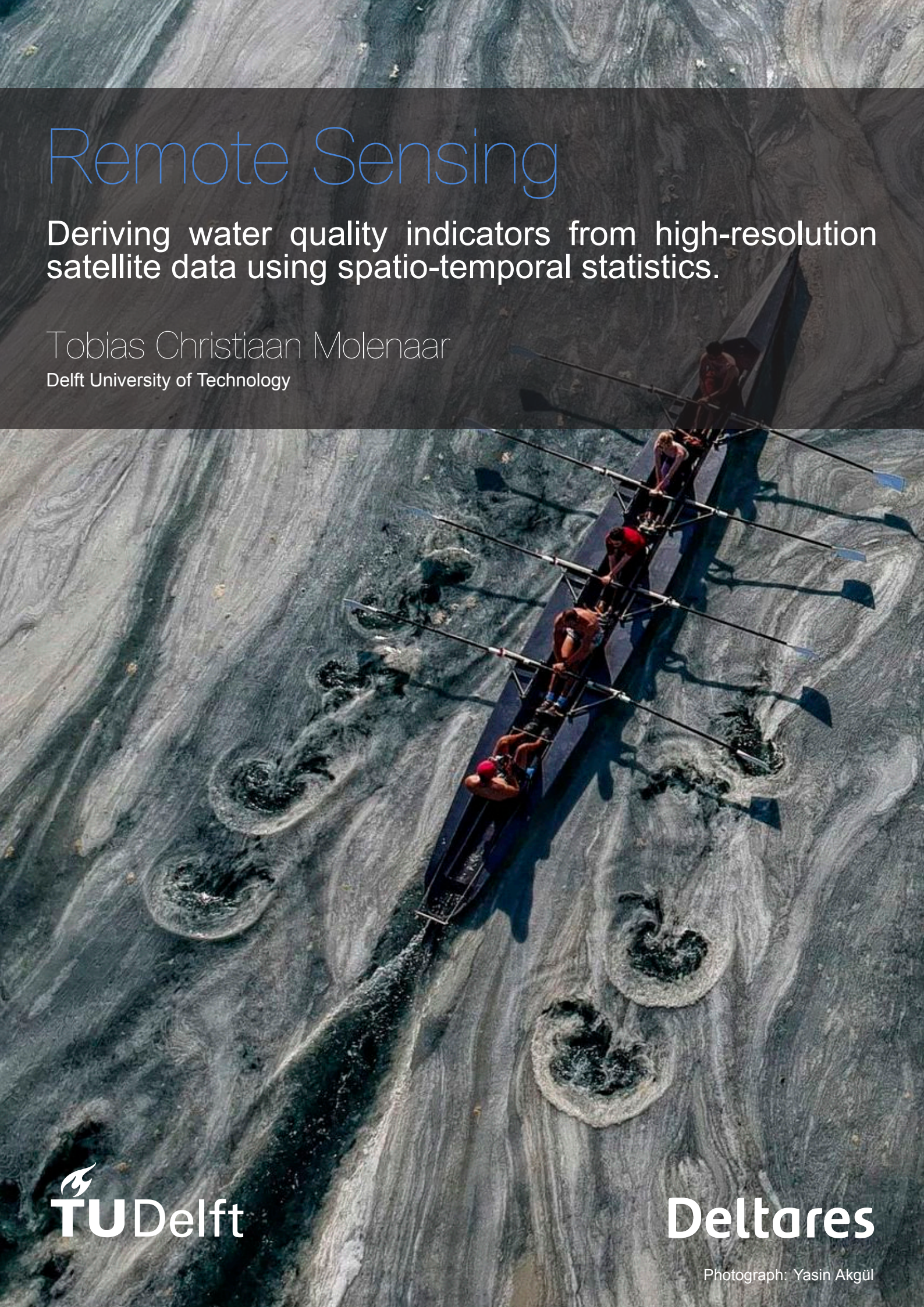
Remote Sensing

Deriving water quality indicators from high-resolution satellite data using spatio-temporal statistics.

Tobias Christiaan Molenaar
Delft University of Technology



TU Delft

Deltares

Photograph: Yasin Akgül

Remote Sensing

Deriving water quality indicators from high-resolution satellite data using spatio-temporal statistics.

by

Tobias Christiaan Molenaar

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday September 9, 2021.

