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

Abstract

Short summary of my thesis.

Contents

0.1 General Statements . . . xiii
0.2 Abbreviation . . . xiii
0.3 Symbols . . . xiii

Notation **xiii**

1 Introduction **1**
1.1 Motivation and Thesis Objective . . . 1
1.2 Problem Statement . . . 2
1.2.1 Characteristics of the Blood Pressure Time Series . . . 2
1.2.2 Target Measures . . . 3
1.3 Thesis Outline . . . 3

2 Characteristics of Time Series **5**
2.1 Time Series Definition . . . 5
2.2 Moments of a Time Series . . . 5
2.3 Stationarity . . . 5
2.4 Special Cases of Time Series Processes . . . 6

3 Time Series Decomposition and Linear Regression **7**
3.1 Linear Regression with Uncorrelated Errors . . . 7
3.2 Linear Regression with Correlated Errors . . . 8
3.2.1 Maximum-Likelihood Estimation . . . 8
3.2.2 Sandwich Estimation . . . 9
3.2.3 Extension to Irregularly Spaced Time Series . . . 9
3.2.4 Confidence Intervals for the Mean Function . . . 9

4 Gaussian Process Regression **11**
4.1 Gaussian Process Definition . . . 11
4.2 Bayesian Linear Regression . . . 12
4.3 Bayesian Linear Regression as Gaussian Process Regression . . . 15
4.3.1 Time Series Gaussian Process Regression . . . 16
4.4 Mean Function . . . 17
4.5 Kernel Functions . . . 18
4.5.1 Squared Exponential Kernel . . . 18
4.5.2 Matérn Class of Kernels . . . 18
4.5.3 Periodic Kernel . . . 20
4.5.4 Additive Kernels and Decomposition of Predictive Mean . . . 20
4.6 Performance Assessment . . . 21
4.7 Model Selection . . . 21
4.7.1 Bayesian Model Selection . . . 21

5 Methods **25**
5.1 Problem Statement . . . 25
5.2 Overview . . . 26
5.3 Target Measures . . . 26
5.4 Blood Pressure Time Series Simulation . . . 27
5.4.1 Mean function . . . 27
5.4.2 Kernel function . . . 27

5.4.3 Simulation of the BP Measurements . . . 28
5.5 Gaussian Process Regression . . . 28
5.6 Baseline Methods . . . 29
5.6.1 Linear Regression . . . 30
5.6.2 Smoothing Spline . . . 30
5.6.3 Overall Mean . . . 31
5.6.4 Naive TTR . . . 31
5.7 Adversarial Analysis . . . 31
5.8 Computational Frameworks . . . 33

6 First Chapter 35
6.1 To include a picture . . . 35
6.2 To make a proof . . . 36
6.3 To include R code . . . 36
6.4 Other information . . . 36

7 Summary 37
7.1 Future Work . . . 37

Bibliography 39

A Complementary information 41
A.1 Ornstein-Uhlenbeck Process . . . 41
A.2 Properties of the Simulated Time Series Samples . . . 42

B Yet another appendix.... 45
B.1 Description . . . 45
B.2 Tables . . . 45

C 2nd Appendix: More sophisticated R code listing 47
C.1 Chapter 5 . . . 47

Epilogue 49

List of Figures

4.1 RBF Kernel: Kernel function with sample path 19
4.2 Matérn Kernel: Kernel function with sample path 19
4.3 Periodic Kernel: Kernel function with sample path 20

5.1 The true kernel function $k(x, x')$ 28
5.2 Three samples (right side) drawn from the true GP and the decomposition of theses samples (left side) 29
5.3 Linear Regression and Smoothing spline used to estimate some example of true BP values F_X. The estimated BP values $\hat{F}_X$ (black line), the true BP values (red dashed line) and the training data (red dots). 32
5.4 Panel b shows the sample $f(x)$ drawn from the true GP with measurments generated by seasonal sampling. Panel a, shows $f(x)$ decomposed into its components. 33

6.1 Geyser data: binned histogram, Silverman's and another kernel 35
6.2 Geyser data: binned histogram, Silverman's and another kernel 35

A.1 The one-week sample variance should span from from 16 to 144 mmHg^2, with an average of 49 mmHg^2 42
A.2 Half of the difference between average daytime and nighttime BP measurements. Should fall within the range of 0 to 10 mmHg 43
A.3 The variance of the AR(1) component. There exists no target values for this. 43
A.4 The variance of the RBF component. There exists no target values for this. 44
A.5 The variance of the periodic component. 44

List of Tables

B.1 Test results 45

Notation

0.1 General Statements

Prediction: Estimation of the expected time series value at some time x^*, with x^* being within the time range of available observations.
Forcasting: Estimation of the expected time series value at some time x^*, with x^* being after the last available observations
Vectors are column vectors unless stated otherwise.

Blood Pressure or BP always refers to the systolic blood pressure

0.2 Abbreviation

GP: Gaussian process.
BP: (Systolic) Blood pressure.
CI: Refers to both confidence and credible interval
OLS: Ordinary Least Squares.
iid: Independent and identically distributed.

0.3 Symbols

$\mathcal{N}(\mu, \sigma^2)$: Normal distribution with mean μ and standard deviation σ
$X_1 \ldots X_n$ iid $\sim F$: $X_1 \ldots X_n$ are iid with distribution F
$|M|$: determinant of matrix M
$\mathbf{E}[X]$: Expectation of X
$\mathrm{Cov}(X, Y)$: Covariance between X and Y
$\mathrm{Var}(X)$: Variance of X

Chapter 1

Introduction

1.1 Motivation and Thesis Objective

This thesis aims at presenting Gaussian process regression as a powerful tool for modeling time series based on irregularly spaced observations.

The motivation for this research stems from a pressing real-world problem in the field of medicine, which will serve as a recurring example throughout this thesis. The problem revolves around estimating critical time series properties, from a dataset consisting of irregularly spaced blood pressure (BP) measurements. High BP is a well-established risk factor for cardiovascular disease, and summarizing an individual's BP levels typically involves calculating the average BP value over available measurements within a specified time range. A novel monitoring device has been developed by the company Aktiia. The device collects continuous BP estimates by converting photoplethysmography (PPG) signals into BP measurements. The sampling frequency of this system can vary widely, typically yielding around 1.5 BP measurements per hour. However, factors such as PPG signal quality and external conditions can influence this frequency, resulting in irregularly spaced measurements. Obtaining accurate estimates of true BP values at unobserved time points is essential for improving cardiovascular risk assessment and developing valuable metrics.

Standard time series analysis methods traditionally assume discrete equispaced time intervals. Introductory textbooks on time series analysis either neglect the irregularly spaced case entirely or dedicate only a limited section to continuous time models or state-space models with missing observations (Brockwell and Davis, Brockwell and Davis, Cryer and Chan, Chatfield).

Therefore, the primary objective of this thesis is to address the challenges posed by irregularly sampled time series data and demonstrate why conventional time series methods fail to deal with it. Additionally, we will elucidate why Gaussian processes are a suitable approach for modeling time series with irregularly spaced observations, using the BP time series example.

1.2 Problem Statement

To begin modeling BP measurements, we introduce the time series process $Y(x)$, which combines the true BP process $f(x)$ with independent and identically distributed (iid) Gaussian measurement noise ϵ:

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

Both time series, $f(x)$ and $Y(x)$, are described as random functions. While the former is completely unobserved we have unequally spaced observations from the latter, i.e. $(Y_{t_i} : i \in \{1, 2, \ldots n\})$. These observations represent Aktiia's user data.

The goal of this research is to learn about the underlying true BP process $f(x)$ based on one week of irregularly spaced observations. Instead of using real data, data will be simulated by generating the true BP process $f(x)$ and adding measurement noise ϵ. This approach offers the advantage of complete knowledge about $f(x)$, enabling us to quantify the accuracy of its reconstruction from data. However, it also introduces the challenge of simulating a time series and observations that closely mimic reality. The time series characteristics to mimic are desribed in the next subsection 1.2.1.

Instead of solely focusing on predicting $f(x)$, this thesis emphasizes a set of target measures deemed most relevant for assessing cardiovascular risk. These target measures are detailed in subsection 1.2.2.

In addition to point estimates, this research considers the construction of confidence intervals (CIs) around these estimates. Notably, the width of the CI intervals around the mean function varies over time, depending on factors such as the availability of data in the vicinity of a given time point.

For simplicity, this study exclusively deals with systolic blood pressure and does not consider diastolic measurements. All references to ”blood pressure” or ”BP” pertain to systolic blood pressure.

1.2.1 Characteristics of the Blood Pressure Time Series

Based on the Aktiia user data, several properties of **the BP measurements,** $(Y_{t_i} : i \in \{1, 2, \ldots n\})$, have been identified:

i.) The measurements are irregularly spaced, meaning that the time between consecutive measurements varies.

ii.) Observations are not uniformly sampled across time; instead, their density follows a circadian cycle, resulting in seasonal sampling.

iii.) The sampling frequency ranges from 0.5 to 4 measurements per hour.

iv.) The difference between average daytime and nighttime BP measurements falls within the range of 0 to 20 mmHg, with an average difference of 10 mmHg.

v.) The mean BP across all users is 120 mmHg.

vi.) The within-subject one-week sample variance spans from 16 to 144 mmHg$^{\mathbf{2}}$, with an average of 49 mmHg$^{\mathbf{2}}$.

The true BP time series process, $f(x)$, cannot be directly observed. However, in this thesis, it is assumed to be a combination of the following components:

- A seasonal component representing the circadian cycle, as BP tends to be higher during the day than at night.
- An autoregressive component, reflecting the dependence of the output variable on its previous values.
- A long-term trend.

The magnitude of the measurement noise, denoted as ϵ, remains unknown. Nevertheless, Aktiia measurements have undergone validation against a reference method. The measured variance of the differences between Aktiia measurements and this reference is 62 mmHg$^{\mathbf{2}}$. Consequently, we can express:

$$\begin{aligned}\mathrm{Var}(BP_{Ref} - BP_{Aktiia}) &= 62 \text{ mmHg}^2 = \mathrm{Var}(\epsilon_{Ref} - \epsilon_{Aktiia}) \\ &= \mathrm{Var}(\epsilon_{Ref}) + \mathrm{Var}(\epsilon_{Aktiia}) - 2\,\mathrm{Cov}(\epsilon_{Ref}, \epsilon_{Aktiia})\end{aligned}$$

Assuming that the noise variance of the reference method, $\mathrm{Var}(\epsilon_{Ref})$, equals that of the Aktiia measurements, $\mathrm{Var}(\epsilon_{Aktiia})$, and that $\mathrm{Cov}(\epsilon_{Ref}, \epsilon_{Aktiia}) = 0$, we would obtain a noise variance for the Aktiia measurements of 31 mmHg$^{\mathbf{2}}$.

1.2.2 Target Measures

The primary focus of this research lies on a set of target measures crucial for estimating an individual's cardiovascular risk. These measures include:

The mean BP calculated over different time windows, such as one-hour, one-day, and one-week mean BP. The mean BP is a pivotal and frequently reported metric. Presently, it is computed based on the available measurements within the corresponding time range.

Time in Target Range (TTR) evaluates the duration during which BP values fall within a specified target range relative to the total time. It is currently determined by dividing the number of BP measurements within the range of 90 to 125 mmHg ("target range") by the total number of BP measurements available within one week.

It is noteworthy that the estimation of these target measures does not depend on forecasting future BP values but solely relies on predicting BP values within the one-week range of available data. Consequently, this thesis concentrates on reconstructing BP values between the first and last time point in the dataset.

1.3 Thesis Outline

[Insert your thesis outline here]

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 pairs (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:

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

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

$$\mu_X(t) = \mathbf{E}\left[X_t\right]$$

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

$$\gamma_X(r, s) = \operatorname{Cov}\left(X_r, X_s\right) = \mathbf{E}\left[(X_r - \mu_X(r))(X_s - \mu_X(s))\right]$$

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 if and only if 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*

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

and

$$\gamma_X(t+h, t) \text{ is independent of } t \text{ for each } h.$$

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 an **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 a **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 an **autoregressive moving average process** of autoregressive order p and moving average order 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, $k < t$.

Chapter 3

Time Series Decomposition and Linear 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 $R(t)$. This can be expressed in the form of a regression problem:

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

The decomposition allows to extract a stationary component $R(t)$, 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 with Uncorrelated Errors

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 $\alpha \cos(2\pi f t - \phi)$ with phase shift ϕ and known frequency f, we get the following model for the BP time series $Y(t)$:

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

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:

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

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 $\mathbf{R} = [R_{t_1}, \dots R_{t_n}]^\top$ 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 $R_{t_1}, \dots, R_{t_n}$ (White). In the case of uncorrelated errors with constant variance σ^2 we have $Var(\mathbf{R}) = \sigma^2 I_n$ 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 $\mathbf{R}$ 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 Linear 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 $Var(\mathbf{R}) = \Sigma$ 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 $\mathbf{R}$ 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 $\mathbf{R}$ and equispaced time points, hence $\mathbf{R} = (R_t : t \in \{1, 2, \dots n\})$ and:

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

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 $(R_t : t \in \{1, 2, \dots n\})$. 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:

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

The general form of the estimators for Φ is:

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

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.

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

$$\begin{aligned} m(x) &= \mathbf{E}\left[f(x)\right] \\ k(x, x') &= \mathbf{E}\left[(f(x) - m(x))(f(x') - m(x'))\right] \end{aligned}$$

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:

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

for

$$K_{AA} = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \ldots & k(x_1,x_n) \\ \vdots & & \vdots & \vdots \\ k(x_n,x_1) & k(x_n,x_1) & \ldots & k(x_n,x_n) \end{bmatrix} \text{ and } \mu_A = \begin{bmatrix} m(x_1) \\ \vdots \\ m(x_n) \end{bmatrix} \qquad (4.1.0.1)$$

The finite marginals $F_{x_1}, \ldots, F_{x_n}$ 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

In the context of Bayesian regression, the objective is to estimate the posterior distribution of $f^* := f(x^*)$, at some input x^*, based on 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. Consequently, in Bayesian linear regression, a distribution over the regression coefficients β is assumed.

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 R_i does not necessarily need to be from a time series process $(R_t : t \in \{t_1, t_2, \ldots t_n\})$. We obtain the following data generating model:

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

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:

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

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 $\mathbf{R} = [R_1, \dots R_n]^\top$:

$$\mathbf{R} \sim \mathcal{N}(0, \Sigma_r)$$

If $\mathbf{R}$ is an ARMA process, then every element of the time series R_i is itself a sum of innovations. Therefore, $\mathbf{R}$ 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:

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

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:

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

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:

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

with $\Sigma_p X^\top + \Sigma_r \in \mathbb{R}^{n\times n}$ 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:

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

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

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

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 $p(\mathbf{Y}, f^*|X, x^*)$:

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

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:

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

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 $\mathbf{R}$ not simply as an error term but an actual part of our signal which we can predict. If $\mathbf{R}$ is not independent noise but for example a time series, where the elements of $\mathbf{R}$ 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 $Y^* := Y(x^*) = f(x^*) + R(x^*)$.

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

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

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 $K_{XX} = X\Sigma_p X^\top + \Sigma_r$. 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 idd measurement noise on the time series $f(x) + R(x)$. We then have for the observed time series $Y(x)$:

$$Y(x_i) = f(x_i) + R(x_i) + \epsilon_i \qquad \epsilon_1 \dots \epsilon_n \text{ iid} \sim \mathcal{N}(0, \sigma_n^2)$$

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 $R(x)$. The observed time series $Y(x)$ will thus be equivalent to $f(x)$ up to some additive independent noise term ϵ. We can write:

$$Y(x_i) = f(x_i) + \epsilon_i \qquad \epsilon_1 \dots \epsilon_n \text{ iid} \sim \mathcal{N}(0, \sigma_n^2)$$

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

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

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 $K_{XX} = X\Sigma_p X^\top + \Sigma_r$.

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 $K_{XX} = X\Sigma_p X^\top + \Sigma_r + \sigma_n^2 I$. 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_r, the covariance matrix of $\mathbf{R}$, to be known. However, deriving Σ_r for an ARMA process with irregularly spaced samples is not straight forward, as has already been shown in chapter 3. Section 4.5 will illustrate how choosing a specific kernel function solves this problem.

4.4 Mean Function

A Gaussian process is defined by its mean function, $\mu(x)$, and covariance function, $k(x, x')$. The mean function can be subtracted from the data without affecting the covariance. In Gaussian process regression, this means the predictive variance remains independent of the mean function. Consider $Y(x) = f(x) + \epsilon$, with $f(x) = m(x) + R(x)$. Where, $m(x)$ is a deterministic mean function, $R(X)$ is a time series process and ϵ is Gaussian iid noise with variance σ_n^2. The noise term is independent of the time series process $R(x)$.

This setup allows us to model $f(x)$ using a Gaussian Process:

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

Conditioning leads to the predictive distribution for $f^* := f(x^*)$:

$$\begin{gathered} p(f^*|\mathbf{Y} = y, X, x^*) = N(\bar{\mu}, \bar{\Sigma}), \\ \bar{\mu} = m(x^*) + K_{x^*X}(K_{XX} + \sigma_n^2 I)^{-1}(y - m(X)), \\ \bar{\Sigma} = K_{x^*x^*} - K_{x^*X}(K_{XX} + \sigma_n^2 I)^{-1}K(X, x^*) \end{gathered}$$

The predictive variance $\bar{\Sigma}$ remains unaffected by $m(x)$.

Alternatively, fitting a GP to $f(x) - m(x) = R(x)$ gives:

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

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

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

The Predictive distribution over f^* is recovered by adding $m(x^*)$ to $\bar{\mu}_{R^*}$. The predictive variance $\bar{\Sigma}$ remains unchanged, unaffected by observations or $m(x)$.

Most frameworks for GP regression assume a zero mean prior. Therefore, when you have prior knowledge about $m(x)$, it's advisable to subtract it before fitting a GP. It is also common practice to subtract the empirical mean from your data, before fitting a GP.

For further insights, Rasmussen and Williams elaborates on this topic in his book, particularly on page 27.

4.5 Kernel Functions

The Gaussian Process is defined by its mean and covariance functions. Assuming a zero-mean Gaussian process, defining a prior distribution over $f(x)$ or $Y(x)$ involves selecting a kernel function only. The kernel is evaluated at inputs $X = [x_1, \ldots, x_n]^\top$ to establish the predictive distribution of f^* or y^*.

Kernel choice depends on assumptions about correlation in your output for input pairs x and x'. For modeling the BP time series, three relevant stationary covariance functions are considered and presented in this section. Following from stationarity, these functions depend on $\tau = x - x'$ only. We discussed covariance functions and stationarity for time series in Sections 2.2 and 2.3, respectively.

The following subsections draw upon the doctoral thesis of Duvenaud and the book of Rasmussen and Williams, which provide comprehensive coverage of covariance functions for Gaussian Processes.

4.5.1 Squared Exponential Kernel

The squared exponential kernel is also known as radial basis function (RBF) kernel or Gaussian kernel and has the form:

$$k(\tau) = \exp(-\frac{\tau^2}{2l^2})$$

Here, l is the length scale, and $\tau = x - x'$. The length scale governs the rate of function change, as shown in Figure 4.1. The RBF kernel generates infinitely differentiable, smooth outputs, regardless of the length scale.

4.5.2 Matérn Class of Kernels

The Matérn covariance function is expressed as:

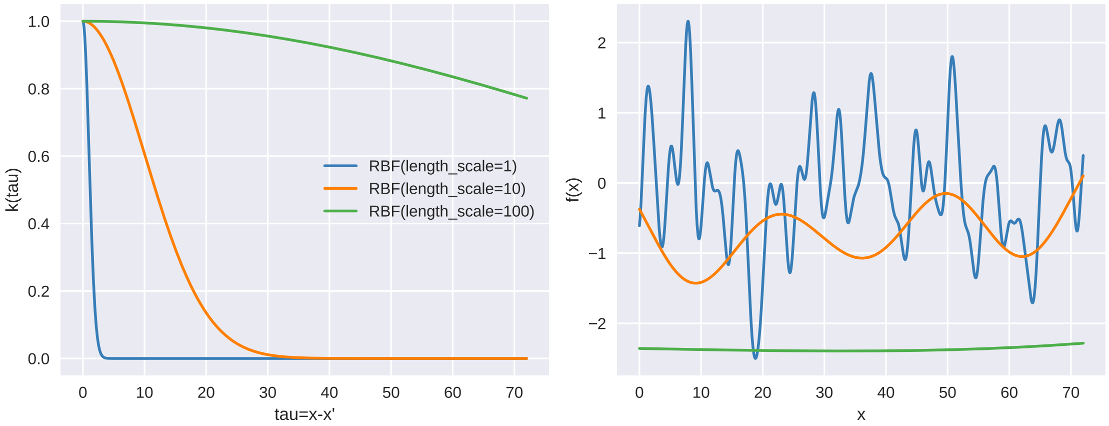


Figure 4.1: RBF Kernel function for different length scale (left panel) and a sample generated by such a GP (right panel)

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

Here, ν and l are positive parameters, and K_ν is a modified Bessel function. Figure 4.2 illustrates the Matérn covariance function and corresponding sample paths for various ν values.

For $\nu = r + 1/2, r \in \mathbb{N}$, the Matérn covariance function simplifies to:

$$k_{\nu=r+1/2}(\tau) = \exp\left(-\frac{\sqrt{2r+1}\tau}{l} \right) \frac{r!}{(2p)!} \sum_{i=0}^{r} \frac{(r+i)!}{i!(r-i)!} \left(\frac{2\sqrt{2r+1}\tau}{l} \right)^{r-i} \tag{4.5.2.1}$$

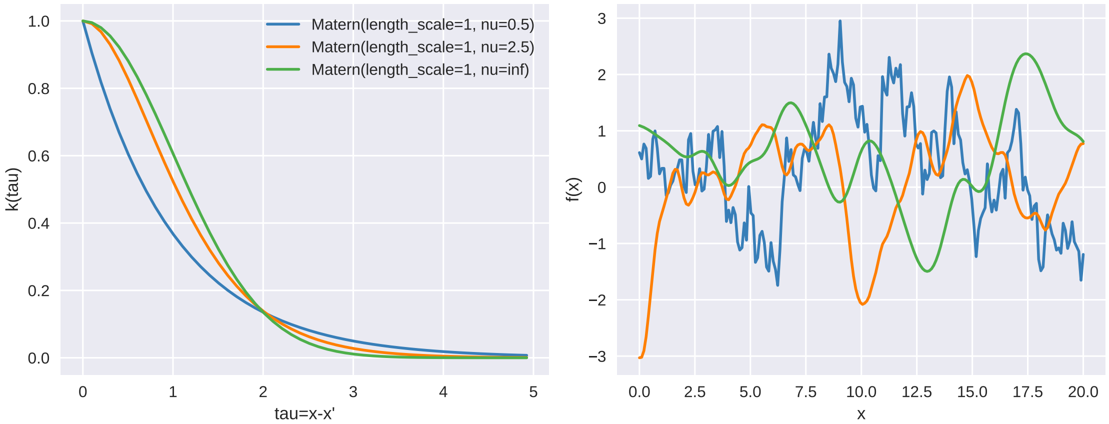


Figure 4.2: Matérn kernel function for different ν (left panel) and a sample generated by the corresponding GP (right panel)

Setting $\nu = 1/2$ with input domain $X \subset \mathbb{R}$ results in a continuous-time AR(1) process, also known as the Ornstein-Uhlenbeck process. With $\nu = 1/2$, i.e., $r = 0$, the Matérn covariance function becomes:

$$k(\tau) = \exp\left(-\frac{\tau}{l}\right) \tag{4.5.2.2}$$

More generally, for $\nu = p - 1/2$ and $X \subset \mathbb{R}$, the Matérn kernel matches the covariance function of a specific case of a continuous AR(p) process. For a deeper understanding of this topic, please refer to chapter 4 of the book by Rasmussen and Williams.

4.5.3 Periodic Kernel

The periodic kernel allows modeling functions with repeating patterns and is defined as:

$$k(x, x') = \sigma^2 \exp\left(-\frac{2\sin^2(\pi|x - x'|/p)}{l^2}\right)$$

Here, p represents the period, and l is the length scale. Figure 4.3 illustrates the impact of different length scales.

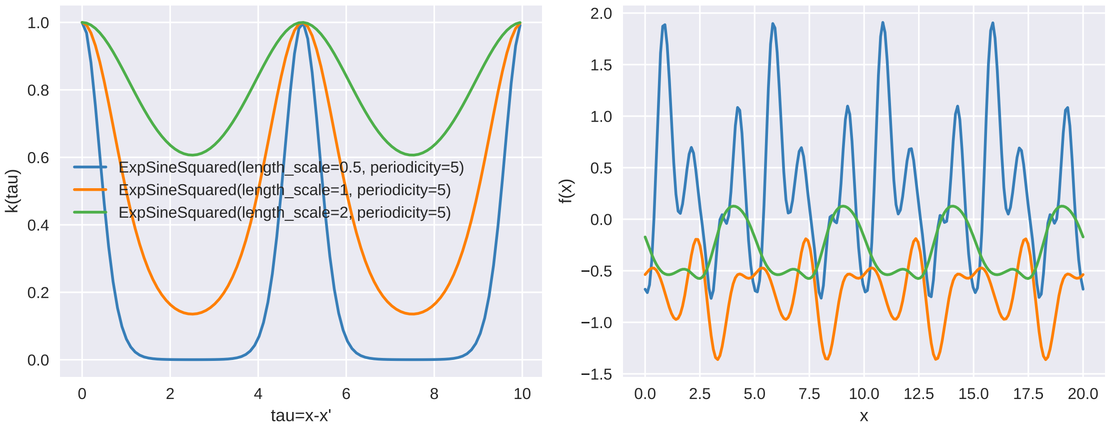


Figure 4.3: Periodic kernel function for different length scales (left panel) and a sample generated by the corresponding GP (right panel)

Throughout this thesis, we will also refer to the periodic kernel as the ExpSineSquared kernel.

4.5.4 Additive Kernels and Decomposition of Predictive Mean

Additivity of the kernel implies additivity of the predictive mean. For instance if we choose $Y(x) \sim GP(0, k(x, x'))$ with $k(x, x') = k_1(x, x') + k_2(x, x')$, then the predictive (posterior) mean $\bar{\mu}(x^*)$ is given by:

$$\begin{aligned}\bar{\mu}(x^*) &= (K_{1,x^*X} + K_{2,x^*X})(K_{XX})^{-1}\mathbf{Y} = K_{1,x^*X}(K_{XX})^{-1}\mathbf{Y} + K_{2,x^*X})(K_{XX})^{-1}\mathbf{Y} \\ &= \bar{\mu}_1(x^*) + \bar{\mu}_2(x^*)\end{aligned}$$

where:

$$K_{1,x^*X} = \begin{bmatrix} k_1(x_1, x^*) & \dots & k_1(x_n, x^*) \end{bmatrix},$$
$$K_{2,x^*X} = \begin{bmatrix} k_2(x_1, x^*) & \dots & k_2(x_n, x^*) \end{bmatrix},$$

and

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

This decomposition allows us to study the contribution of the different (additive) kernel components on the predictive mean function.

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 $\mathbf{Y}^* := [Y(x_1^*), \dots, Y(x_k^*)]^\top$ at arbitrary inputs $X^* = [x_1^*, \dots, x_k^*]$. Expanding the expression from 4.3.1.4 we obtain:

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

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 contrast 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. Similarly to the parametric setting, the posterior distribution over the function values $f^* = f(x^*)$ at some arbitrary input x^* is given by:

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

This is equivalent to the expression in 4.3.1.3 for the posterior distribution over the function values f^* when assuming a Gaussian process prior $f \sim GP(0, k(x, x'))$. However, in the equation above, $k(x, x')$ is expressed through θ and M_i.

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 $p(\mathbf{Y}|X, \beta) = \mathcal{N}(X\beta, \Sigma_r)$. The following expression for the marginal likelihood is obtained by marginalizing over β:

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

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 $K_{XX} = X\Sigma_p X^\top + \Sigma_r$.

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

Methods

The last chapter introduced Gaussian process regression to establish a mapping between a time point x and its corresponding BP value. Since Gaussian Processes are capable of modeling time series in continuous time and hence deal with irregularly spaced data, they seem to be a good candidate for modeling a time series from which we only have irregularly sampled noisy measurements.

This chapter outlines the methodology for evaluating the performance of Gaussian process regression and baseline methods for estimating blood pressure values from noisy measurements. It also discusses the analysis of adversarial factors that may affect estimation accuracy.

5.1 Problem Statement

Recall the problem statement from section 1.2. First, we assumed the following model for the BP measurements $Y(x)$ at a time point x:

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

where $f(x)$ denotes the true BP process and ϵ is iid measurement noise, independent from $f(x)$.

The goal is to estimate the true BP values $f(x)$ at some input time X, based on noisy observations of $f(x)$ at some training time points X_{train}. For the sections of this chapter, we define:

- $X := \{x_1, \dots, x_n\}$: An index set spanning the one-week time range of interest, with 10 BP values per hour. It defines the time points at which we want to predict BP values and hence represents the regression input.
- $X_{train} \subset X$: The training indexes.
- $G_X := (g(x) : x \in X)$ for some function $g(x)$. Specifically:

  $F_X := (f(x) : x \in X)$: The true BP values at inputs X.

$Y_X = (Y(x) : x \in X)$: The noisy BP measurements at inputs X, which constitute the response variable in the context of regression.

$Y_{X_{\text{train}}} := (Y(x) : x \in X_{\text{train}})$: The noisy measurements at training indexes.

- $(X_{train}, Y_{X_{train}})$: The training data used for estimating F_X
- $K_{XX'} := \begin{bmatrix} k(x_1, x'_1) & \dots & k(x_1, x'_n) \\ \vdots & & \vdots \\ k(x_n, x'_1) & \dots & k(x_n, x'_n) \end{bmatrix}$,for some kernel function, $k(x, x')$

  and some inputs $X = (x_1, \dots x_n)$ and $X' = (x'_1, \dots x'_n)$.

Additionally, when referring to the estimated values, $\hat{F}_X$ is used instead of F_X.

5.2 Overview

To assess the suitability of GPs for this problem, the following tasks have been defined:

- Simulate F_X and the training data $(X_{train}, Y_{X_{train}})$ (section 5.4)
- Employ Gaussian process regression to obtain $\hat{F}_X$ from the training data (section 5.5)
- Derive target measures from $\hat{F}_X$, including 95% credible intervals (section 5.3)
- Evaluate performance using:
  - CiCoverage: Equals one if the true target measure value extracted from F_X was covered by the credible interval, zero otherwise.
  - CiWidth: The width of the credible interval

These steps are repeated $S = 100$ times, and the final performance is assessed by averaging CiCoverage and CiWidth. Pseudocode 5.5 provides a more detailed illustration of this process.

To contextualize the performance of GP regression, it is compared to the performance of baseline methods (section 5.6). Additionally, the impact of adversarial factors on estimation accuracy is discussed in section 5.7.

5.3 Target Measures

In subsection 1.2.2, the mean BP over different time windows and TTR has been defined as the measures of interest. These measures are extracted from F_X to obtain the true target measure values and from $\hat{F}_X$ to obtain the estimated target measures.

The **one-week mean BP**, $\bar{F}_X$, was calculated as the mean of all values in F_X:

$$\bar{F}_X = \frac{1}{n} \sum_{x \in X} f(x)$$

The **one-hour and one-day BP means**, $\bar{F}_{X_1} \dots \bar{F}_{X_W}$, were calculated by taking the mean value of $f(x)$ evaluated at the different time windows $X_1 \dots X_W \subset X$. For the first one-hour or one-day window X_1, this is:

$$\bar{F}_{X_1} = \frac{1}{n_1} \sum_{x \in X_1} f(x),$$

with n_1 being the number of elements in X_1

To obtain a single measure, in each simulation iteration s, a time window was chosen uniformly at random from $X_1 \dots X_W$. The estimation performance was assessed for this time window only, and the mean performance over all S simulations was reported.

TTR was calculated by dividing the number of BP values in F_X within the target range by the total number of values in F_X:

$$\frac{1}{n} \sum_{x \in X} \mathbb{1}\{\ 90 < f(x) < 125\}$$

5.4 Blood Pressure Time Series Simulation

For simulating the blood pressure time series, the goal is to match the properties described in section 1.2. Simulation starts by generating the true BP time series process, $f(x)$. This process is then sampled at the desired time points X to obtain F_X. Finally, noise is added to obtain Y_X.

The true BP process $f(x)$ is modeled by a Gaussian process (true GP) since GPs are flexible enough to represent the properties specified for $f(x)$ in section 1.2.1.

5.4.1 Mean function

A reasonable assumption for the mean function is to keep it constant and equal to the global mean BP value of 120 mmHg. We have:

$$f(x) \sim GP(120, k(x, x'))$$

From section 4.4, we know that this is the same as writing:

$$f(x) - 120 \sim GP(0, k(x, x'))$$

For simplicity, we are going to completely ignore this constant mean function throughout the rest of the thesis and model the true BP process $f(x)$ with the following GP:

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

where we write $f(x)$, although we actually mean $f(x) - 120$.

5.4.2 Kernel function

The chosen kernel function to match the properties from section 1.2.1 is:

$$k(x, x') = 2.24^2 * \text{Matérn}(l = 3, \nu = 0.5) + 14^2 * \text{Periodic}(l = 3, p = 24) + 2.24^2 * \text{RBF}(l = 50)$$

where l denotes the length scale, and p denotes the periodicity of the corresponding kernel function in hours. The formal definition of the Matérn, Periodic, and RBF kernel functions and their parameters is provided in section 4.5.

Each of these kernels models one of the components described in 1.2.1:

- The Matérn kernel with $\nu = 0.5$ models the AR(1) component
- The Periodic kernel models the circadian cycle
- The RBF kernel models a long-term trend

The kernel function is illustrated in figure 5.1, and some samples drawn from this GP are shown in Figure 5.2.

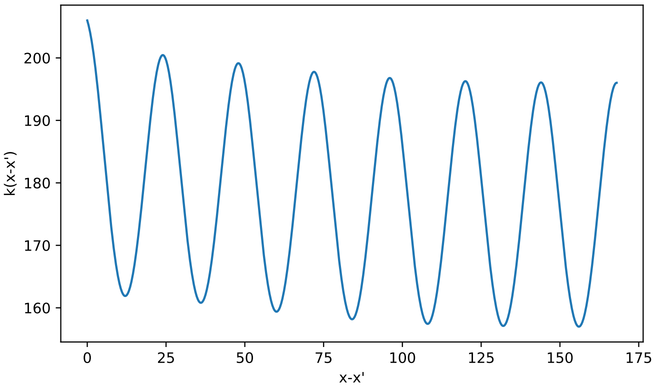


Figure 5.1: The true kernel function $k(x, x')$

5.4.3 Simulation of the BP Measurements

The BP measurements time series process $Y(x)$ is obtained by adding iid measurement noise $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$ to $f(x)$. The measurement noise variance σ_n^2 is set to 31 mmHg², as explained in subsection 1.2.1. The measurement indexes X_{train} are then chosen from X, yielding the training data, $Y_{X_{train}}$. The different downsampling patterns used to produce the training data are described in section 5.7.

Appendix A.2 additionally presents the distributions of some simulated BP measurement properties.

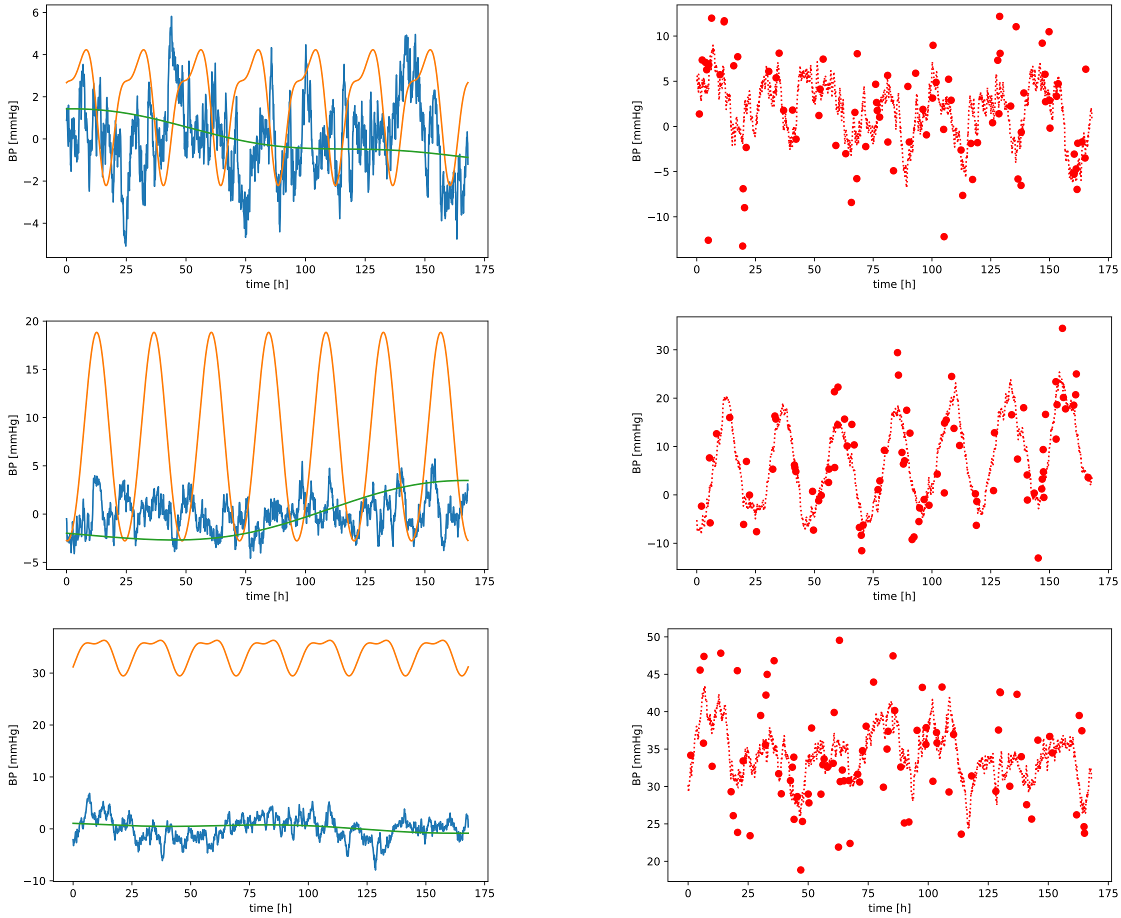
5.5 Gaussian Process Regression

A Gaussian process regression was fitted to $Y_{X_{\text{train}}}$ to estimate F_X. The kernel function used has the same form as the one used for simulation but with variable hyperparameters:

$$k(x, x') = \sigma_M^2 \cdot \text{Matérn}(l, \nu = 0.5) + \sigma_P^2 \cdot \text{Periodic}(l, p = 24) + \sigma_R^2 \cdot \text{RBF}(l)$$

The hyperparameters, σ_M^2, σ_P^2, σ_R^2, and l, were found by maximizing the marginal likelihood, as described in subsection 4.7.1, yielding the optimal kernel $\hat{k}(x, x')$.

From $\hat{k}(x, x')$, the predictive distribution over F_X was calculated. Furthermore, the target measures, including credible intervals, were estimated by sampling from the predictive distribution. Pseudocode 5.5 describes the entire simulation flow for evaluating the performance of GP regression for a specific target measure.



(a) The sample F_X shown to the right, decomposed in to the contribution of the Periodic kernel (orange), Matérn kernel (blue), RBF kernel (green).

(b) Each figure shows one sample F_X drawn from the true GP (red dashed line) with noisy observations (red dots) sampled at a frequency of 0.5/hour

Figure 5.2: Three samples (right side) drawn from the true GP and the decomposition of theses samples (left side)

5.6 Baseline Methods

Some other methods were fitted to $Y_{X_{train}}$ as a reference, to which the GP performance was compared to. The chosen baseline methods presented in this section are: linear regression, smoothing spline, overall mean, and naive TTR. All methods, but naive TTR, estimate the target measure through the estimation of F_X. The calculation of the target measure and confidence interval is described in Pseudocode 5.6. The procedure is equivalent to GP regression, except that one does not sample from the posterior distribution but uses bootstrap samples instead.

Algorithm 1 Simulation and Evaluation Flow

```
Inputs:
$X$                                  ▷ The regression input
$K_{XX}$                             ▷ The true kernel function evaluated at the input $X$
TargetMeasure                        ▷ Function to extract target measure from $F_X$ or $\hat{F}_X$
Output:
CiCoverage                           ▷ Credible interval coverage
CiWidth                              ▷ Credible interval width

Initialize: CiCoverageList = [ ], CiWidthList = [ ],
for $s = 0 \dots S$ do
    $F_X$ = sample from $\mathcal{N}(0, K_{XX})$                ▷ Sample from the true GP
    $X_{train} \subset X$                                       ▷ Choose training indexes
    $Y_{X_{train}} = F_{X_{train}} + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma_n^2)$
    $\hat{k}(x, x') = $ GP.fit$(X_{train}, Y_{X_{train}})$      ▷ Find the optimal kernel
    $\hat{F}_X = \hat{K}_{XX_{train}}(\hat{K}_{X_{train}X_{train}} + \sigma_n^2 I)^{-1} Y_{train}$    ▷ predictive mean
    $\hat{\Sigma}_{F_X} = \hat{K}_{XX} - \hat{K}_{XX_{train}}(\hat{K}_{X_{train}X_{train}} + \sigma_n^2 I)^{-1} \hat{K}_{X_{train}X}$    ▷ predictive covariance
    Initialize: $\hat{M} = [\ ]$
    for $k = 0 \dots K$ do
        $\hat{F}_{X,k}$ = sample from $\mathcal{N}(\hat{F}_X, \hat{\Sigma}_{F_X})$    ▷ Sample from predictive distribution
        $\hat{M}$.append(TargetMeasure($\hat{F}_{X,k}$))         ▷ Extract target measure
    end for
    $m$ = TargetMeasure($F_X$)                                  ▷ Extract true target measure
    $\hat{m}$ =mean($\hat{M}$)
    $ci\_lb = (2\hat{m} -$ quantile$_{1-\alpha/2}(\hat{M})$     ▷ Credible interval lower bound
    $ci\_ub = (2\hat{m} -$ quantile$_{\alpha/2}(\hat{M}))$      ▷ Credible interval upper bound
    CiCoverageList.append($ci\_lb \leq m \leq ci\_ub$)
    CiWidthList.append($ci\_ub - ci\_lb$)
end for
CiCoverage = mean(CiCoverageList)
CiWidth = mean(CiWidthList)
```

5.6.1 Linear Regression

The model used has already been presented in section 3.1 and it features a linear trend and seasonal component:

$$Y(x) = \beta_0 + \beta_1 x + \beta_2 \cos(2\pi f x) + \beta_3 \sin(2\pi f x) + R(t),$$

where f, the frequency, is known and equals 1/period = 1/24.

The seasonal component has variable phase shift and amplitude. Ordinary least square regression has been fit to the training data $(X_{train}, Y_{X_{train}})$ to obtain the regression coefficients and thus $\hat{F}_X$.

5.6.2 Smoothing Spline

The scikit-learn Python package has been used to generate smoothing splines for predicting F_X. First, X_{train} and X were transformed to cubic B-splines using the

Algorithm 2 Target Measrue Estimation with CI

```
Inputs:
$X$                                  ▷ The regression input
$F_X$                                ▷ True BP values at inputs $X$
$X_{train}, Y_{X_{train}}$           ▷ The training data
RegressionMethod                     ▷ The baseline method
TargetMeasure                        ▷ Function to extract target measure from $F_X$ or $\hat{F}_X$
Output:
CiCoverage                           ▷ Credible interval coverage
CiWidth                              ▷ Credible interval width

Initialize: $\hat{M} = [\,]$
for $k = 0 \ldots K$ do                                          ▷ K bootstrap iterations
    $X^* =$ sample with replacement from $X_{train}$
    $\hat{F}_{X,k} =$ RegressionMethod.fit($X^*, Y_{X^*}$).predict($X$)
    $\hat{M}$.append(TargetMeasure($\hat{F}_{X,k}$))               ▷ Extract target measure
end for
$m =$ TargetMeasure($F_X$)                                       ▷ Extract true target measure
$\hat{m} =$mean($\hat{M}$)
$ci\_lb = (2\hat{m} -$ quantile$_{1-\alpha/2}(\hat{M})$)          ▷ Credible interval lower bound
$ci\_ub = (2\hat{m} -$ quantile$_{\alpha/2}(\hat{M})$)            ▷ Credible interval upper bound
CiCoverage $= ci\_lb \leq m \leq ci\_ub$)
CiWidth $= ci\_ub - ci\_lb$)
```

”sklearn.preprocessing.SplineTransformer” class. Knots have been placed uniformly along the quantiles of X_{train}. Ordinary least square regression is then fit to the transformed training input $Xtrans_{train}$ and the response Y_{train}. The number of knots determines the smoothness of the resulting function, and the optimal number has been identified through 10-fold cross-validation. The code section 5.6.2 provides more implementation details, and figure 5.3b shows an example of $\hat{F}_X$ estimated from training data using smoothing splines.

5.6.3 Overall Mean

This method sets $\hat{F}_X$ to the mean of all measurements $Y_{X_{train}}$ everywhere.

5.6.4 Naive TTR

This method directly estimates the target measures from the noisy measurements $Y_{X_{train}}$, without estimating F_X first. The one-hour, one-day, and one-week means were calculated by taking the mean of the available measurements within the time period. If no measurements are available within that period, the mean overall measurements were used.

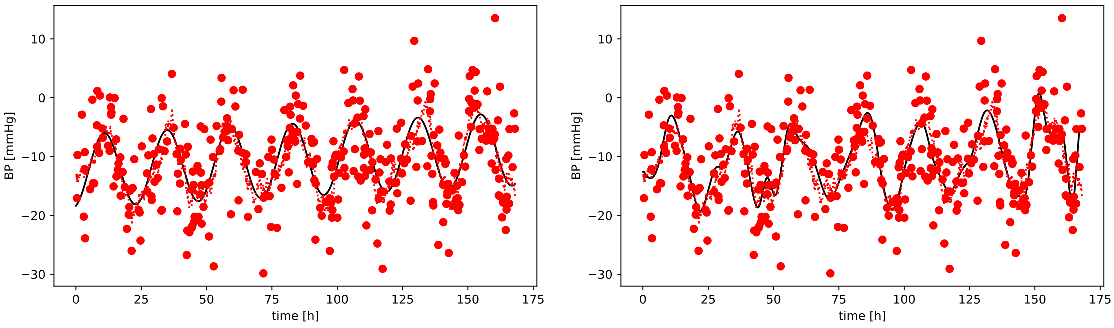
For calculating TTR, the number of measurements within the range over the total number of available data points.

5.7 Adversarial Analysis

Finally, we want to study the impact of the sampling pattern on the target measure estimates. As described in section 1.2.1, the data density is expected to vary within

```python
import numpy as np
from sklearn.preprocessing import SplineTransformer
from sklearn.linear_model import LinearRegression

def fit_and_predict_smoothing_spline(
                X_train: np.ndarray,Y_X_train: np.ndarray, X: np.ndarray,
                n_knots: int) -> np.ndarray:
    """
    Parameters
    ----------
    X_train, Y_X_train: The training data
    X: The time indexes at which to generate predictions
    n_knots : Number of knots of the splines

    Returns
    ---------
    F_X_hat: The BP value predictions at inputs X
    """

    spline = SplineTransformer(degree=3, n_knots=n_knots,
                               extrapolation="constant",
                               knots="quantile")
    # Compute knot positions of splines.
    spline.fit(X_train)
    # Transform to B-splines
    Xtrans_train = spline.transform(X_train)
    Xtrans = spline.transform(X)

    # Fit a linear regression
    lm = LinearRegression(fit_intercept=False).fit(Xtrans_train, Y_X_train)

    # Predict BP values at inputs X
    F_X_hat = lm.predict(Xtrans)
    return F_X_hat
```

Listing 1: Smooting Spline Estimation of F_X



(a) Linear Regression

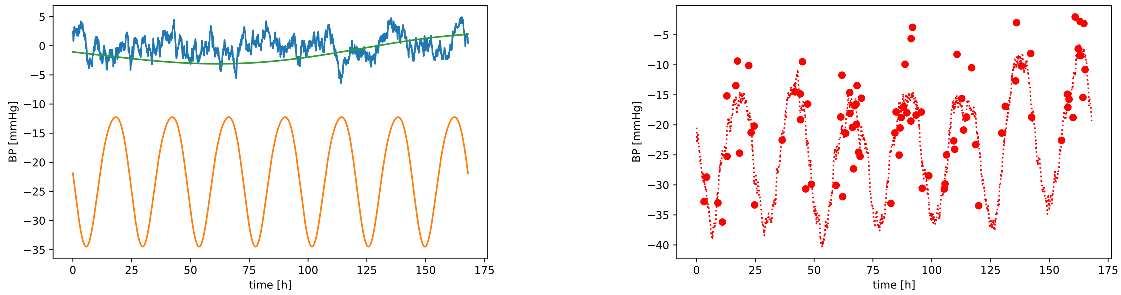
(b) Smoothing Spline

Figure 5.3: Linear Regression and Smoothing spline used to estimate some example of true BP values F_X. The estimated BP values $\hat{F}_X$ (black line), the true BP values (red dashed line) and the training data (red dots).

the Aktiia population, and measurements might also not be sampled uniformly, but data density might follow the circadian cycle. We will refer to the latter phenomenon as seasonal sampling.

The degrees of data density investigated were: 0.5, 1, 2, and 4 measurements per hour.

Seasonal sampling was achieved by extracting the true seasonal component from the true BP sample. The values of the seasonal components were used as probability weights when choosing X_{train} from X. The decomposed seasonal component and the resulting seasonal sampling are illustrated in figure 5.4.



(a) The decomposition of the true BP values $f(x)$. The periodic component used as probability weights in seasonal smapling is shown in orange

(b) The true BP values $f(x)$ (red dashed line) and the noisy measurments (red dots) generated from seasonal sampling.

Figure 5.4: Panel b shows the sample $f(x)$ drawn from the true GP with measurments generated by seasonal sampling. Panel a shows $f(x)$ decomposed into its components.

5.8 Computational Frameworks

All code has been written in Python. For Gaussian process simulation and regression, for fitting the Smoothing Spline and Linear Regression, the Pyton package scikit-learn has been used.

Chapter 6

First Chapter

6.1 To include a picture

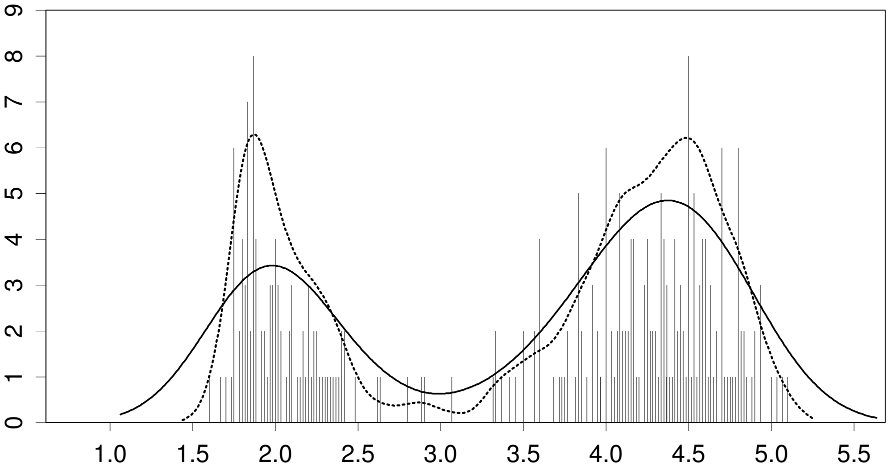


Figure 6.1: Old Faithful Geyser eruption lengths, $n = 272$; binned data and two (Gaussian) kernel density estimates ($\times 10$) with $h = h^* = .3348$ and $h = .1$ (dotted).

Or also with `includegraphics`:

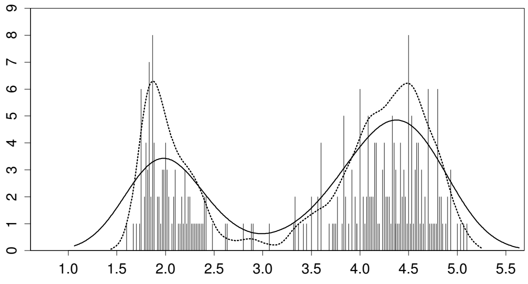


Figure 6.2: Old Faithful Geyser eruption lengths, $n = 272$; binned data and two (Gaussian) kernel density estimates ($\times 10$) with $h = h^* = .3348$ and $h = .1$ (dotted).

6.2 To make a proof

Proof. $1 + 1 = 2$ □

6.3 To include R code

See information in Appendix A.

6.4 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 7

Summary

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

7.1 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.

Duvenaud, D. (2014, June). *Automatic Model Construction with Gaussian Processes.* Doctor of Philosophy, University of Cambridge.

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 Ornstein-Uhlenbeck Process

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 4.5.2.2, 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 $b = \sigma_w\sqrt{\Delta t}$

A.2 Properties of the Simulated Time Series Samples

This section presents the distribution of some crucial properties from the simulated BP time series. These histograms have been created by drawing 100 samples from the true GP.

The shown property distributions should match those from Section 1.2.1.

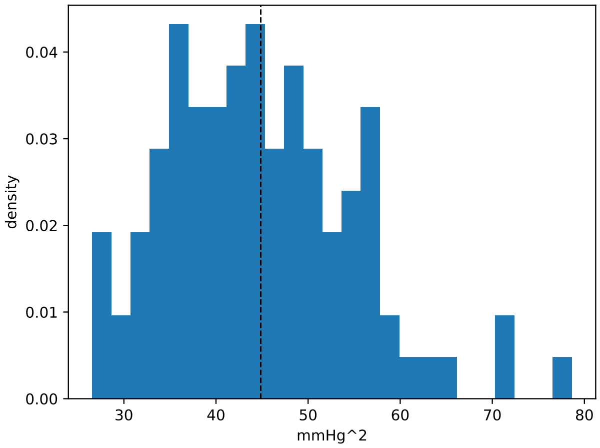


Figure A.1: The one-week sample variance should span from from 16 to 144 mmHg^2, with an average of 49 mmHg^2

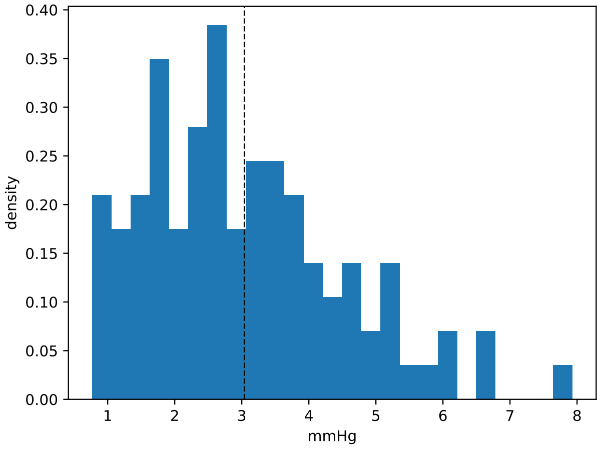


Figure A.2: Half of the difference between average daytime and nighttime BP measurements. Should fall within the range of 0 to 10 mmHg

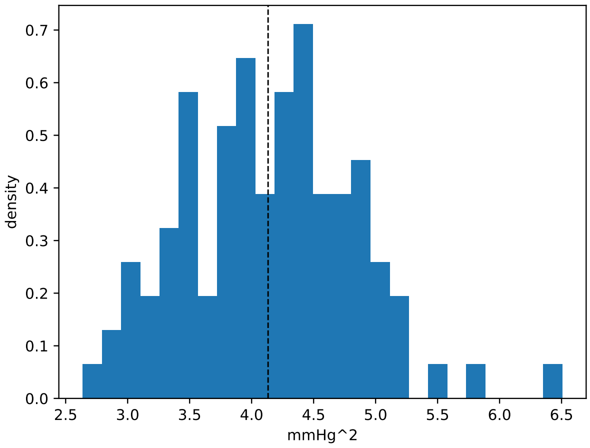


Figure A.3: The variance of the AR(1) component. There exists no target values for this.

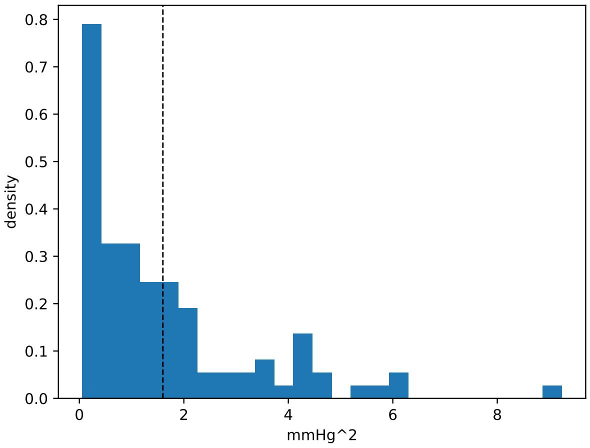


Figure A.4: The variance of the RBF component. There exists no target values for this.

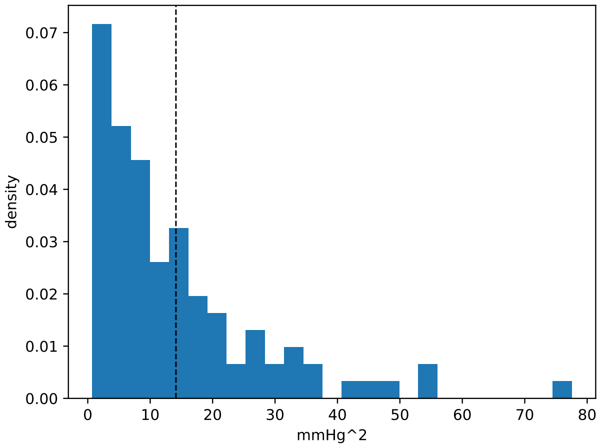


Figure A.5: The variance of the periodic component.

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:	**Signature(s):**
Zurich August 19th 2009	bla

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