Swiss Federal Institute of Technology Zurich

Seminar for Statistics

Department of Mathematics

Master Thesis Summer 2023

Student Muster

Time Series Analysis for Irregularly Sampled Data

Submission Date: 13 March 2023

Co-Advisor	Your co-supervisor
Advisor:	Prof. Dr. Your supervisor

To some special person

Preface

First words and acknowledgements.

Abstract

Short summary of my thesis.

Contents

List of Figures

List of Tables

Notation

BP: blood pressure
CI: Confidence Interval
CRI: Credible Interval

OLS: Ordinary Least Squares
Prediction: TODO
Forcasting: TODO
Filtering: TODO
Smoothing: TODO

$\mathcal{N}(\mu, \sigma^2)$: Normal distribution with mean μ and standard deviation σ

$|M|$: determinant of matrix M

Chapter 1

Introduction

1.1 Thesis Objective

The thesis aims at giving an overview of time series analysis methods for irregularly sampled data.

The standard time series analysis methods usually assume discrete equispaced time and introductory textbooks on time series analysis either completely omit the irregularly spaced case or they only dedicate a very small section to continuous time models or to state-space models with missing observations (Brockwell and Davis, Brockwell and Davis, Cryer and Chan, Chatfield).

I will thereafter present the most important concepts and what I have identified to be the basic methods for the analysis of irregularly spaced time series.

The topic is motivated by a ”real world” problem from medicine. The problem at hand is the one of extracting time series characteristics from a dataset featuring blood pressure (BP) measurements sampled at irregularly spaced time points. High BP is known to be a risk factor for cardiovascular disease. A person’s BP level is generally summarized using the average BP value over available measurements within a given time range. A novel monitoring device already allows to collect BP estimates round the clock. The device is collecting photoplethysmography (PPG) signals and converting them into BP measurements. Typically, the system will yield approximately 1.5 BP measurements per hour, but depending on the quality of the PPG signal and some additional external factors, this sampling frequency can widely vary and the expected range lies roughly between 0 and 5 measurements per hour. Having good estimates of the true BP values at any, potentially not observed, time would allow for a better estimation of the person’s cardiovascular risk, and enable the development of novel valuable metrics. The thesis will focus on a set of time series characteristics, which have been considered most relevant for estimating the person’s cardiovascular risk. The characteristics of interest are:

- the mean function of the BP time series
- the one-week mean BP value
- any ”long-term” trends
- characteristics of the circadian cycle, such as the mean day and night BP

Besides the point estimates also their CIs are of interest. Importantly, the CI should be able to capture the uncertainty due to the lack of data in the proximity of the point of prediction. This implies, that the width of the CI intervals around the mean function will not be constant over time but depend, among other factors, on how much data is available in the proximity of a given time point. The described endpoints are all based on prediction at the not observed passed time points however not on forcasting at new time points in the future. Hence, the thesis will only focus on the task of reconstructing BP values between the first and last time point in the dataset.

This ”real world” problem will serve as a running example throughout the Thesis. Although the topic is motivated by a real dataset we will restrict ourselves to simulated data, which will mimic the most important characteristics of BP time series data.

1.2 Thesis Outline

TODO

Chapter 2

Characteristics of Time Series

2.1 Time Series Definition

A potentially unevenly spaced **time series** is a sequence of observation time and value pair (t_i, x_i) with strictly increasing observation times. Let $\mathbb{T}$ be a set of observation time points, then the sequence of random variables $(X_t : t \in \mathbb{T})$ or simply (X_t) is a **time series process** with observation times $t \in \mathbb{T}$. More specifically:

- $(X_t : t \in \{1, 2, \dots n\})$ refers to a discrete and equispaced time series of length n
- $(X_{t_i} : i \in \{1, 2, \dots n\})$ refers to an irregularly spaced time series of length n with observations at time points $t_1 < t_2 < \dots t_n$
- $(X_t : t \in (0, T])$ refers to a continuous time series

When $\mathbb{T}$ has finite length, we will often use a random column vector $\mathbf{X}$ to refer to the time series process (X_t). Sometimes a time series model will be expressed as a random function $f : \mathbb{T} \to \mathbb{R}$ instead of a collection of random variables. Throughout the thesis, the term time series is used both to refer to the data (x_t) and the process (X_t) from which it is generated.

2.2 Moments of a Time Series

A time series process (X_t) is usually characterized by its first and second moment.

Definition 2.2.0.1. *(Brockwell and Davis) The **mean function** of a time series (X_t) is:*

$$\mu_X(t) = E(X_t)$$

*The **covariance function** of a time series (X_t) is:*

$$\gamma_X(r, s) = Cov(X_r, X_s) = E[(X_r - \mu_X(r))(X_s - \mu_X(s))]$$

2.3 Stationarity

Given that one has only one observation x_t per time point t, a necessary condition to statistically learn from a time series is stationarity.

Definition 2.3.0.1. *(Brockwell and Davis) A time series (X_t) is strictly stationary iff the distribution of $(X_{t_1}, \dots X_{t_n})$ is identical to the distribution of $(X_{t_{1+h}}, \dots, X_{t_{n+h}})$ for all $n \in \mathbb{N}^+$ and shifts $h \in \mathbb{Z}$:*

Definition 2.3.0.2. *(Brockwell and Davis) A time series (X_t) is weakly stationary if*

$$\mu_X(t) \textit{ is independent of } t,$$

and

$$\gamma_X(t+h, t) \textit{ is indpendent of } t \textit{ for each } h$$

Whenever the term stationary is used, it is referring to weak stationarity.

2.4 Special cases of Time Series Processes

Example 2.4.0.1. If (X_t) is a **white noise** process, then $X_t \sim WN(0, \sigma^2)$, that is $X_t \sim F$ iid for some distribution F with mean 0 and varaince σ^2. A special case is Gaussian White noise where $W_t \sim \mathcal{N}(0, \sigma^2)$ and $F = \Phi$

Example 2.4.0.2. An equispaced time series process $(X_t : t \in \{1, 2, \dots\})$ is called **autoregressive process** of order p or AR(p) if:

$$X_t = \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} + W_t$$

where $\phi_p \neq 0$ and (W_t) is a white noise process. The variable W_t is called the innovation at time t and is independent of all X_k, $k < t$.

Example 2.4.0.3. An equispaced time series process $(X_t : t \in \{1, 2, \dots\})$ is called **moving average process** of order q or MA(q) if:

$$X_t = W_t + \theta_1 W_{t-1} + \dots \theta_q W_{t-q}$$

where $\theta_q \neq 0$ and (W_t) is a white noise process. The variable W_t is called the innovation at time t and is independent of all X_k, $k < t$.

Example 2.4.0.4. An equispaced time series process $(X_t : t \in \{1, 2, \dots\})$ is called **autoregressive moving average process** of autoregressive order p and moving average oder q or ARMA(p,q) if:

$$X_t = \phi_1 X_{t-1} + \dots + \phi_p X_{t-p} + \theta_1 W_{t-1} + \dots \theta_q W_{t-q} + W_t$$

where $\phi_p \neq 0, \theta_q \neq 0$ and (W_t) is a white noise process. The variable W_t is called the innovation at time t and is independent of all X_k, $k < t$

2.5 Characteristics of the Blood Pressure Time Series

TODO circadian cycle

Chapter 3

Time Series Decomposition and Regression

As most time series, the mean function of the BP time series is not constant in time and hence it is not stationary. One can try to decompose the time series $Y(t)$ into a deterministic component, the mean function $\mu(t)$ and a zero mean stationary process $E(t)$. This can be expressed in the form of a regression problem:

$$Y(t) = \mu(t) + E(t)$$

The decomposition allows to extract a stationary component $E(t)$, for which we can find a probabilistic model using the theory of such stationary time series processes. The idea is to then use this model in combination with an estimate of $\mu(t)$ to obtain a probability distribution of Y^* at some time t^*. Hence time series decomposition comes for free in regression analysis and we start with estimation of the deterministic component $\mu(t)$ which might be an arbitrary function of t.

3.1 Linear Regression

Based on the knowledge we have about the system we might restrict ourselves to a family of functions for $\mu(t)$. An obvious choice for the BP time series is the family of functions featuring a linear trend with an additive seasonal component. If the seasonal component is represented by a cosine of the form $\alpha\cos(2\pi ft-\phi)$ with phase shift ϕ and known frequency f, we get the following model for the BP time series $Y(t)$:

$$Y(t) = \beta_0 + \beta_1 t + \beta_2 \cos(2\pi ft) + \beta_3 \sin(2\pi ft) + E(t),$$

where based on the trigonometric angle sum identities we know that $\beta_2 = \alpha\cos(\phi)$ and $\beta_3 = \alpha\sin(\phi)$.

If we assume BP observations at potentially unequally spaced time points $t_1, t_2 \ldots t_n$ and $t_1 < t_2 < \ldots t_n$, we can write in matrix notation:

$$\mathbf{Y} = X\beta + \mathbf{E}$$

Where $\mathbf{Y} = [Y_{t_1}, \dots Y_{t_n}]^\top$ is the observed time series, $X = [x_{t_1}, \dots x_{t_n}]^\top \in \mathbb{R}^{n\times 4}$ is the design matrix with i-th row, written as a column vector $x_{t_i} = [1, t_i, \cos(2\pi f t_i), \sin(2\pi f t_i)]^\top$ and $\mathbf{E} = [E_{t_1}, \dots E_{t_n}]^\top$ the zero-mean stationary time series, which we will call errors.

We can use ordinary least squares to find unbiased and asymptotically normal estimates $\hat{\beta}_{OLS} = (X^\top X)^{-1} X^\top Y$ for the regression coefficients β, without the requirement of regularly spaced data points or uncorrelated errors $E_{t_1}, \dots, E_{t_n}$ (White). In the case of uncorrelated errors with constant variance σ^2 we have $Var(\mathbf{E}) = \sigma^2 I_n$ and an unbiased and consistent estimator for $\Psi = Var(\hat{\beta}_{OLS})$ is given by:

$$\hat{\Psi} = \hat{\sigma}^2 (X^\top X)^{-1}$$

$$\text{where } \hat{\sigma}^2 = \frac{1}{n-p} \sum_{i=1}^{n} (y_{t_i} - x_{t_i}^\top \hat{\beta}_{OLS}) \text{ and } p = 4 \text{ in our example}$$

Since $\mathbf{E}$ is a time series, the assumption of uncorrelated errors is usually violated and the covariance matrix $\hat{\Psi}$ is thus no longer unbiased (Brockwell and Davis).

3.2 Regression with Correlated Errors

The argument presented in this section is based on the textbook of Brockwell and Davis.

If the covariance matrix of the errors $Var(\mathbf{E}) = \Sigma$ is known, we can use generalized least squares to obtain a unbiased, consistent and efficient coefficient estimate:

$$\hat{\beta}_{GLS} = (X^\top \Sigma^{-1} X)^{-1} X^\top \Sigma^{-1} Y$$

with unbiased and consistent covariance matrix estimate:

$$Var(\hat{\beta}_{GLS}) = (X^\top \Sigma^{-1} X)^{-1}$$

If Σ is unknown one can exploit the knowledge we have about the stationary time series process $\mathbf{E}$ to estimate it. The following subsections will present two approaches to estimate Σ, β and its covariance matrix. Both methods assume an ARMA(p,q) process for $\mathbf{E}$ and equispaced time points, hence $\mathbf{E} = (E_t : t \in \{1, 2, \dots n\})$ and:

$$\Phi(B) E_t = \Theta(B) W_t, \text{where } W_t \sim WN(0, \sigma_w^2)$$

3.2.1 Maximum-Likelihood Estimation

If we additionally assume $W_t \sim N(0, \sigma_w^2)$, we can simultaneously estimate the regression coefficients and Σ by maximizing the Gaussian likelihood:

$$L(\beta, \phi, \theta, \sigma_w^2) = (2\pi)^{-\frac{n}{2}} |\Sigma_n|^{-\frac{1}{2}} \exp(-\frac{1}{2} (\mathbf{Y} - X\beta)^\top \Sigma_n^{-1} (\mathbf{Y} - X\beta))$$

Where the covariance matrix $\Sigma_n(\theta, \phi, \sigma_w^2)$ is parametrized by the coefficients θ, ϕ, σ_w^2, which define the ARMA process assumed for $(E_t : t \in \{1, 2, \dots n\})$. Assuming an ARMA(2,3) process we can implement this approach in R using the nlme library (Box, Jenkins, and Reinsel) :

```
library(nlme)
cs <- corARMA(from = ~t, p=2, q=3)
fit.gls <- gls(y ~ t + cos(2 * pi * f * t) + sin(2 * pi * f * t), corr=cs)
```

3.2.2 Sandwich Estimation

The second approach is to fit an OLS regression first and correct the estimated covariance matrix of the regression coefficients Ψ with a sandwich estimator. In the presence of autocorrelation one usually estimates $\Phi = \frac{1}{n}X^\top \Sigma X$, the covariance matrix of the scores or estimating functions $V_i(\beta) = x_{t_i}(y_{t_i} - x_{t_i}^\top \beta)$, which can then be used to derive Ψ:

$$\Psi = Var(\hat{\beta}_{OLS}) = (X^\top X)^{-1} X^\top \Sigma X (X^\top X)^{-1} = (\frac{1}{n}X^\top X)^{-1} \frac{1}{n} \Phi (\frac{1}{n}X^\top X)^{-1} \tag{3.2.2.1}$$

The general form of the estimators for Φ is:

$$\hat{\Phi} = \frac{1}{n} \sum_{i,j=1}^{n} w_{|i-j|} \hat{V_i} \hat{V_j}^\top \tag{3.2.2.2}$$

where $w = [w_0, \dots w_{n-1}]^\top$ is a weight vector and $\hat{V_i} = V_i(\hat{\beta}_{OLS})$.

Plugging $\hat{\Phi}$ into the equation 3.2.2.1 one obtains the heteroskedasticity and autocorrelation consistent (HAC) covariance estimate $\hat{\Psi}_{HAC}$.

Newey and West, Andrews and others have suggested different approaches for calculating the weights w. They all yield decreasing weights with increasing lag $l = |i - j|$. The R sandwich package implements some of these methods to estimate $\hat{\Psi}_{HAC}$. An introduction to the sandwich package and how it can be used for inference is described by Zeileis.

3.2.3 Extension to Irregularly Spaced Time Series

Although literature and ”ready to use” implementations only exist for the equispaced case, both of the approaches described above could probably be extended to the case of irregularly spaced time series. For the Maximum-Likelihood approach the parametrization of the covariance matrix Σ_n as described in 3.2.1 would need to be adapted, such that the covariance of the errors at different time points depends on the actual time difference rather than the lag. Similarly for the sandwich estimator, the weights in 3.2.2.2 should depend on the time difference rather than on the lag.

3.2.4 Confidence Intervals for the Mean Function

The objective, as described in the introduction, is not only to estimate the mean function $\mu(t)$ of the time BP time series but also to find confidence intervals for it. The model for the BP time series described in 3.1 has the following mean function:

$$\mu(t) = x_t^\top \beta$$
$$\text{with } x_t = [1, t, \cos(2\pi f t), \sin(2\pi f t)]^\top$$

Hence, we may also write $\mu(x_t)$ and its $1-\alpha$ confidence interval is:

$$x_t^\top \hat{\beta} \pm qt_{n-p}(1-\frac{\alpha}{2})\sqrt{x_t^\top \Psi x_t}$$

where $\Psi = Var(\hat{\beta})$ is the covariance matrix of the estimated regression coefficients and $qt_{n-p}(1-\frac{\alpha}{2})$ denotes the $1-\frac{\alpha}{2}$ quantile of the student's t-distribution of $n-p$ degrees of freedom.

As the CI for $\mu(t)$ is based on the variance of the estimated global model parameters Ψ, it cannot adapt to the local observation density. Even if we were able to derive realistic confidence interval for the mean function of the irregularly spaced time series, the uncertainty due to the lack of data in the proximity of a time point can still not be reflected.

TODO: Prediction interval $1-\alpha$ prediction interval is:

$$x_t^\top \hat{\beta} \pm qt_{n-p}(1-\frac{\alpha}{2})\sqrt{\sigma^2 + x_t^\top \Psi x_t}$$

with $\sigma^2 = \Sigma_{11}$

Chapter 4

Gaussian Process Regression

The objective of regression is generally to establish a mapping between the input variable x and its corresponding output $f(x)$. In order to solve such a problem one usually needs some additional constraints on $f(x)$. In chapter 3 we restricted ourselves to the class of linear functions. However, an alternative approach is to assign prior probabilities to all possible functions, giving higher probabilities to those considered more plausible. In this Bayesian framework, inference revolves around the posterior distribution of these functions, given some potentially noisy observations of $f(x)$.

This chapter begins by providing a formal definition of a Gaussian Process and subsequently explores its application in solving regression problems. The arguments presented in this chapter are based on the textbook of Rasmussen and Williams.

4.1 Gaussian Process Definition

A Gaussian process (GP) can be viewed as a gaussian distribution over functions or as an infinite set of random variables representing the values of the function $f(x)$ at location x. The Gaussian process is thus a generalization of the Gaussian distribution and a formal definition is given by Rasmussen and Williams :

Definition 4.1.0.1 (Gaussian Process)**.** *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

As a (multivariate) Gaussian distribution is defined by its mean and covariance matrix, a GP is uniquely identified by its mean $m(x)$ and covariance (kernel) function $k(x, x')$.

We write

$$f(x) \sim GP(m(x), k(x, x'))$$

with

$$m(x) = \mathbb{E}[f(x)]$$
$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

If we assume X to be the index set or set of possible inputs of f, then there is a random variable $F_x := f(x)$ such that for a set $A \subset X$ with $A = x_1, \ldots x_n$ it holds that:

$$F_A = [F_{x_1}, \dots, F_{x_n}] \sim \mathcal{N}(\mu_A,\, K_{AA})$$

for

$$K_{AA} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ \vdots & & \vdots & \vdots \\ k(x_n, x_1) & k(x_n, x_1) & \dots & k(x_n, x_n) \end{bmatrix} \text{ and } \mu_A = \begin{bmatrix} \mu(x_1) \\ \vdots \\ \mu(x_n) \end{bmatrix} \tag{4.1.0.1}$$

The finite marginals $F_{x_1}, \dots, F_{x_n}$ of the GP thus have a multivariate gaussian distribution. In our running example we might consider X to be the time interval $T_0 = [0, T]$ however it could be higher dimensional.

Note that a GP with finite index set and hence with joint gaussian distribution is just a specific case of GP. If we assume an ARMA process with gaussian innovations for the blood pressure time series, one can view the time series as a collection of multivariate normally distributed random variables and thus as a GP.

If we consider the linear regression case from chapter 3 and assume a prior distribution on β, i.e. $\beta \sim N(0, I)$ then the predictive distribution over $\mu = X\beta$ is Gaussian:

$$\mu \sim \mathcal{N}(0, XX^\top)$$

This is equivalent to a GP with mean function $m(x) = 0$ and kernel function $k(x, x') = x^\top x'$. This special case of gaussian process regression with this specific kernel function is known as Bayesian linear regression and will be presented in the next section.

4.2 Bayesian Linear Regression

Predictions in the Bayesian regression setting is finding the posterior distribution of $f^* := f(x^*)$ at some input x^*, given some potentially noisy observations of $f(x)$. This is made possible by employing a prior distribution over the function $f(x)$. As shown in section 4.1, a GP is essentially assuming a Gaussian distribution over functions. This section however still stays in the domain of parametric models, in which case we assume a distribution over the parameters of the function $f(x)$, rather than over the function itself. In Bayesian linear regression we are thus assuming a distribution over the regression coefficients β.

Recall the linear regression model from chapter 3. However, we are assuming a more general setting, where the data generating process does not need to be a time series process. The function is denoted with $f(x)$ instead of $\mu(t)$ and Y_i is again a noisy observation of $f(x_i)$, where the additive error E_i does not necessarily need to be from a time series process $(E_t : t \in \{t_1, t_2, \dots t_n\})$. We obtain the following data generating model:

$$f(x_i) = x_i^\top \beta, \qquad Y_i = f(x_i) + E_i, \qquad (i = 1, \dots n)$$

with $x_i \in \mathbb{R}^p$ being again the input vector and $\beta \in \mathbb{R}^p$ is the vector with the regression coefficients.

In matrix from:

$$\mathbf{Y} = X\beta + \mathbf{E}$$

Where $\mathbf{Y} = [Y_1, \dots Y_n]^\top$ is the observed data, $X = [x_1, \dots x_n]^\top \in \mathbb{R}^{n \times p}$ is the design matrix. We assume again gaussian but potentially correlated errors $\mathbf{E} = [E_1, \dots E_n]^\top$:

$$\mathbf{E} \sim \mathcal{N}(0, \Sigma_e)$$

If $\mathbf{E}$ is an ARMA process, then every element of the time series E_i is itself a sum of innovations. Therefore, $\mathbf{E}$ is gaussian as long as it has gaussian innovations.

The likelihood, i.e. the probability of the observations $\mathbf{Y}$ given X and β is then:

$$p(\mathbf{Y}|X, \beta) = \frac{1}{(2\pi)^{n/2}\sqrt{\det(\Sigma_e)}} \exp(-\frac{1}{2}(y - X\beta)^\top \Sigma_e^{-1}(y - X\beta)) = \mathcal{N}(X\beta, \Sigma_e)$$

Until now the regression model is exactly the same as in chapter 3. The Bayesian approach is different in that we additionally assume a prior distribution over the regression coefficients β, based on what we believe are likely values for the coefficients. To stay in the realm of gaussian processes the prior has to be Gaussian and we choose:

$$p(\beta) = \mathcal{N}(0, \Sigma_p)$$

Note how the function $f(x_i) = x_i^\top \beta$ is now no longer deterministic but a random function.

Given our observations $\mathbf{Y}$ we can use Bayes' theorem to calculate the posterior distribution over β:

$$p(\beta|\mathbf{Y}, X) = \frac{p(\mathbf{Y}, \beta|X)}{p(\mathbf{Y}|X)} = \frac{p(\mathbf{Y}|X, \beta)p(\beta)}{p(\mathbf{Y}|X)}$$

One approach is to just plug in the expressions for $p(\mathbf{Y}|X, \beta)$ and $p(\beta|\mathbf{Y}, X)$ from above, with the marginal likelihood:

$$p(\mathbf{Y}|X) = \int p(\mathbf{Y}|X, \beta)p(\beta)d\beta = \mathcal{N}(0, X\Sigma_p X^\top + \Sigma_e) \tag{4.2.0.1}$$

The term marginal likelihood arises from the marginalization over the parameter values β.

Or it can be helpful to combine the coefficients and the observations into a single random vector with multivariate normal distribution:

$$\begin{bmatrix} \mathbf{Y} \\ \beta \end{bmatrix} = \begin{bmatrix} X \\ I_p \end{bmatrix} \beta + \begin{bmatrix} I_n \\ 0 \end{bmatrix} \mathbf{E} \sim \mathcal{N} \left(\begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \left[\begin{array}{c:c} X\Sigma_p X^\top + \Sigma_e & X\Sigma_p \\ \hdashline \Sigma_p X^\top & \Sigma_p \end{array} \right] \right) = p(\mathbf{Y}, \beta|X) \tag{4.2.0.2}$$

with $\Sigma_p X^\top + \Sigma_e \in \mathbb{R}^{n\times n}$ and $\Sigma_p X^\top \in \mathbb{R}^{p\times n}$.

To find now the posterior distribution $p(\beta|\mathbf{Y}, X)$ one can use the rules for deriving conditional distributions for multivariate Gaussian's presented in theorem 4.2.0.1.

Theorem 4.2.0.1. *(von Mises)*

Let $A \sim \mathcal{N}(\mu_A, \Sigma_{AA})$ and $B \sim \mathcal{N}(\mu_B, \Sigma_{BB})$ be Gaussian random vectors with the following joint distribution:

$$p(A, B) = \mathcal{N}\left(\begin{bmatrix}\mu_A \\ \mu_B\end{bmatrix}, \begin{bmatrix}\Sigma_{AA} & \Sigma_{AB} \\ \Sigma_{BA} & \Sigma_{BB}\end{bmatrix}\right)$$

Then the conditional distribution $p(\mathbf{B}|\mathbf{A} = a)$ is also normally distributed with mean $\bar{\mu}$ and covariance $\bar{\Sigma}$ of the following form:

$$\bar{\Sigma} = \Sigma_{BB} - \Sigma_{BA}\Sigma_{AA}^{-1}\Sigma_{AB} \qquad\qquad \bar{\mu} = \mu_B + \Sigma_{BA}\Sigma_{AA}^{-1}(a - \mu_A)$$

Using theorem 4.2.0.1 the posterior distribution over β is then given by:

$$\begin{aligned} p(\beta|\mathbf{Y} = y, X) &\sim \mathcal{N}(\bar{\mu}, \bar{\Sigma}), \\ \bar{\Sigma} &= \Sigma_p - \Sigma_p X^\top (X\Sigma_p X^\top + \Sigma_e)^{-1} X\Sigma_p, \\ \bar{\mu} &= \mu_\beta + \Sigma_p X^\top (X\Sigma_p X^\top + \Sigma_e)^{-1} y \end{aligned}$$

The expression for the posterior mean and covariance matrix can be further simplified using Woodbury matrix identity and we obtain:

$$\bar{\Sigma} = (X^\top \Sigma_e^{-1} X + \Sigma_p^{-1})^{-1} \qquad\qquad \bar{\mu} = \bar{\Sigma} X^\top \Sigma_e^{-1} y \tag{4.2.0.3}$$

Since $f(x) = x^\top \beta$, one can use the posterior mean and covariance matrix from 4.2.0.3 to obtain the predictive distribution of $f^* := f(x^*)$ at x^* given our observations:

$$p(f^*|\mathbf{Y}, X, x^*) = \mathcal{N}({x^*}^\top \bar{\mu}, {x^*}^\top \bar{\Sigma} x^*) \tag{4.2.0.4}$$

One can also use the rules for conditioning to directly derive $f^*|\mathbf{Y}, X, x^*$. Similar to before we can write the joint distribution $p(\mathbf{Y}, f^*|X, x^*)$:

$$\begin{bmatrix}\mathbf{Y} \\ f^*\end{bmatrix} = \begin{bmatrix}X \\ x^*\end{bmatrix}\beta + \begin{bmatrix}I_n \\ 0\end{bmatrix}\mathbf{E} \sim \mathcal{N}\left(\begin{bmatrix}0 \\ \vdots \\ \vdots \\ 0 \\ 0\end{bmatrix}, \left[\begin{array}{c:c} X\Sigma_p X^\top + \Sigma_e & X\Sigma_p x^* \\ \hdashline {x^*}^\top \Sigma_p X^\top & \Sigma_p \end{array}\right]\right) = p(\mathbf{Y}, f^*|X, x^*) \tag{4.2.0.5}$$

The expression in 4.2.0.4 can then be derived using theorem 4.2.0.1 on conditioning of multivariate Gaussian's.

The next section will extend the Bayesian approach to non-parametric models and illustrate how Bayesian linear regression is just a special case of GP regression.

4.3 Bayesian Linear Regression as Gaussian Process Regression

The linear model discussed so far, with a cyclic component represented by a cosine and a linear trend component, might be an evident first guess. However, it is unlikely that the BP values are exactly following this pattern. Instead of reducing the function space to this specific class of linear functions, we may use our domain knowledge to tell which functions of the infinite space of all functions are more likely to have generated our data. As these functions are not characterized with explicit sets of parameters, this approach belongs to the branch of non-parametric modelling. By abandoning the parameters β, Gaussian process regression directly aims for the predictive distribution of $f^* := f(x^*)$ at an input x^* given our observations.

Starting with the Bayesian linear regression example from last section and transforming it into a GP regression problem, we recall that the distribution of $F_X = [f(x_1)\dots f(x_n))]^\top$ with given $X = [x_1 \dots x_n]^\top$ is:

$$F_X \sim \mathcal{N}(0, X\Sigma_p X^\top)$$

Alternatively this can be written as a distribution over the function $f(x)$:

$$f(x) \sim GP(0, k(x, x'))$$

where $k(x, x')$ needs to be chosen such that for an input X we obtain $K_{XX} = X\Sigma_p X^\top$. Given $\Sigma_p = \sigma_p I$, we would choose $k(x, x') = \sigma_p x^\top x'$, with the input pairs x and x' only entering as a dot product.

Combining f^* and $\mathbf{Y}$ into a single random vector we can use the theorem 4.2.0.1 to arrive at the same posterior predictive distribution $p(f^*|\mathbf{Y}, X, x^*)$ as presented in 4.2.0.4. The joint distribution of f^* and $\mathbf{Y}$ can be expressed as follows:

$$\begin{bmatrix} \mathbf{Y} \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{XX} + \Sigma_e & K_{Xx^*} \\ K_{x^*X} & K_{x^*x^*} \end{bmatrix} \right) = p(\mathbf{Y}, f^*|X, x^*) \tag{4.3.0.1}$$

where:

$$K_{XX} = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \dots & k(x_1, x_n) \\ \vdots & & \vdots & \vdots \\ k(x_n, x_1) & k(x_n, x_1) & \dots & k(x_n, x_n) \end{bmatrix},$$

$$K_{Xx^*} = K_{x^*X}^\top = \begin{bmatrix} k(x_1, x^*) \\ \vdots \\ k(x_n, x^*) \end{bmatrix} \text{ and } K_{x^*x^*} = k(x^*x^*)$$

4.3.1 Time Series Gaussian Process Regression

Unlike in chapter 3, $f(x)$ is no longer assumed to be a deterministic and parametric function. This way, GP regression allows us to treat $\mathbf{E}$ not simply as an error term but an actual part of our signal which we can predict. If $\mathbf{E}$ is not independent noise but for example a time series, where the elements of $\mathbf{E}$ are correlated, we want to leverage the information we have about an unobserved time point given our observations. Hence, we are not interested in the posterior distribution of f^* only, but also of $Y^* := Y(x^*) = f(x^*) + E(x^*)$.

Recall the expression for the marginal likelihood $p(\mathbf{Y}|X)$ from 4.2.0.1:

$$\mathbf{Y}|X \sim \mathcal{N}(0, X\Sigma_p X^\top + \Sigma_e)$$

Alternatively, this can be expressed as a distribution over the function $Y(x)$:

$$Y(x) \sim GP(0, k(x, x'))$$

The kernel function $k(x, x')$ needs to be chosen such that for an index set X we obtain $K_{XX} = X\Sigma_p X^\top + \Sigma_e$. One can then follow again the same procedure as before and combine Y^* and $\mathbf{Y}$ into a single random vector:

$$\begin{bmatrix} \mathbf{Y} \\ Y^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{XX} & K_{Xx^*} \\ K_{x^*X} & K_{x^*x^*} \end{bmatrix} \right) = p(\mathbf{Y}, Y^*|X, x^*) \tag{4.3.1.1}$$

The predictive distribution $p(Y^*|\mathbf{Y}, X, x^*)$ is then again derived by conditioning.

One could also assume additional measurement noise on the time series $f(x) + E(x)$. We then have for the observed time series $Y(x)$:

$$Y(x) = f(x) + E(x) + \epsilon \qquad\qquad \epsilon \sim \mathcal{N}(0, \sigma_n^2)$$

To be inline with the literature on Gaussian process regression, we will from now on consider our goal to find some function $f(x)$, which is a combination of the mean function, until now denoted by $f(x)$, and the stationary time series $E(x)$. The observed time series $Y(x)$ will thus be equivalent to $f(x)$ up to some additive independent noise term ϵ. We can write:

$$Y(x) = f(x) + \epsilon \qquad\qquad \epsilon \sim \mathcal{N}(0, \sigma_n^2)$$

Assuming the same linear model as before, we have for $F_X = [f(x_1), \dots f(x_n)]^\top$:

$$F_X = X\beta + \mathbf{E}, \text{ with } \beta \sim \mathcal{N}(0, \Sigma_p) \text{ and } \mathbf{E} \sim \mathcal{N}(0, \Sigma_e)$$

Analogously we can write:

$$f(x) \sim GP(0, k(x, x')),$$

with $k(x, x')$ such that for an input $X = [x_1 \dots x_n]^\top$ we obtain $K_{XX} = X\Sigma_p X^\top + \Sigma_e$.

The joint distribution of $\mathbf{Y}$ and $f^* := f(x^*)$ is given by:

$$\begin{bmatrix} \mathbf{Y} \\ f^* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} K_{XX} + \sigma_n^2 I & K_{Xx^*} \\ K_{x^*X} & K_{x^*x^*} \end{bmatrix} \right) = p(\mathbf{Y}, f^* | X, x^*) \tag{4.3.1.2}$$

The posterior (or predictive) distribution over f^* can then again be derived by conditioning:

$$p(f^*|\mathbf{Y}, X) = \mathcal{N}(K_{x^*X}(K_{XX} + \sigma_n^2 I)^{-1}\mathbf{Y}, K_{x^*x^*} - K_{x^*X}(K_{XX} + \sigma_n^2 I)^{-1}K_{Xx^*}) \tag{4.3.1.3}$$

If we are interested in predicting $Y(X)$, i.e. the iid gaussian noise term ϵ should be included in the prediction. We choose $k(x, x')$ such that $K_{XX} = X\Sigma_p X^\top + \Sigma_e + \sigma_n^2 I$. The predictive distribution over $Y^* := Y(x^*)$ is then simply:

$$p(Y^*|\mathbf{Y}, X) = \mathcal{N}(K_{x^*X}K_{XX}^{-1}\mathbf{Y}, K_{x^*x^*} - K_{x^*X}K_{XX}^{-1}K_{Xx^*}) \tag{4.3.1.4}$$

Also note how until now we have still assumed Σ_e, the covariance matrix of $\mathbf{E}$, to be known. However, deriving Σ_e for an ARMA process with irregularly spaced samples is not straight forward, as has already been shown in chapter 3. The next section will illustrate how choosing a specific kernel function solves this problem.

4.4 Mean Function

A Gaussian process is fully specified by its mean function, $\mu(x)$, and its covariance function, $k(x, x')$. However, the mean function can always be subtracted from the observed data without changing the covariance structure of the data. In other words, if we subtract the mean function from the observed data, we obtain a new dataset with zero mean, but the same covariance structure.

Incorporating Explicit Basis Functions Rasmussen and Williams p.27

Assuming we want to model $Y(x) = f(x) + \epsilon$, with $f(x) = m(x) + E(x)$ where $m(x)$ is a deterministic mean function and the $E(x)$ is a time series process and ϵ is some iid. gaussian noise term with variance σ_n^2.

Then we can model $f(x)$ with a GP:

$$f(x) \sim GP(m(x), k(x, x'))$$

By using the conditioning rule we arrive at the following predictive distribution for $f^* := f(x^*)$:

$$\begin{gathered} p(f^*|\mathbf{Y} = y, X, x^*) = N(\bar{\mu}, \bar{\Sigma}), \\ \bar{\mu} = m(x^*) + K_{x^*X}(K_{XX} + \sigma^n I)^{-1}(y - m(X)), \\ \bar{\Sigma} = K_{x^*x^*} - K_{x^*X}(K_{XX} + \sigma^n I)^{-1}K(X, x^*) \end{gathered}$$

Note how the predictive variance $\bar{\Sigma}$ is not affected by $m(x)$. If instead a GP is fitted to $f(x) - m(x) = E(x)$ we can write:

$$E(x) = f(x) - m(x) \sim GP(0, k(x, x'))$$

The predictive distribution over $E^* := E(x^*)$ given $\mathbf{Z} := \mathbf{Y} - m(X)$ is then:

$$p(E^*|\mathbf{Z} = z, X, x^*) = N(\bar{\mu}_{E^*}, \bar{\Sigma}),$$
$$\bar{\mu}_{E^*} = K_{x^*X}(K_{XX} + \sigma^n I)^{-1} z,$$

Since $z = y - m(X)$, the predictive distribution over f^* is recovered by adding $m(x^*)$ to the predictive mean $\bar{\mu}_{E^*}$. The predictive variance $\bar{\Sigma}$ remains unchanged, since it is not affected by the observations nor by $m(x)$.

Hence, when having some knowledge about $m(x)$ one should subtract it first before fitting a GP with a zero mean prior.

If unknown one can optimize over the paramters of the mean function and the hyperparameters of the convariance function jointly.

Also in the case of $m(x)$ being a constant c one one can simply add c^2 to the covariance function and have $m(x) = 0$

$f(x) \sim GP(m(x) = 0, k(x_i, x_j))$

that is to say,

$E[f(x)] = 0$, $cov[f(x_i), f(x_j)] = k(x_i, x_j)$

If a constant c is added to the kernel,

$cov[f(x_i), f(x_j)] = E[(f(x_i) - E[f(x_i)])(f(x_j) - E[f(x_i)])] = E[f(x_i)f(x_j)] - m(x_i)m(x_j) = E[f(x_i)f(x_j)] = k(x_i, x_j) + c$

So actually it is the same as a GP with

$sqrt(c), k(x_i, x_j)$

because

$cov[f(x_i), f(x_j)] = E[f(x_i)f(x_j)] - m(x_i)m(x_j) = E[f(x_i)f(x_j)] - c = k(x_i, x_j)$

4.5 Kernel Functions

In the last section we started of with a describing the prior distribution over $\mathbf{Y}$ or $F_X = [f(x_1) \ldots f(x_n))]^\top$ and shoved that a kernel function $k(x, x')$ needs to be chosen to match this distribution. However, in Gaussian process regression it generally goes the other way around. One would choose a prior distribution over $f(x)$ or $Y(x)$ first, which boils down to choosing a kernel function. The kernel function evaluated at your inputs $X = [x_1 \ldots x_n]^\top$ is then needed to calculate the predictive distribution of f^* or y^*.

The choice of kernel function depends on the assumptions about correlation in your output given arbitrary input pairs x and x'.

4.5.1 Stationary Covariance Function

Does only depend on $\tau = x - x'$.

The Matérn Class of Covariance Functions

A expression for the Matérn covariance function is given by Rasmussen and Williams:

$$k_\nu(\tau) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} (\frac{\sqrt{2\nu}\tau}{l})^\nu K_\nu(\frac{\sqrt{2\nu}\tau}{l})$$

where ν and l are positive parameters, K_ν is a modified Bessel function and $\sigma^2 = k_\nu(0)$

For $\nu = r + 1/2, r \in \mathbb{N}$ the expression for the Matérn covariance function can be simplified:

$$k_{\nu=r+1/2}(\tau) = \sigma^2 \exp(-\frac{\sqrt{2r+1}\tau}{l}) \frac{r!}{(2p)!} \sum_{i=0}^{r} \frac{(r+i)!}{i!(r-i)!} (\frac{2\sqrt{2r+1}\tau}{l})^{r-i} \tag{4.5.1.1}$$

Or also with `includegraphics`:

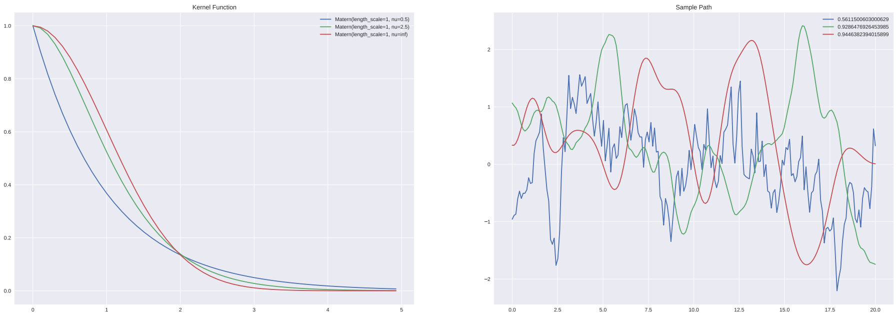

Figure 4.1: Matérn kernel function and sample path for different ν

Setting $\nu = 1/2$ with input domain $X \subset \mathbb{R}$ gives raise to a continuous-time AR(1) process, also called Ornstein-Uhlenbeck process. With $\nu = 1/2$, i.e. $r = 0$, the Matérn covariance function is given by:

$$k(\tau) = \sigma^2 exp(-\tau/l) \tag{4.5.1.2}$$

The autocovariance function of an Ornstein-Uhlenbeck process can be derived by solving the stochastic differential equation (SDE) that defines the process.

Starting with the SDE for an OU process:

$$dX_t = \theta(\mu - X_t)dt + \sigma_w dW_t,$$

where X_t is the value of the process at time t, θ is a positive constant that determines the speed of mean reversion, μ is the long-term mean of the process, σ_w is the standard deviation of the random shocks, and W_t is a standard Wiener process or Brownian motion.

The solution to the SDE is:

$$X_t = X_0 e^{-\theta t} + \mu(1 - e^{-\theta t}) + \sigma_w e^{-\theta t} \int_0^t e^{\theta s} dW_s$$

The process is stationary if $\theta > 0$. The autocovariance function of an OU process is given by $Cov(X_t, X_{t-k}) = \frac{\sigma_w^2}{2\theta} e^{-\theta k}$, where $k \geq 0$ and $\theta > 0$.

This is the same expression as we have obtained in 4.5.1.2, where $k(0) = \sigma^2 = \frac{\sigma_w^2}{2\theta}$ and $l = 1/\theta$

To see how the Ornstein-Uhlenbeck can be considered a continuous time analogue to the discrete time AR(1) process one can use the Euler-Maryuama discretization of the process. Considering again the SDE for an OU process:

$$dX_t = \theta(\mu - X_t)dt + \sigma_w dW_t,$$

The process can be discretized at times $(k\Delta t)_{k \in \mathbb{N}_0}$:

$$X_{k+1} - X_k = \theta\mu\delta t - \theta X_k \Delta t + \sigma_w(W_{k+1} - W_k)$$

The random variables $(W_{k+1} - W_k)$ are independent and identically distributed normal random variables with expected value zero and variance Δt. Therefore, we can set $\sigma_w(W_{k+1} - W_k) = \sigma_w \sqrt{\Delta t}\epsilon$ with $\epsilon \sim \mathcal{N}(0, 1)$ to obtain the following recursion:

$$X_{k+1} = \theta\mu\Delta t - (\theta\Delta t - 1)X_k + \sigma_w \sqrt{\Delta t}\epsilon$$

The recursion for an AR(1) process is:

$$X_{k+1} = c + aX_k + b\epsilon$$

Which is identical to the expression above if $c = \theta\mu\Delta t$, $a = 1 - \theta\Delta t$ and $b = \sigma_w \sqrt{\Delta t}$

More generally a continuous AR(p) process has Matérn covariance function with $\nu = p - 1/2$. TODO: different kernels, stationary kernels, link to power spectral density, with plots A more detailed description of covariance functions and their property can be found in Rasmussen and Williams (chapter 4).

4.6 Performance Assessment

Inference, in the case of Gaussian process regression, revolves around the posterior (predictive) distribution of the response variable. To evaluate how effectively the predictive distribution explains the observed values $\mathbf{y}^*$, it is common practice to calculate the probability of these values based on the predictive distribution. Equation 4.3.1.4 presents an expression for the predictive distribution of $\mathbf{Y}^* := [Y(x_1^*), \ldots, Y(x_k^*)]^\top$ at arbitrary inputs $X^* = [x_1^*, \ldots, x_k^*]$. Expanding the expression from 4.3.1.4 we obtain:

$$\log p(\mathbf{Y}^* = \mathbf{y}^* | \mathbf{Y}, X) = -\frac{k}{2} \log 2\pi - \frac{1}{2} \log |\bar{\Sigma}| - \frac{1}{2}(\mathbf{y}^* - \bar{\mu})^\top \bar{\Sigma}^{-1}(\mathbf{y}^* - \bar{\mu}) \tag{4.6.0.1}$$

where $\bar{\Sigma} = K_{X^*X^*} - K_{X^*X} K_{XX}^{-1} K_{XX^*}$ and $\bar{\mu} = K_{X^*X} K_{XX}^{-1} \mathbf{Y}$.

The higher the log probability, the better the fit to the data. In contract to other performance metrics it accounts for the complete predictive distribution rather than just a point estimate. For instance, when employing the sum of squared errors between the true values y^* and the predictive mean $\bar{\mu}$, the predictive covariance matrix $\bar{\Sigma}$ is completely ignored.

4.7 Model Selection

Model selection in Gaussian process regression involves identifying the optimal covariance function along with the optimal hyperparameters. Two common approaches for model selection are cross-validation, using a performance-based loss function as discussed in Section 4.6, and Bayesian model selection, which will be explored in the subsequent subsections. The concepts and ideas discussed in this section are primarily derived from Chapter 5 of the textbook from Rasmussen and Williams.

4.7.1 Bayesian Model Selection

Bayesian model selection aims to find the most probable model given the available data using a hierarchical specification of the model. In a parametric model setting, the lowest level consists of the parameters β, followed by the hyperparameters θ, which control the parameter distribution. The highest level encompasses the set of possible model structures M_i.

The posterior distribution over the parameters β is determined using Bayes' rule:

$$p(\beta|\mathbf{Y}, X, \theta, M_i) = \frac{p(\mathbf{Y}|X, \beta, M_i)p(\beta|\theta, M_i)}{p(\mathbf{Y}|X, \theta, M_i)}$$

Here, $p(\mathbf{Y}|X, \beta, M_i)$ represents the likelihood, $p(\beta|\theta, M_i)$ denotes the prior, and $p(\mathbf{Y}|X, \theta, M_i)$ represents the marginal likelihood.

However, in the non-parametric setting of Gaussian processes, the parameter β does not exist and is replaced by the function f itself. Consequently, at the lowest level, the distribution over the function f is modeled using a Gaussian process, parametrized by θ. Similarly to the parametric setting, the posterior distribution over the function values $f^* = f(x^*)$ at some arbitrary input x^* is given by:

$$p(f^*|\mathbf{Y}, X, \theta, M_i) = \frac{p(\mathbf{Y}|f^*, M_i)p(f^*|\theta, M_i)}{p(\mathbf{Y}|X, \theta, M_i)}$$

Please refer to Equation 4.3.1.3 for the derivation of the posterior distribution over the function values f^* when assuming $f \sim GP(0, k(x, x'))$.

By assuming a prior distribution over the hyperparameters θ, a similar expression can be obtained for the posterior distribution over the hyperparameters:

$$p(\theta|\mathbf{Y}, X, M_i) = \frac{p(\mathbf{Y}|X, M_i, \theta)p(\theta|M_i)}{p(\mathbf{Y}|X, M_i)}$$

Maximizing $p(\theta|\mathbf{Y}, X, M_i)$ yields the optimal hyperparameters. However, when non-Gaussian priors are assumed for θ, evaluating $p(\theta|\mathbf{Y}, X, M_i)$ can be challenging. In such cases, it is common to maximize the marginal likelihood $p(\mathbf{Y}|X, \theta, M_i)$ with respect to the hyperparameters θ. This approach is equivalent to assuming uniform distributions over the hyperparameters. The next subsection will provide more details on how to calculate and maximize the marginal likelihood for Gaussian process regression.

Note that the scheme mentioned above can be extended to maximize the posterior over the model structures M_i in order to determine the optimal model structure. In Gaussian process regression, this corresponds to finding the optimal kernel function type. However, instead of directly evaluating the posterior, it is often achieved through simultaneous optimization of the marginal likelihood with respect to the model structure M_i and its hyperparameters θ. By jointly optimizing these components, we can effectively identify the most suitable kernel function for the given problem.

Marginal Likelihood

In the context of Bayesian linear regression, the marginal likelihood expression was previously introduced in subsection 4.2, assuming a prior distribution of $p(\beta) = \mathcal{N}(0, \Sigma_p)$ and a likelihood function of $p(\mathbf{Y}|X, \beta) = \mathcal{N}(X\beta, \Sigma_e)$. The following expression for the marginal likelihood is obtained by marginalizing over β:

$$p(\mathbf{Y}|X) = \int p(\mathbf{Y}|X, \beta)p(\beta)d\beta = \mathcal{N}(0, X\Sigma_p X^\top + \Sigma_e) \tag{4.7.1.1}$$

Furthermore, as discussed in section 4.3, the marginal likelihood can also be represented as a distribution over the function $Y(x)$:

$$Y(x) \sim GP(0, k(x, x'))$$

Here, the kernel function $k(x, x')$ is chosen such that for an index set X, we obtain $K_{XX} = X\Sigma_p X^\top + \Sigma_e$.

By the definition of a Gaussian process, $\mathbf{Y}|X$ follows a multivariate normal distribution with a covariance matrix of $K_{XX}(\theta)$, which is a function of the hyperparameters θ. The log marginal likelihood is hence given by:

$$\log p(\mathbf{Y}|X, \theta) = -\frac{1}{2}\mathbf{Y}^\top K_{XX}^{-1}(\theta)\mathbf{Y} - \frac{1}{2}\log|K_{XX}(\theta)| - \frac{n}{2}\log 2\pi \tag{4.7.1.2}$$

Since the marginal likelihood already incorporates a trade-off between model fit and model complexity, it is a suitable candidate for solving the model selection problem. The first term, $-\frac{1}{2}\mathbf{Y}^\top K_{XX}^{-1}(\theta)\mathbf{Y}$, represents a measure of the data fit. The second term, $\frac{1}{2}\log|K_{XX}(\theta)|$, penalizes more complex models. The last term $\frac{n}{2}\log 2\pi$ serves as a normalization constant.

Chapter 5

First Chapter

5.1 To include a picture

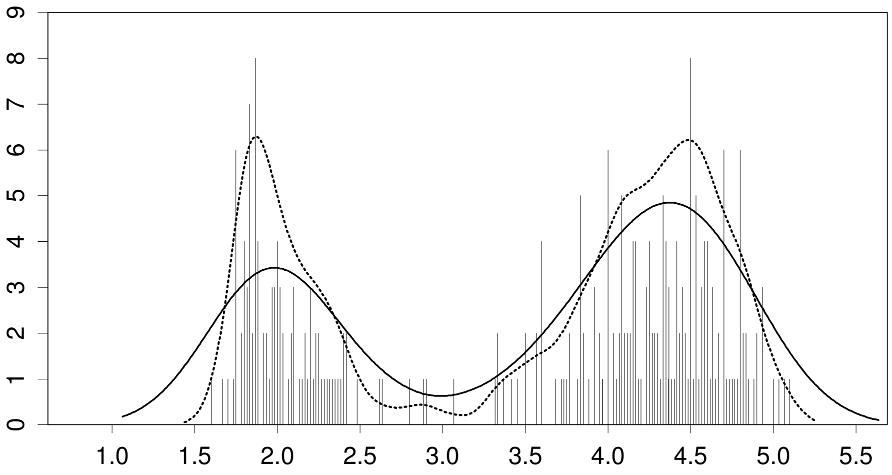

Figure 5.1: Old Faithful Geyser eruption lengths, $n = 272$; binned data and two (Gaussian) kernel density estimates ($\times 10$) with $h = h^* = .3348$ and $h = .1$ (dotted).

Or also with `includegraphics`:

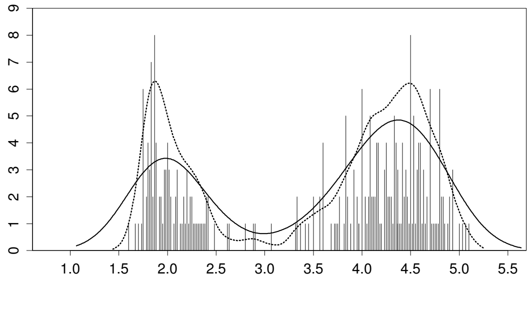

Figure 5.2: Old Faithful Geyser eruption lengths, $n = 272$; binned data and two (Gaussian) kernel density estimates ($\times 10$) with $h = h^* = .3348$ and $h = .1$ (dotted).

5.2 To make a proof

Proof. $1 + 1 = 2$ □

5.3 To include R code

See information in Appendix A.

5.4 Other information

Put a text between quotes: make sure to use nice quotes, such as “quote”.

Cite an article or book you refer shortly here, and then listed in the bibliography. Or mention that Robinson (a person) (two persons) have already done quite a bit work.

Marvasti and Wolf

Referencing a different part of your work: please refer to Appendix A.

Chapter 6

Summary

Summarize the presented work. Why is it useful to the research field or institute?

6.1 Future Work

Possible ways to extend the work.

Bibliography

Andrews, D. W. K. (1991, May). Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation. *Econometrica 59*(3), 817. Number: 3.

Box, G. E. P., G. M. Jenkins, and G. C. Reinsel (1994). *Time series analysis: forecasting and control* (3rd ed ed.). Englewood Cliffs, N.J: Prentice Hall.

Brockwell, P. J. and R. A. Davis (1991). *Time Series: Theory and Methods.* Springer Series in Statistics. New York, NY: Springer New York.

Brockwell, P. J. and R. A. Davis (2016). *Introduction to Time Series and Forecasting.* Springer Texts in Statistics. Cham: Springer International Publishing.

Chatfield, C. (2003, July). *The Analysis of Time Series* (0 ed.). Chapman and Hall/CRC.

Cryer, J. D. and K.-s. Chan (2008). *Time series analysis: with applications in R* (2nd ed ed.). Springer texts in statistics. New York: Springer. OCLC: ocn191760003.

Marvasti, F. and J. K. Wolf (Eds.) (2001). *Nonuniform Sampling.* Information Technology: Transmission, Processing, and Storage. Boston, MA: Springer US. Series Editors: _:n5.

Newey, W. K. and K. D. West (1994, October). Automatic Lag Selection in Covariance Matrix Estimation. *The Review of Economic Studies 61*(4), 631–653. Number: 4.

Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian processes for machine learning.* Adaptive computation and machine learning. Cambridge, Mass: MIT Press. OCLC: ocm61285753.

Robinson, P. (1977, November). Estimation of a time series model from unequally spaced data. *Stochastic Processes and their Applications 6*(1), 9–24. Number: 1.

von Mises, R. (1964). *Mathematical Theory of Probability and Statistics.* Elsevier.

White, H. (2001). *Asymptotic theory for econometricians* (Rev. ed ed.). San Diego: Academic Press.

Zeileis, A. (2004). Econometric Computing with HC and HAC Covariance Matrix Estimators. *Journal of Statistical Software 11*(10). Number: 10.

Appendix A

Complementary information

Additional material. For example long mathematical derivations could be given in the appendix. Or you could include part of your code that is needed in printed form. You can add several Appendices to your thesis (as you can include several chapters in the main part of your work).

A.1 Including R code with verbatim

A simple (rather too simple, see A.2) way to include code or *R* output is to use `verbatim`. It just prints the text however it is (including all spaces, “strange” symbols,...) in a slightly different font.

```
## loading packages
library(RBGL)
library(Rgraphviz)
library(boot)

## global variables
X_MAX <- 150

   This allows me to put as many s  p a   c es    as I want.
I can also use \ and ‘ and & and all the rest that is usually only
accepted in the math mode.

I can also make as
                  many
             line
    breaks as
I want... and
             where I want.
```

But really recommended, much better is the following:

A.2 Including R code with the *listings* package

However, it is much nicer to use the *listings* package to include R code in your report. It allows you to number the lines, color the comments differently than the code, and so on. All the following is produced by simply writing `\lstinputlisting{Pictures/picture.R}` in your LaTeX “code”:

```
## Example to generate an .pdf file with the function pdf.latex()
## Author: Sarah Gerster and Martin Maechler (UTF-8 Umlaute seem to fail here !?)
## Last revision: 16 Aug 2011

require("sfsmisc") # pdf.latex(), pdf.end(), etc

pdf.latex(file='test_plot.pdf') #, main=TRUE)
## no main=TRUE is needed to leave enough space for the plot title
## but see below

## make sure the legends are large enough
par(cex=1.5)

## Make sure your lines are "visible" enough. Otherwise your plot
## won't look very nicely in your text.
plot(-10:10, (-10:10)**2, type="l", lty=5,
     xlab="my_x", ylab="my_y",
     ## no main title: NOT recommended for figures in text which
     ## have a \caption{..}
     lwd=4, col='blue')
lines(-10:10, 0:20, type="p", lwd=4, pch=23,col='red')
legend(-3, 90, c("func1","func2"),lwd=4,col=c('blue', 'red'),
       lty=c(1,1),cex=1)
pdf.end() # starts the previewer (which refreshes itself;
          # at least on Linux at SfS
```

or `\lstinputlisting{/u/maechler/R/Pkgs/sfsmisc/R/ellipse.R}` :

```
ellipsePoints ← function(a,b, alpha = 0, loc = c(0,0), n = 201,
                          keep.ab.order = FALSE)
{
    ## Purpose: ellipse points,radially equispaced, given geometric par.s
    ## ------------------------------------------------------------------------
    ## Arguments: a, b : length of half axes in (x,y) direction
    ##            alpha: angle (in degrees) for rotation
    ##            loc  : center of ellipse
    ##            n    : number of points
    ## ------------------------------------------------------------------------
    ## Author: Martin Maechler, Date: 19 Mar 2002

    stopifnot(is.numeric(a), is.numeric(b))
    reorder ← a < b && keep.ab.order
    B ← min(a,b)
    A ← max(a,b)
    ## B <= A
    d2 ← (A-B)*(A+B) ## = A^2 - B^2
    phi ← 2*pi*seq(0,1, len = n)
    sp ← sin(phi)
    cp ← cos(phi)
    r ← a*b / sqrt(B^2 + d2 * sp^2)
    xy ← r * if(reorder) cbind(sp, cp) else cbind(cp, sp)
    ## xy are the ellipse points for alpha = 0 and loc = (0,0)
    al ← alpha * pi/180
    ca ← cos(al)
    sa ← sin(al)
    xy %*% rbind(c(ca, sa), c(-sa, ca)) + cbind(rep(loc[1],n),
                                                rep(loc[2],n))
}
```

A.3 Using Sweave (or knitr) to include R code (and more) in your report

The easiest (and most elegant) way to include R code and its output (and have all your figures up to date with your report) is to use Sweave—or the **knitr** R package with even more possibilities.

Search the web to find lots of intro material on how to use Sweave or knitr (on Wikipedia).

Appendix B

Yet another appendix....

B.1 Description

Something details.

Something else other definition.

B.2 Tables

Refer to Table B.1 to see a left justified table with caption on top.

Table B.1: Results.

Student	**Grade**
Marie	6
Alain	5.5
Josette	4.5
Pierre	5

Appendix C

2nd Appendix: More sophisticated R code listing

Chapter-wise listing of parts of R code, using

- `firstline=n1`
- `lastline=n2`
- `title=<text>`

e.g., for the first example below

```
\lstinputlisting[firstline=1,lastline=20,
                 title= \texttt{ellipse.R}]{ellipse.R}
```

and the second example

```
\lstinputlisting[firstline=20,lastline=40,
             title=\texttt{ellipse.R}]{ellipse.R}
```

C.1 Chapter 5

```
ellipsePoints ← function(a,b, alpha = 0, loc = c(0,0), n = 201,
                          keep.ab.order = FALSE)
{
    ## Purpose: ellipse points,radially equispaced, given geometric par.s
    ## -------------------------------------------------------------------------
    ## Arguments: a, b : length of half axes in (x,y) direction
    ##            alpha: angle (in degrees) for rotation
    ##            loc  : center of ellipse
    ##            n    : number of points
    ## -------------------------------------------------------------------------
    ## Author: Martin Maechler, Date: 19 Mar 2002

    stopifnot(is.numeric(a), is.numeric(b))
    reorder ← a < b && keep.ab.order
    B ← min(a,b)
    A ← max(a,b)
    ## B <= A
    d2 ← (A-B)*(A+B) ## = A^2 - B^2
    phi ← 2*pi*seq(0,1, len = n)
    sp ← sin(phi)
```

ellipse.R

```
    sp ← sin(phi)
    cp ← cos(phi)
    r ← a*b / sqrt(B^2 + d2 * sp^2)
    xy ← r * if(reorder) cbind(sp, cp) else cbind(cp, sp)
    ## xy are the ellipse points for alpha = 0 and loc = (0,0)
    al ← alpha * pi/180
    ca ← cos(al)
    sa ← sin(al)
    xy %*% rbind(c(ca, sa), c(-sa, ca)) + cbind(rep(loc[1],n),
                                                rep(loc[2],n))
}
```

ellipse.R

Epilogue

A few final words.

Declaration of Originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor .

Title of work (in block letters):

. . .

Authored by (in block letters):
For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
Muster	Student

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the Citation etiquette information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work .
- I am aware that the work may be screened electronically for plagiarism.
- I have understood and followed the guidelines in the document *Scientific Works in Mathematics.*

Place, date:

Zurich August 19th 2009

Signature(s):

bla

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire content of the written paper.