Student number:	4372859	
Faculty:	EEMCS	
Master Programme:	Applied mathematics	
Specialization:	Stochastics	
Project duration:	December 1, 2020 – September 9, 2021	
Thesis committee:	Dr. ir. F.H. van der Meulen,	TU Delft, supervisor
	Prof. Dr. H. M. Schuttelaars,	TU Delft
	L. Mészáros MSc.,	Deltares, supervisor
	A. Spinosa MSc.,	Deltares, supervisor

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft Delft University of Technology

Deltares

Abstract

The derivation of water quality indicators is of importance, especially in coastal areas, as most of the economic activities are located here. However, the availability of high-spatial-resolution water quality information in coastal zones is limited. Nowadays, high-resolution satellite data is becoming available and can fill in this knowledge gap. This satellite data contains spectral reflectances, so a model needs to be designed to map these reflectances to water quality indicators. In this thesis, a Gaussian process regression (GPR) method will be introduced and analyzed extensively in terms of covariance functions, hyperparameters and computational costs. Remote sensing data is collected from the Sentinel-2 mission and the in-situ data is obtained from the ODYSSEA programme. The Matérn 3/2 kernel produces the best results and these are compared with the current models that rely on machine learning techniques. GPR shows promising results in terms of estimation accuracy and chlorophyll-a maps are made for different areas and depths. Various approximation methods are tested to speed up the computation time. Singular value decomposition shows promising results for doing predictions to reduce the computation time. Moreover, GPR can handle limited availability of in-situ data well and uncertainty quantification is induced by the Bayesian framework.

Keywords: Gaussian process regression (GPR); chlorophyll-a; high-resolution satellite data; multispectral; covariance functions

Preface

Dear reader,

Here it is, my final deliverable of the master Applied Mathematics. At the time I started my bachelor of mathematics, seven years ago, I would not have thought that mathematics can be used to estimate water quality estimators. I have learned a lot in the past years, even about fields in mathematics that I did not know existed. In the final year of my masters, I got the opportunity to see the possibilities of applied mathematics at Deltares.

I would like to thank Deltares for letting me do my master thesis within the Data Science and Water Quality team. Although I could not visit Deltares the entire year, I felt part of the team and we had some interesting and fun events.

I am grateful for my supervisory team, who helped me during the past nine months with interesting discussions and feedback. Thanks Anna, for your help in understanding the water-related aspects of this thesis and your critical questions of my findings. Thank you Lörinc, for your positivity and enthusiasm to get the most out of my thesis. Luckily, we could meet each other outside with some drinks as all our meetings were online. Thanks Frank, for your mathematical input and the desire to leave not a single stone unturned.

Many thanks to my friends and family who were always there when I needed them. Thanks for the laughs and willingness to listen to my struggles and achievements. In particular, Nathalie, who proofread every single word and for being there when things got tough.

Enjoy reading!

Tobias Christiaan Molenaar
Delft, September 2021

Contents

List of Figures

List of Tables

List of Abbreviations

AC	Atmosphere Correction
ANN	Artificial Neural Network
ASI	Agenzia Spaziale Italiana
AWI	Alfred Wegener Institute
BOA	Bottom of Atmosphere
C2RCC	Case-2 Regional CoastColour
CDOM	Coloured Dissolved Organic Matter
CMEMS	Copernicus Marine Environment Monitoring Service
DCM	Deep Chlorophyll Maximum
EMODNet	European Marine Observation and Data Network
EO	Earth Observation
EP	Expectation Propagation
ESA	European Space Agency
FIR	Far InfraRed
FPS	Frames per Seconds
GPR	Gaussian Process Regression
GSD	Ground Sample Distance
HS	HyperSpectral
IDW	Inverse Distance Weighting
IOC	Intergouvernmental Oceanographic Commision
IOP	Inherent Optical Property
KOMPSAT	Korean Multi-Purpose Satellite
LEO	Low Earth Orbit
LOOCV	Leave-One-Out Cross-Validation
LOWESS	Locally Weighted Scatterplot Smoothing
MARUM	Center for Marine Environmental Sciences
MBR	Maximum Band Ratio

MERIS	MEdium Resolution Imaging Spectrometer
MIR	Mid InfraRed
MLP	Multilayer Perceptron
MONGOOS	Mediterranean Oceanography Network for the Global Ocean Observing System
MS	MultiSpectral
MSE	Mean Squared Error
NDVI	Normalized Difference Vegetation Index
NetCDF	Network Common Data Form
NIR	Near-InfraRed
NN	Neural Network
NNI	Nearest Neighbour Interpolation
Pan	Panchromatic
PRISMA	PRecursore IperSpettrale della Missione Applicativa
RBF	Radial Basis Function
RGB	Red-Green-Blue
SNAP	Sentinel Application Platform
SPOT	Satellite Pour l'Observation de la Terre
SVM	Support Vector Machine
SVR	Support Vector Regression
SWIR	ShortWave Infrared
TAP	Trans National Data Access Platforms
TOA	Top of Atmosphere
UV	UltraViolet

1

Introduction

Since the Soviet Union launched Sputnik-1 on 4 October 1957, about 10,490 satellites have been put into Earth orbit (European Space Agency, 2020). Several applications of today's satellites are astrophysics, communication, navigation and Earth observation (EO). A few hundred Earth remote sensing satellites (developed for EO) are currently in Earth orbit and are collecting over 150 terabytes of data every day (European Space Agency, 2019). The information obtained by Earth remote sensing satellites has widely been used to estimate ecological indicators in both terrestrial and marine ecosystems (Mishra and Mishra, 2012; Sishodia et al., 2020). However, despite recent improvements, the availability of high-spatial-resolution water quality information in coastal zones is limited (Almeida et al., 2019; Kabbara et al., 2008). According to Liew et al. (2011), it is of importance to study the water quality of (non-)coastal waters using high-resolution satellite sensors, since most of the economic activities are located in coastal areas. Therefore, to satisfy the data requirements for high-resolution information on coastal zones, the derivation of water quality indicators is needed. The newly available high-resolution optical data can fill this knowledge gap. However, these datasets often contain spectral reflectances only (wavelengths of the reflected light in various bands). For that reason, a model needs to be designed to map these reflectances to water indicators.

A large number of models is already available (Hooker et al., 2000; O'Reilly et al., 1998; O'Reilly and Werdell, 2019), however, these models are designed for specific sensors, so for new satellites (i.e. new sensors) the models need to be recalibrated. Furthermore, the majority of these models are designed in such a way that they are fast in computational time as there is a lot of data to handle. More complex models, e.g. neural networks (Brockmann et al., 2016; Lee et al., 1998), are used for the Copernicus Sentinel 2 and 3 missions. However, the specific information about these models is not available and they require fine-tuning for the particular sensors.

This research was carried out at Deltares, which is an independent research institute that mainly focuses on applied research in water, subsurface and infrastructure. Deltares designs, develops, manages and maintains both software and facilities to simulate nature-related events, such as the Delta Flume and Delft3D. Working together with governments, universities, businesses and other research institutes makes Deltares an expert in describing chemical and ecological processes. This research is part of the HiSea project, which is funded by the European Union as part of the Horizon Europe program (under grant agreement ID: 821934). The HiSea project aims to develop, test and demonstrate information services that provide high-resolution data of water quality at sea. The vision of the project is to build a platform that creates the opportunity for better efficiency and productivity in marine-related businesses, while at the same time the marine and coastal environment are preserved and able to thrive.

The goal of the HiSea project is to supply meaningful data to gain a better understanding of the marine environment and to better predict future events. Finally, the ODYSSEA project is consulted for their available datasets. ODYSSEA is also a Horizon 2020 project funded by the European Union.

1.1. Problem Description

In this research, the water quality indicator that is focused on will be chlorophyll-a. Chlorophyll originates from the Greek language and literally translated means “green leaf” (χλωρός which means green and φύλλον which means leaf). The pigment chlorophyll is present in all green plants and gives them their green color. It is also responsible for the absorption of light to create energy for photosynthesis (Petruzzello, 2020). Chlorophyll-a ($C_{55}H_{72}O_5N_4Mg$) is a specific form of chlorophyll and is found in all photosynthesizing plants, algae and phytoplankton. Usually, a high concentration of chlorophyll-a indicates poor water quality, conversely a low concentration suggests good conditions. This can be explained by considering a high concentration of chlorophyll-a which indicates, for example, a high concentration of algae. A rapid increase or accumulation of algae can result in algal blooms that cause shade and rapid changes in dissolved oxygen which causes a reduction in fish populations and plant diversity. This process is called *eutrophication* (ευτροφία in greek means well-nourished) which is a worldwide problem nowadays. The recent example for this is the ‘sea-snot’ that plagues the Turkish coasts and endangers the fishery in the Marmara sea (cover image by Akgul (2021)) (Uğurtaş, 2021). Chlorophyll-a has a ‘natural’ cause to change in concentration which is seasonality, see Figure 1.1. Due to the rise in temperature and more availability of sunlight in summer, the chlorophyll-a concentration is able to increase. Note that the average concentration in summer can sometimes change quickly, where there are periods where algae bloom and die in rapid succession. For example, in April there is a high concentration of chlorophyll-a in the North Sea, which then suddenly drops due to a lack of nutrients. However, when the concentration drops so drastically, a lot of nutrients are available again and thus the concentration can rise.

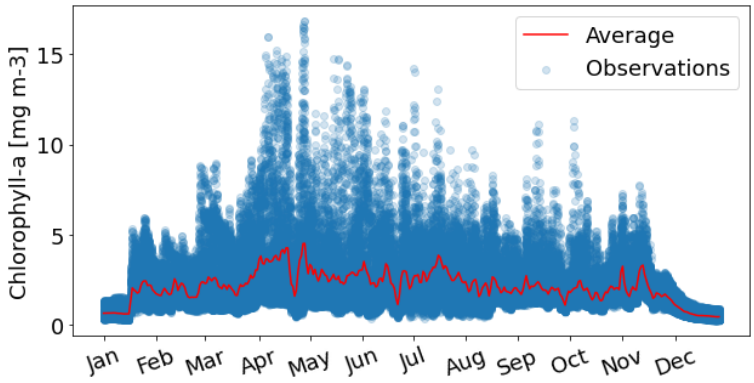


Figure 1.1: Chlorophyll-a concentration observations in the North Sea in 2006 including the daily average (red). Data obtained from the MEdium Resolution Imaging Spectrometer (MERIS).

Another cause of algae blooms and therefore eutrophication is the excessive human activities such as overdevelopment of watersheds and agricultural irrigation (Anderson et al., 2002). Moreover, nutrients are added to waters as a result of domestic and industrial wastewater that is partially treated. This has not only an impact on the marine-life and plant diversity, but also on the fishery and tourism (Ho et al., 2019). As a consequence, it is interesting to study the chlorophyll-a concentration to be able to prevent events such as in the Marmara sea and to

improve the water quality. Furthermore, the impact of nutrients that flush into the sea by rivers can be studied and the source can be detected (especially using high-resolution satellite data).

Using remote-sensing by satellites, it is possible to cover the complete Earth and obtain data using the sensors onboard. The color of water appears to be (dark) blue to the naked eye since most of the other wavelengths are absorbed significantly more, whereas the blue wavelength is scattered more by the water molecules. The distinction between clear water and water containing chlorophyll-a is the strong absorption of the blue and red wavelength and the lower absorption of the green wavelength in water containing chlorophyll-a. This suggests that it is possible to estimate the chlorophyll-a concentration using satellite imagery.

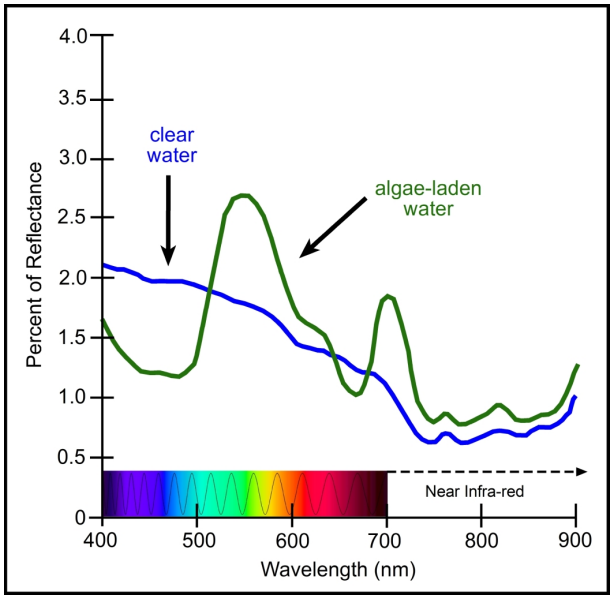


Figure 1.2: Percent reflectance of clear (blue) and algae-laden (green) water (Han, 1997; Klose et al., n.d.).

Some problems arise with satellite imagery, as the radiance recorded by the sensor on board of the satellite will typically not be the true radiance from the water. First of all, the measurements can be disturbed by the presence of clouds, not to mention the day and night cycle which causes the observations to not be useful. Secondly, when the circumstances are good (i.e. no clouds), the radiance is still disturbed by the atmosphere, components in the water and radiance that reach the bottom of the water (typically in coastal waters). Finally, the angle of the sunlight and the wind that causes waves influence the radiance. Hence, before any analysis can be done, the data needs to be preprocessed to obtain the 'correct' radiance reflected by the water. Some part of this preprocessing is done manually in this thesis, though some of the preprocessing has been done by using a higher level data-product.

The remote-sensing data will be in the form of a grid, where the in-situ data will be available only in very specific locations. Ideally, these locations match, though this will not be possible in reality. This creates the problem of having the dependent and explanatory variables on different locations, so some sort of interpolation or kriging is necessary to do regression. Moreover, these algorithms are often evaluated on individual or small population of lakes and coastal waters where in-situ measurements are available. Therefore, these algorithms can fail in predicting water quality estimators globally due to the different optical properties of the water columns. Finally, the model needs to incorporate time as well as the location, so a spatio-temporal model will be constructed. Since the availability of high-resolution satellite data, the amount of data has increased significantly. For that reason, any model needs to be able to process a large amount of data in a reasonable time.

1.2. Research questions

The main research question for this project is: *What spatio-temporal model, that provides uncertainty quantification, can be used to estimate the chlorophyll-a concentration using high-resolution optical remote sensing data?* I will try to find an answer to this question by answering the following sub-questions:

- How can we obtain and preprocess spectral reflectance data?
- How are the spectral reflectances related to the chlorophyll-a concentration?
- What are the current models to estimate the chlorophyll-a concentration?
- How does the time of year influence the chlorophyll-a concentration?
- How can we incorporate the spatial correlation in our model?
- What is the influence of depth in relation with the chlorophyll-a concentration?
- How does our proposed model compare to the state-of-the-art models?
- What approximation method is best for fast computation?

1.3. Methodology

To be able to answer the sub-questions and the main research question, this study will use remote sensing data as well as in-situ data to train, test and validate the models that are applied. The remote-sensing and in-situ data are obtained from the corresponding platforms, though, some have been obtained from Deltares. For the satellite data, a tool called the Sentinel Application Platform (SNAP) is used to import, edit and export data (SNAP, 2020). Converting the data into a netCDF (network common data form) format makes it possible to import the satellite data into Python (Van Rossum and Drake Jr, 1995). Then, an explanatory analysis can be applied in order to attempt to answer the raised questions. Furthermore, a spatio-temporal model will be introduced and investigated thoroughly to be able to answer the main question. The produced code can be found on GitHub[1].

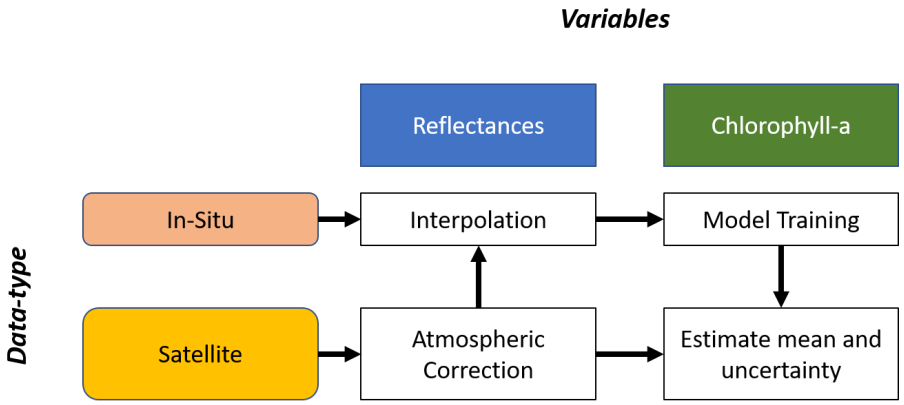


Figure 1.3: Schematic overview of the steps necessary to obtain chlorophyll-a estimations. First, the raw satellite data needs to preprocessed by an atmospheric correction algorithm. An interpolation method can be applied to retrieve the reflectances at in-situ locations. In combination with the chlorophyll-a measurements, a model can be trained with the in-situ data. Finally, an estimation and uncertainty quantification can be computed at for every satellite observation.

A schematic view of the steps in this thesis are shown in Figure 1.3. First, the raw satellite data needs to be corrected by an atmospheric correction algorithm (manually or by using higher level products). With the obtained reflectances and using the coordinates from the

[1] https://github.com/tobimoli/MSc-Thesis

in-situ data, the reflectances at the in-situ locations can be computed with an interpolation method. Then, in combination with the obtained chlorophyll-a measurements in the in-situ data, a model can be trained to learn/find relationships between the reflectances and the chlorophyll-a concentration. Once trained, the model can be applied to the reflectances of the satellite data and an estimation including an uncertainty quantification can be computed at every (satellite) observation.

1.4. