was used was PCA($p = 7, d = 41, k = 240$) on the linear data set. The training set consisted of the observations of the h days previous to the test date.

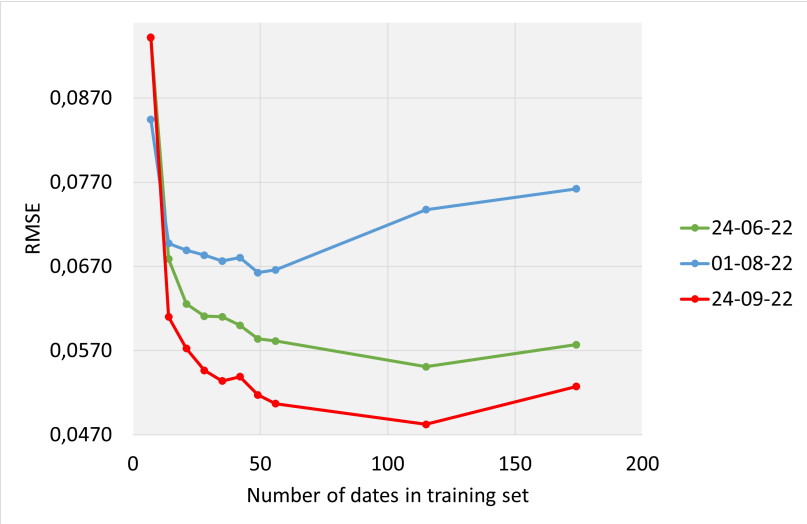


Figure 7.4: RMSE of 7-step predictions of the PCA($p = 7$, $d = 41$, k) model for different sizes of training set $k \in \{7, 14, 21, 28, 35, 42, 49, 56, 115, 174\}$ and different starting dates of predictions (24-06-2022, 01-08-2022, 24-09-22).

We see that by increasing h, the RMSE first decreases, but after some point, the RMSE starts to increase again. This trend is observed for all dates. We have run the PCA model again with $p = 7, d = 41$ and $k = 60$, with training sets of size $h = 49$. Training was repeated before each new 7-step prediction. In figure (7.5), the RMSE of 7-step predictions of this model is plotted for each fifth day in the test set.

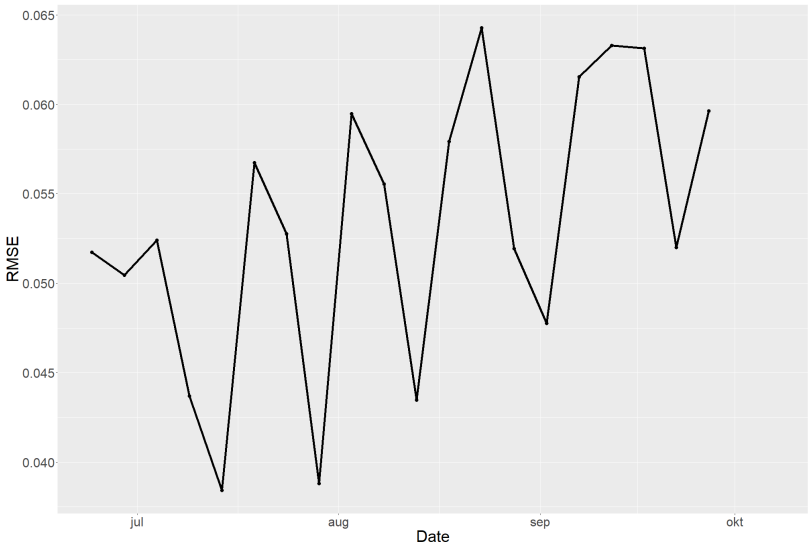


Figure 7.5: RMSE of 7-step predictions for different dates.

All RMSEs in figure (7.5) lie between 0.0384 and 0.0643, which are small values compared to the results in table (7.2). The RMSE shows an increasing trend in time.

7.2 Inspection of the validation errors

In this section, we compare the results on the validation set for the PCA and KNN model for different numbers of variables that were used in the model. In figures (7.6) and (7.7), the RMSE of 1-step predictions on the training set, the RMSE of 7-step predictions on the validation set and the Frobenius norm of the coefficient matrix are given for different number of variables used in the models. For PCA, the number of variables is $p \times k + 1$, for KNN this is $p \times (k + 1) + 1$.

For both data sets, we observe that a decrease in training error is not paired with a decrease in validation error. We observe that the models indeed find the minimum norm solution, as the norm starts to decrease from the point where the error on the training set is approximately zero. For both data sets, the validation error is highest for intermediate numbers of variables. In these cases, coefficient matrices with relatively large norms are found. For the linear data, we suspect the cause for the large norm solution might be the following. As variables are added to the model, the model becomes more flexible, and it will be able to better fit the training data. However, the training data is noisy and contains much variations between data points. So, by better fitting the training data, the variance of predictions $\boldsymbol{A}\boldsymbol{\varphi}$ (where $\boldsymbol{A}$ is the data matrix) becomes larger. We can write

$$\text{Var}\left(\boldsymbol{A}\boldsymbol{\varphi}\right) = \boldsymbol{\varphi}^T \boldsymbol{\Sigma} \boldsymbol{\varphi},$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of $\boldsymbol{A}$. Above expressions describes how increasing variance and larger values in $\boldsymbol{\varphi}$, and thus a larger norm, are related to each other. When the large norm solutions are then used to predict test instances, the differences between test instances and training instances are magnified by the large coefficients.

For the modelled data, the KNN model shows a similar curve in the test error to the KNN and PCA for the linear data. Both KNN and PCA show oscillations in the test error plot. What is remarkable is the extremely large norm of the solution for PCA at approximately 7 variables. We also note that the norm of the solution grows, even though the training error is already close to zero.

In figure (7.8), the coefficient matrices for $PCA(p = 5, d = 1, k)$ are visualised for different values of k. In Appendix B, a description of how the coefficient matrices are visualised is given. The principle variables that were used are depicted with bold black dots.

Overall, the error on the validation set was smallest for $k = 294$ and largest for $k = 32$. For small values of k, the value $k = 7$ resulted in the lowest RMSE. We first focus on the first 4 networks in figure (7.8). We see that increasing k from 6 to 7 makes the network less dense. As the edge width corresponds to the relative coefficient size, we conclude that for $k = 7$ the coefficient matrices contains some kind of *outlier*. The size of these coefficients are so large, that the contributions by other coefficients is small. In the first 8 networks, we notice that adding a variable to the model changes the network significantly. For the transitions from $k = 6$ to $k = 7$, from $k = 7$ to $k = 8$ and $k = 32$ to $k = 33$, the added variable becomes an important predictor, which can be seen by the relatively large degree of that variable. For $k = \{31, 32, 33, 34\}$, the networks contain a small number of strongly connected components, which is best observed in $k = 33$ and $k = 34$. The most strongly connected STP here is Soerendonk. We already showed that intermediate values of k have coefficient matrices with large norms. As we observe only a small number of strongly connected components in the networks for intermediate values of k, we assume that the large norm is caused by only a small number of extreme values, that occur in the same column(s) of the coefficient matrix. For large values of k ($k \in \{292, 293, 294, 295\}$, the edges in the network are distributed more evenly than for smaller values of k. An explanation for this is that by increasing k, there are more solutions to perfectly fit the training instances. As we saw in chapter (3), the minimum norm solution is selected. This minimum norm tries to avoid extremely large absolute coefficients. As k increases, the degree of freedom to do this (avoiding extremely large values) grows.

7.3 Predictive power per STP

In this section, we try to better understand the predictive power of the models by inspecting errors on the level of STPs. In figure (7.9), boxplots of the RMSEs of each STP on the linear data set is given for the models AR, VAR, PCA and KNN.

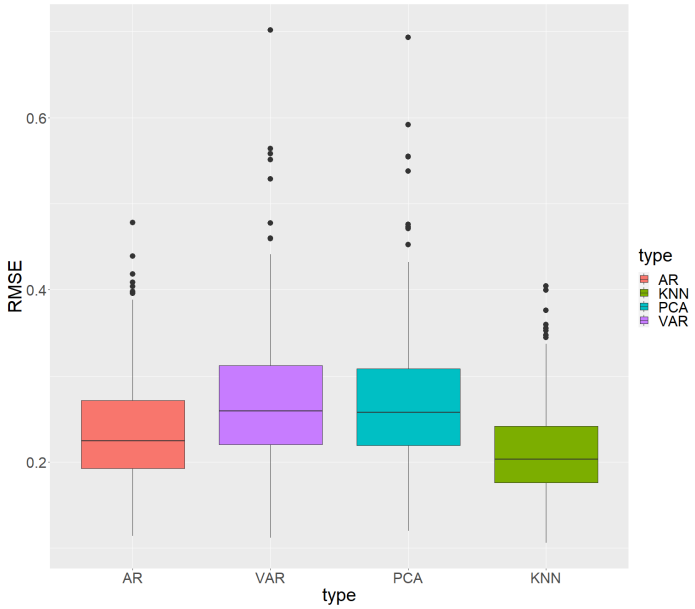


Figure 7.9: Boxplot of RMSEs per STP for the AR, VAR, PCA and KNN model, tested on the linear data set.

What is striking is that when outliers are left out of consideration, the KNN model actually has better results than AR. For PCA and VAR, outliers have higher RMSE than for KNN and AR. The medians lie relatively close to each other.

In figure (7.10), for each STP the log-10 transformed RMSE is plotted against the log-10 transformed population size of its catchment area. For each model, a negative correlation between population size and RMSE is observed, meaning that the predictive power of all models is smaller for STPs with small catchment areas.

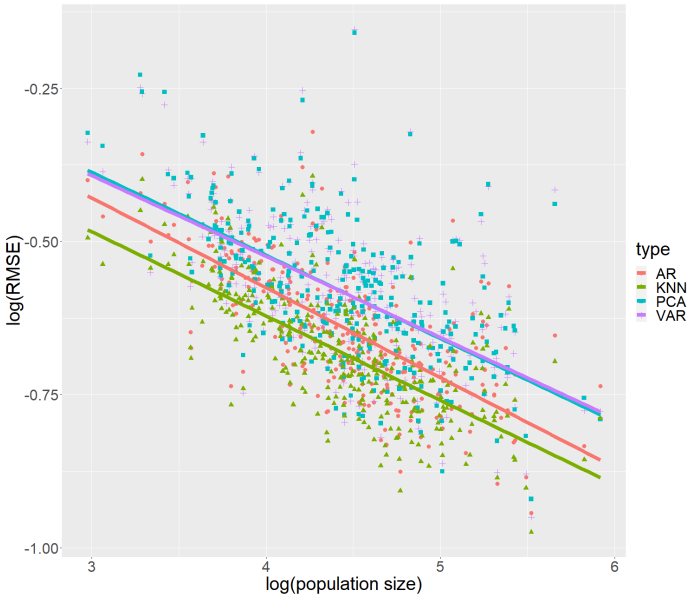


Figure 7.10: The log10 RMSE plotted against the log10-population size for each STP and the models AR, KNN, PCA and VAR. RMSEs are calculated on the test set for the linear data set.

In figure (7.11), the RMSEs for each STP is shown in the map of the Netherlands for AR, VAR, KNN and PCA on the linear data set. The size of the blue dot corresponds to the size of the error. We notice that for all models, the islands and other northern STPs have relatively large errors. For AR and KNN, we see larger errors in the region spanning from Zeeland to Twente. This region, together with two STPs in Leewarden and Groningen with large errors seem to correspond to the Biblebelt.

In figure (7.12), the population sizes of the STPs are included for AR and VAR. Here, the size of the red dot corresponds to the population size of the STP, and the color responds to the RMSE, with darker reds corresponding to larger RMSEs. We note that the STPs in the region mentioned before are relatively small, and thus have larger RMSE. For VAR, we notice that large STPs in northern Limburg and southeast Brabant have remarkably large RMSE.

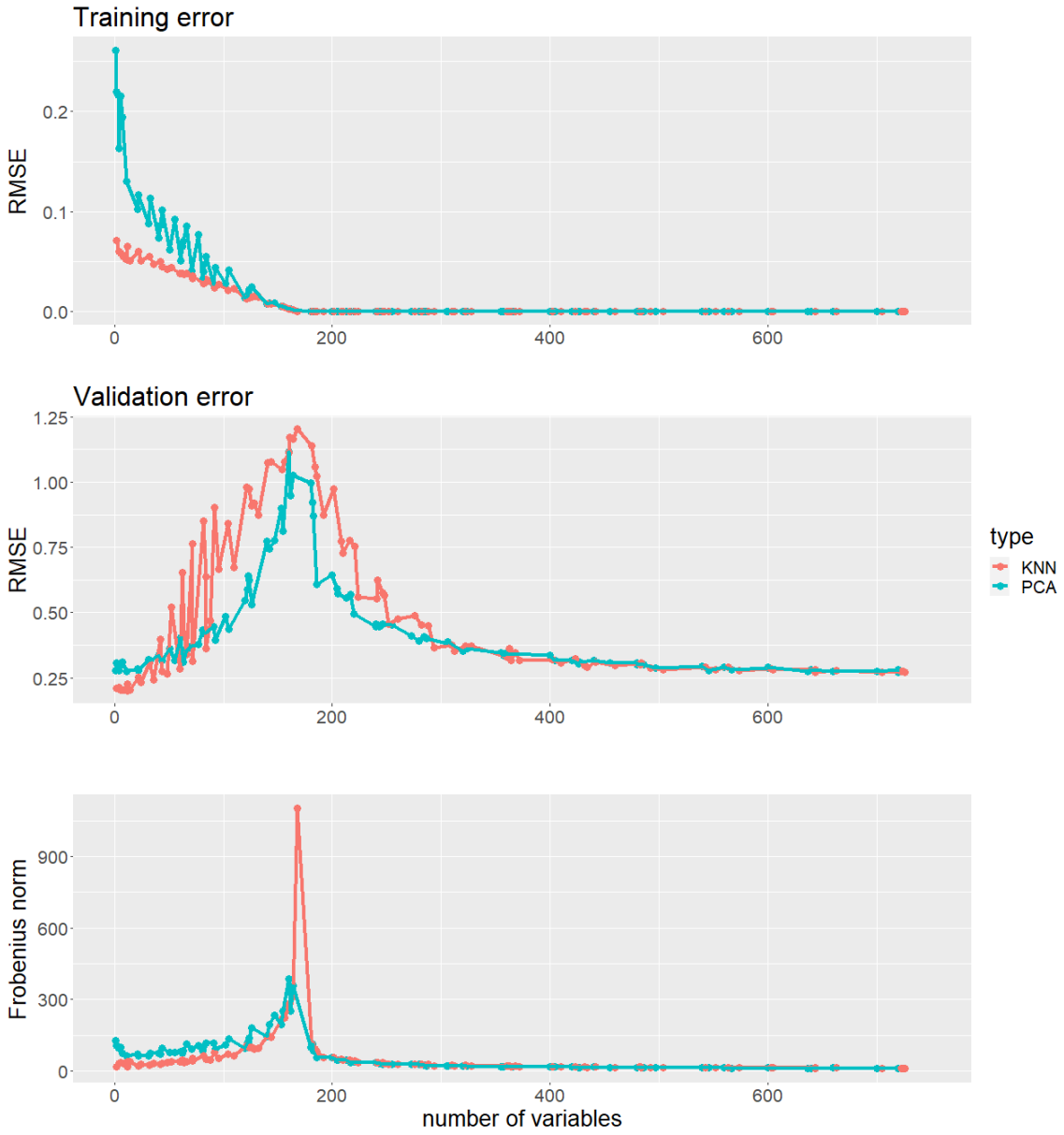


Figure 7.6: The RMSE of 1-step predictions on the training set, the RMSE of 7-step predictions on the validation set and the Frobenius norm of the coefficient matrix for different number of variables for the PCA and KNN models on the linear data set.

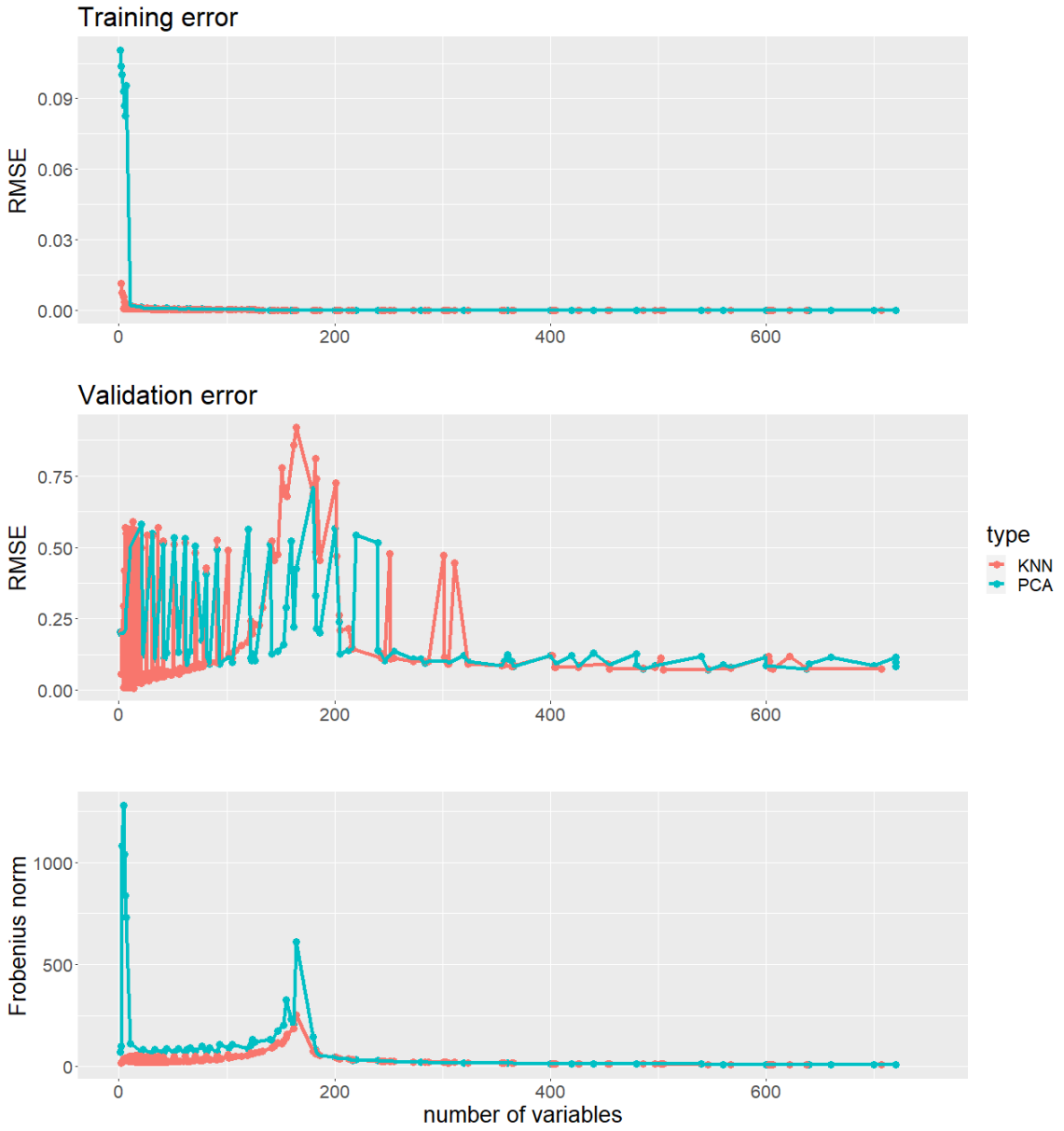
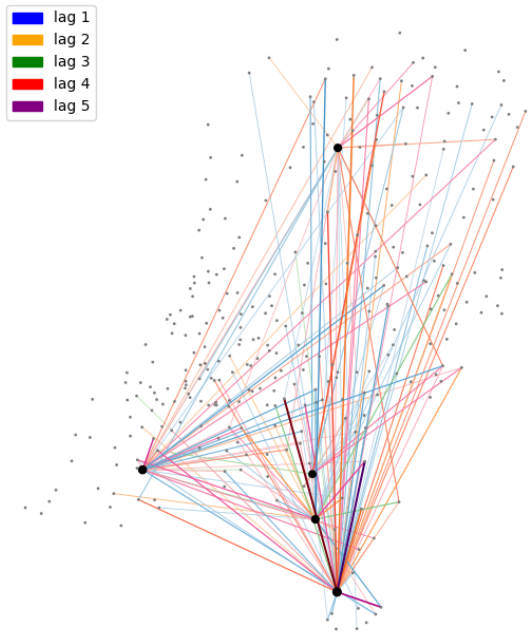
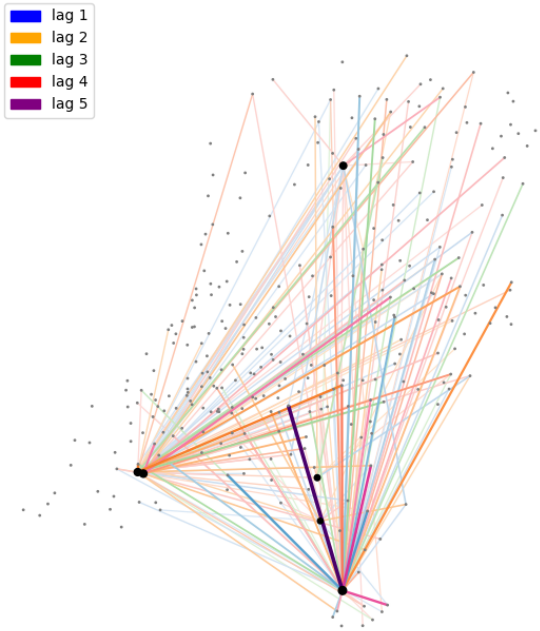


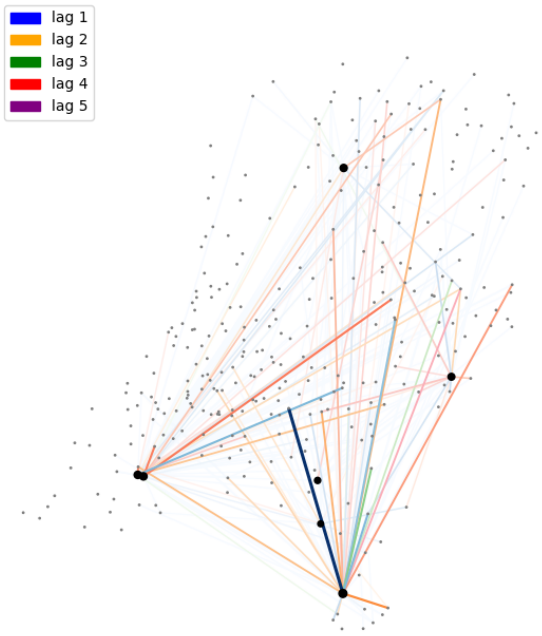
Figure 7.7: The RMSE of 1-step predictions on the training set, the RMSE of 7-step predictions on the validation set and the Frobenius norm of the coefficient matrix for different number of variables for the PCA and KNN models on the modelled data set.



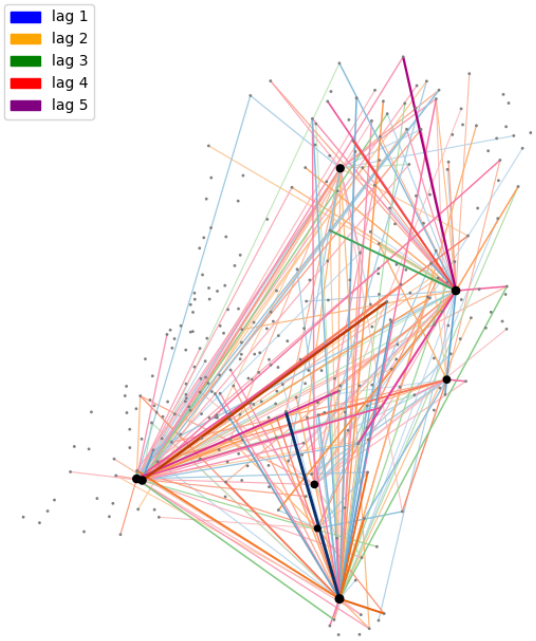
$k = 5$



$k = 6$

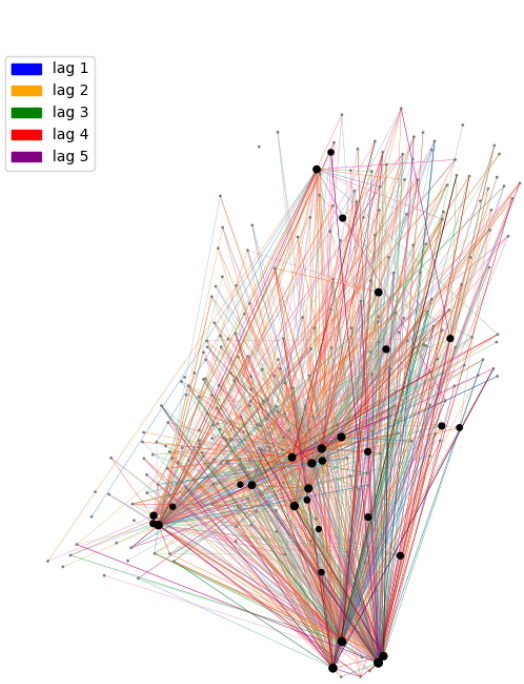


$k = 7$

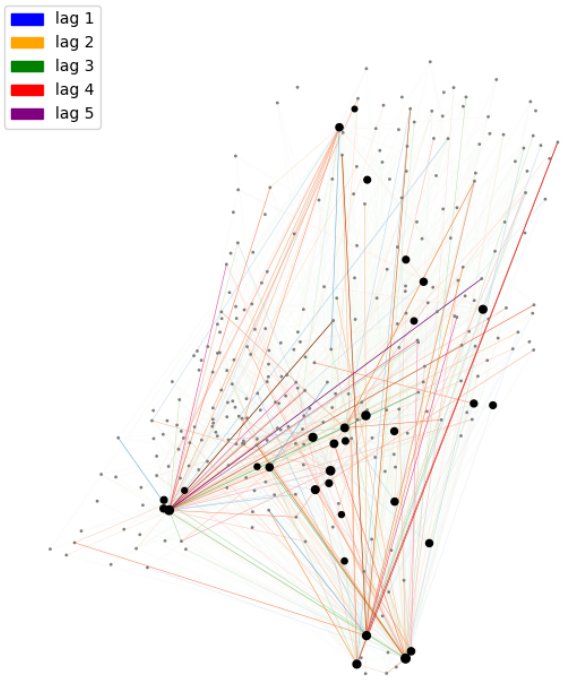


$k = 8$

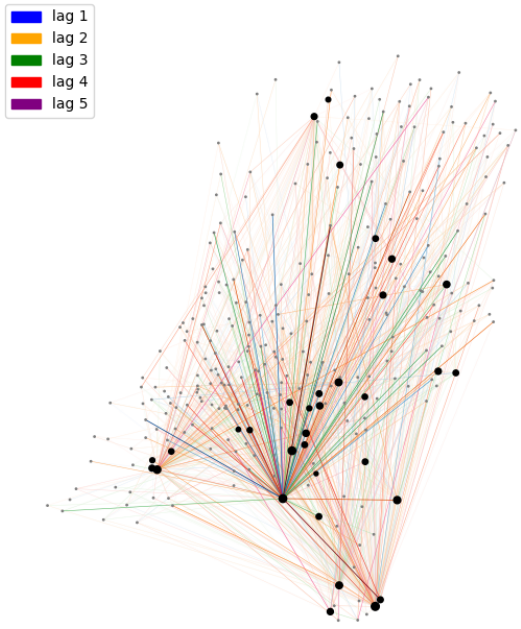
Figure 7.8: Networks from coefficient matrices of PCA for different number of principle variables k.



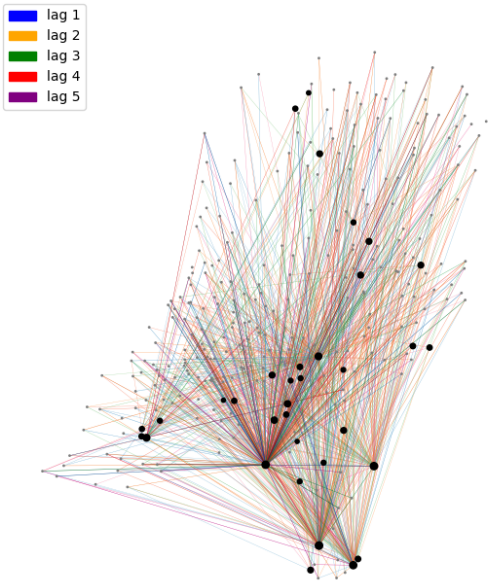
$k = 31$



$k = 32$



$k = 33$



$k = 34$

Figure 7.8: Networks from coefficient matrices of PCA for different number of principle variables k.

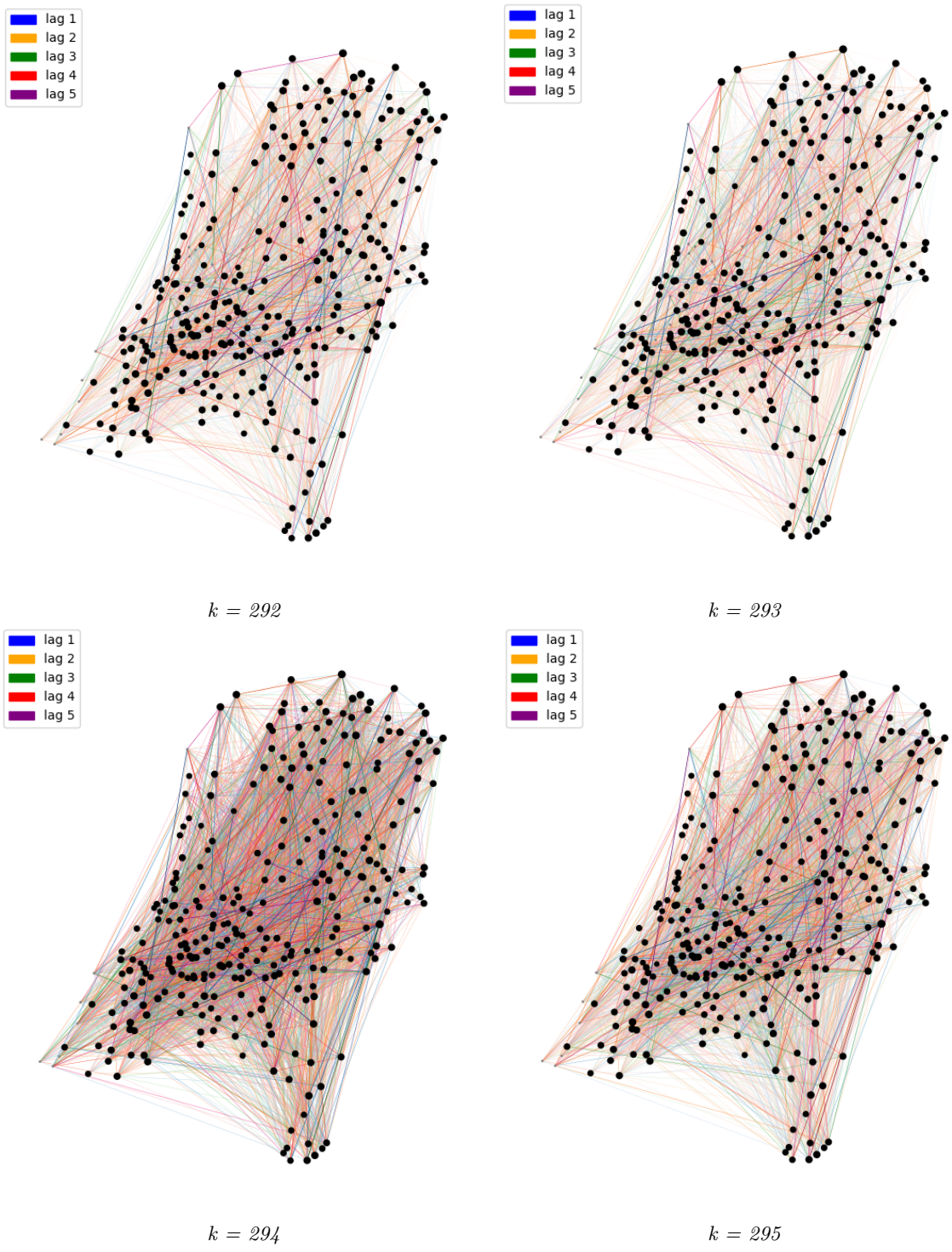


Figure 7.8: Networks from coefficient matrices of PCA for different number of principle variables k.

AR(6)

VAR(7)

KNN(6, 1)

PCA(7, 240)

Figure 7.11: RMSE plotted on a map of the Netherlands. The size of the dots corresponds to the RMSE of the corresponding STP.

AR(6) *VAR(7)*

Figure 7.12: RMSE plotted on a map of the Netherlands. The size of the dots corresponds to the number of inhabitants in the catchment area of the corresponding STP. Dark reds correspond to large values of RMSE, light reds to small values.

Chapter 8

Discussion

In this chapter, we discuss in more detail the results that were found in chapter (7). We showed that for 7-step predictions on a large test set, overall RMSE was lowest for the AR model on the linear data set, and for the VAR model on the modelled data set. However, if outliers were taken out of consideration, adding the nearest neighbour of each variable as predictor variable increases predictive power. We saw that the predictive power decreases as the time between the last training instance and the starting date of a prediction increases. We also showed that adding more observations to the training set does not always result in increased predictive power. Both these observations indicate that (too) old observations should not be included, which could be the result of changing relationships or interactions between STPs and between observations over time. The multivariate models seem to have difficulty in predicting a rise in virus loads after a period of decreasing virus loads. Overall, the models performed better on the modelled data set, with almost each model having 7-steps ahead predictions that lie very close to the actual values. However, on the linear data, the PCA obtains relatively small RMSE values when trained on observations made directly before the prediction date.

We observed remarkable behaviour of the error on the validation set for the number of variables used in the PCA and KNN models, where intermediate values for the number of variables in the model resulted in the worst model performance. Models with intermediate number of variables have coefficient matrices with large norms. These large norms occur because a small number of variables (or columns) are assigned extremely large coefficients. A possible explanation is that coefficient sizes grow to increase the flexibility of the model, thereby enabling the model to fit the noisy training data. We also showed that the coefficient matrices of models with a small or intermediate number of variables change significantly when a variable is added to the model, indicating that for these number of parameter settings, the models are not capable of subtracting the true, underlying spatial relationships of the time series.

On the linear data, predictive power varies much from STP to STP, with smaller errors made for STPs with a large catchment area. This correlation between catchment size and transformed prediction error was found in all models. Time series of STPs which serve a small population are more likely to have larger variations between observations, which makes the prediction task more difficult. Vaughan et al (2023) showed that the performance of Random Forest algorithms is poorer when forecasting virus loads at sites where low or zero values are frequently observed [25]. It would be interesting to explore if low or zero values have a similar effect in the models used in this research.

Both on the linear and the modelled data, the PCA model performs worse than other models. However, when choosing the training set correctly, the PCA model performs well on the modelled data and performs comparable to the other multivariate models on the linear data. This means that when the time series data is sufficiently smooth, the data of only some STPs are needed to predict all 311 STPs. On the modelled data, only 31 principal variables were needed to predict virus loads at the national level well. On the linear data, we have shown that adding more than 60 principal variables to the model, the error on the test set does not decrease significantly. Also, by decreasing the number of training samples and repeating training for each test instance, performance of the PCA model on the linear data was shown to improve considerably. Due to time limitations, the effect of these changes on the other models has not been examined, and so a complete comparison with PCA cannot be made yet. However, we have shown that the PCA variable selection method is a promising tool to reduce the number of variables used in vector autoregression.

The models performed much better on the modelled data than on the linear data. This is not surprising, as the relationship between observations on the modelled data is more straightforward than on the linear data, and thus easier to learn. The linear data contains large variations between measurements. The difference in model performance on the different data sets suggests that more extensive preprocessing on the linear data set could improve model performances. In this research, only moving average smoothing was performed on the linear data set, with a relatively small moving window. Outliers could be detected and replaced before training. Also, a larger window size could be employed.

The implemented models are not equipped to deal with the changing relationships between observations. The models have no information on, for instance, changes in coronavirus restrictions or impacts of dominant virus variants. Also, relationships between catchment areas are likely to change over time, as does population mobility. For instance, during holidays, the population of the Dutch islands grows. Adding mobility data or (dynamic) population sizes of catchment areas to the model could possibly improve results. A recent study by Haak et. al. (2022) found that STPs with the highest peaks in virus loads had higher rates in poverty, lower family incomes and higher population density [11]. Average family incomes and population densities are parameters that could be added to the model as well. However, adding external data to the models can also have a negative impact on model performance. For instance, adding flow data was found to have a negative effect on the prediction of virus loads for the Netherlands [25]. Further research should be done to identify important variables that improve model performance.

It could be that the relationship between STPs is not linear, and that the models will not be able to correctly identify relationships between STPs. In recent years, graph neural networks have proven to be capable in handling relational dependencies between variables. Graph convolutional networks (GCNs) for time series learn graph structures from data, and model both spatial dependencies among variables and temporal dependencies. An advantage of these models is that the underlying graph can change over time. Also, these models are able to capture non-linear relationships by using non-linear activation functions [31]. Recently, GCNs have been applied to estimate COVID-19 case numbers and reproduction rates using wastewater data sets [12].

An advantage of the models used in this thesis is their interpretability. In this research, we have shown that adding a nearest neighbour to the model can improve predictions. Also, we have shown that time series of small STPs are more difficult to predict than time series of large STPs. To the best of our knowledge, the interpretation of the vector autoregression model on a network, and using

properties of that network as feature selection has not been used for multivariate time series forecasting yet. This research introduces new selection methods for VAR based on K-Nearest Neighbours, correlations between (lagged) time series and principal component analysis.

In conclusion, this research has shown that virus load time series can be modelled and forecasted using (multivariate) linear regression models. For time series where variations between observations are small, using multivariate autoregressive models improved predictive power for 7-step predictions compared to autoregressive models. For time series with large variations between observations, results should be explored in more depth, but adding one nearest neighbour as predictor of a variable was shown to improve predictive power. We have shown how a set of principle variables can be selected from a data matrix, such that only a subset of all variables are used in the multivariate autoregressive model. This model performs comparably to other autoregressive models on modelled virus load data, and shows promising results on linear interpolated data when training is repeated before each prediction.

Bibliography

[1] Kohei Adachi. *Matrix-based introduction to multivariate data analysis*. Springer, 2016.

[2] G Bonanno Ferraro et al. "A state-of-the-art scoping review on SARS-CoV-2 in sewage focusing on the potential of wastewater surveillance for the monitoring of the COVID-19 pandemic". In: *Food and Environmental Virology* (2021), pp. 1–40.

[3] Michiel van Boven et al. "Modelling patterns of SARS-CoV-2 circulation in the Netherlands, August 2020-February 2022, revealed by a nationwide sewage surveillance program". In: *medRxiv* (2022), pp. 2022–05.

[4] Peter J Brockwell and Richard A Davis. *Introduction to time series and forecasting*. Springer, 2002.

[5] Peter J Brockwell and Richard A Davis. *Time series: theory and methods*. Springer science & business media, 2009.

[6] Andrew F Brouwer et al. "Epidemiology of the silent polio outbreak in Rahat, Israel, based on modeling of environmental surveillance data". In: *Proceedings of the National Academy of Sciences* 115.45 (2018), E10625–E10633.

[7] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[8] Heumann Christian and Schomaker Michael. *Introduction to Statistics and Data Analysis: With Exercises, Solutions and Applications in R*. 2016.

[9] Jianqing Fan, Jinchi Lv, and Lei Qi. "Sparse high-dimensional models in economics". In: *Annu. Rev. Econ.* 3.1 (2011), pp. 291–317.

[10] *Github page*. `https://github.com/fleurslegers1/Research-Project`.

[11] Laura Haak et al. "Spatial and temporal variability and data bias in wastewater surveillance of SARS-CoV-2 in a sewer system". In: *Science of The Total Environment* 805 (2022), p. 150390.

[12] Guangming Jiang et al. "Artificial neural network-based estimation of COVID-19 case numbers and effective reproduction rate using wastewater-based epidemiology". In: *Water research* 218 (2022), p. 118451.

[13] Takayoshi Kitaoka and Harutaka Takahashi. "Improved prediction of new COVID-19 cases using a simple vector autoregressive model: Evidence from seven New York State counties". In: *Biology Methods and Protocols* 8.1 (2023), bpac035.

[14] Fon-Che Liu. *Real analysis*. Oxford University Press, 2016.

[15] S Wayne Martin et al. "Geographical and temporal distribution of human giardiasis in Ontario, Canada". In: *International journal of health geographics* 2.1 (2003), pp. 1–13.

[16] Colleen C Naughton et al. “Show us the data: global COVID-19 wastewater monitoring efforts, equity, and gaps”. In: *MedRXiv* (2021), pp. 2021–03.

[17] Iman Rahimi, Fang Chen, and Amir H Gandomi. “A review on COVID-19 forecasting models”. In: *Neural Computing and Applications* (2021), pp. 1–11.

[18] *RIVM Covid-19 Sewage*. `https://www.rivm.nl/en/covid-19/sewage`. Accessed: 2023-01-27.

[19] Hannah R Safford, Karen Shapiro, and Heather N Bischel. “Wastewater analysis can be a powerful public health tool—if it’s done sensibly”. In: *Proceedings of the National Academy of Sciences* 119.6 (2022), e2119600119.

[20] Ratchaphon Samphutthanon et al. “Spatio-temporal distribution and hotspots of hand, foot and mouth disease (HFMD) in northern Thailand”. In: *International Journal of Environmental Research and Public Health* 11.1 (2014), pp. 312–336.

[21] Igor R Shafarevich and Alexey O Remizov. *Linear algebra and geometry*. Springer Science & Business Media, 2012.

[22] Shimoni Shah et al. “Wastewater surveillance to infer COVID-19 transmission: A systematic review”. In: *Science of The Total Environment* 804 (2022), p. 150060.

[23] A. W. van der Vaart. *Time Series*. 2010.

[24] Remco Van Der Hofstad. *Random graphs and complex networks*. Vol. 43. Cambridge university press, 2016.

[25] Liam Vaughan et al. “An exploration of challenges associated with machine learning for time series forecasting of COVID-19 community spread using wastewater-based epidemiological data”. In: *Science of The Total Environment* 858 (2023), p. 159748. ISSN: 0048-9697. DOI: `https://doi.org/10.1016/j.scitotenv.2022.159748`. URL: `https://www.sciencedirect.com/science/article/pii/S0048969722068486`.

[26] Michel Verleysen and Damien François. “The curse of dimensionality in data mining and time series prediction”. In: *International work-conference on artificial neural networks*. Springer. 2005, pp. 758–770.

[27] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*. Vol. 47. Cambridge university press, 2018.

[28] René Vidal, Yi Ma, and S Sastry. “Generalized principal component analysis”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.12 (2005), pp. 1–15.

[29] Yuanrong Wang and Tomaso Aste. “Sparsification and Filtering for Spatial-temporal GNN in Multivariate Time-series”. In: *arXiv preprint arXiv:2203.03991* (2022).

[30] Christian H Weiß. *An introduction to discrete-valued time series*. John Wiley & Sons, 2018.

[31] Zonghan Wu et al. “Connecting the dots: Multivariate time series forecasting with graph neural networks”. In: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020, pp. 753–763.

[32] Zhi-Hua Zhou. *Machine learning*. Springer Nature, 2021.

[33] Ettore Zuccato et al. “Estimating community drug abuse by wastewater analysis”. In: *Environmental health perspectives* 116.8 (2008), pp. 1027–1032.

Appendices

Appendix A

Implementation

In this appendix, we describe the outline of the programs that were written for this thesis. Most of the coding for this thesis was done in `R`. The programs were written from scratch based on the theory from chapter (2). The visualization of coefficient matrices was done in `Python`, using the `networkX` package. All code is uploaded to [10]. We now give a general overview of the structure of the programs.

The most important program is `VAR Implementation.R`, with which all autoregression models can be executed on all data sets. In algorithm (1), and overview with description of the parameters and the most important steps of the program are given. In figure (A.1), a pipeline of the program is given.

Algorithm 1 Vector Autoregression

Parameters:

`type`	{AR, VAR, PCA, KNN, CORR}	which autoregression model to use.
`data type`	{int_linear, int_poly, int_weekly}	specifies which data set to use,
`log_`	{TRUE, FALSE}	should values be $\log_{10}$-transformed?
`moving_av`	$\mathbb{N}$	specifies the moving-average window.
`self`	{TRUE, FALSE}	should the diagonal of the adjacency matrix be 1?
`n_col`	$\{1, 2, \ldots, 311\}$	number of variables to use in factor models.
`reg_type`	{ordinary, PCA}	should ordinary or PCA-regression be used?.
`start_train`	between 20-03-2020 and 20-10-2022	date of first training instance.
`end_train`	between 20-03-2020 and 20-10-2022	date of last training instance.
`start_pred`	between 20-03-2020 and 20-10-2022	date of first test instance.
`end_pred`	between 20-03-2020 and 20-10-2022	date of last test instance.
`plotting_`	{TRUE, FALSE}	should plots of predictions be made?
`n_sing_val`	$\{1, 2, \ldots, 311\}$	number of principal components to use in regression if `reg_type` is PCA.

Steps:

Load data
if moving_av > 0 **then**
    Perform moving average smoothing
end if
if log_ **then**
    Perform $\log_{10}$-transformation of data set
end if
Create Adjacency Matrix A
Create empty coefficient Matrix W
for all rows in W **do**
    Get vector y with values $x_{row,t}$
    Get vector x with values $y_{i,t-j}$ for all i for which $A_{row,i} = 1$ and all lags $j = 1, \ldots, p$
    Perform linear regression
    Fill row in W with coefficients
end for
for all dates in test set **do**
    Predict `horizon_` days ahead
end for
Calculate RMSE on test set
Plot predictions

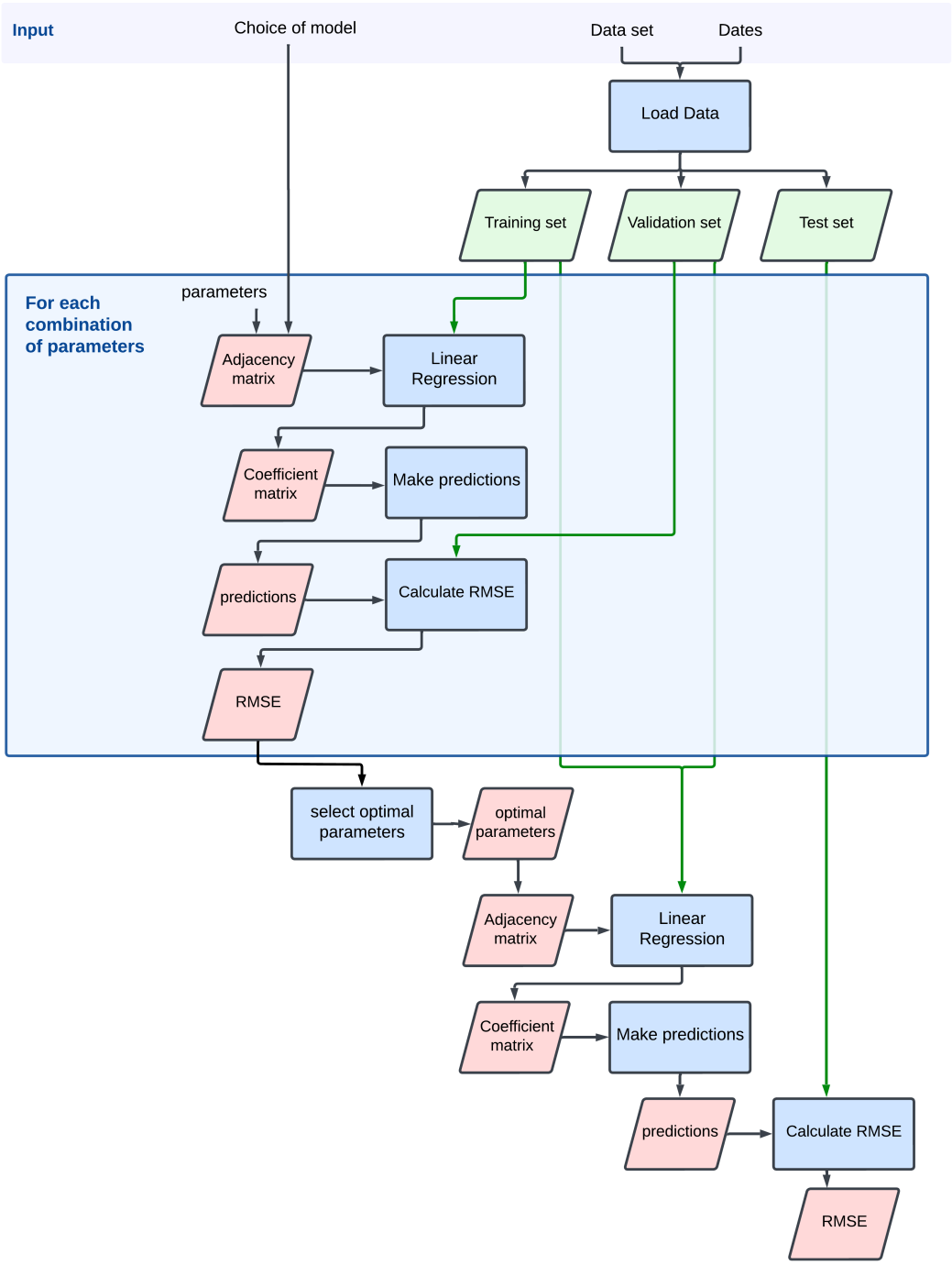


Figure A.1: Flow diagram of the implementation of the models. First, a model and data set are chosen. Then, training, validation and test sets and an adjacency matrix are created. The adjacency matrix is used in linear regression to determine which coefficients are calculated, and which are set to zero. Coefficient matrices are determined and used to make predictions on the validation and test sets. The blue box corresponds to the validation phase, in which training is repeated for multiple parameter combinations.

Appendix B

Network vizualization

The networks of the coefficient matrices in chapter (7) were based on the absolute values of the entries of normalized coefficient matrices. First, we discuss some notions from graph theory on which the network visualization is based. Then, we explain how the coefficient matrices were normalized.

Graph Theory

A *graph* $G = (V, E)$ consists of a set of vertices V and a set of edges E. In a *directed graph*, E contains ordered pairs (u, v) where $u, v \in V$, indicating that there is an edge going from vertex u to vertex v [24].

The *in-degree* $d_{\text{in}}(u)$ and the *out-degree* $d_{\text{out}}(u)$ of a vertex u are the number of edges ending in u and the number of edges originating from u respectively [24]:

$$d_{\text{in}}(u) = \Big|\{v \in V \mid (v, u) \in E\}\Big|, \quad d_{\text{out}}(u) = \Big|\{v \in V \mid (u, v) \in E\}\Big|.$$

A graph is captured by its *adjacency matrix*.

Definition B.1: Adjacency matrix

The **adjacency matrix** A of a directed graph G with n vertices is a $n \times n$-matrix whose entries are

$$A_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E, \\ 0 & \text{if } (i, j) \notin E. \end{cases}$$

Normalization

We assume that the range and mean of observations are (approximately) the same for each variable. For each row in the coefficient matrix, we have scaled the entries so that the sum of the absolute entries in the row is one. In the network, each edge width and color corresponds to the size of the corresponding coefficient in the coefficient matrix. An edge between variables X_i and X_j is assigned a color based on the lag i for which the coefficient size between X_i and $L^i X_j$ is largest. The darkness of the color corresponds to the coefficient size. Only the top 25% largest coefficients are plotted.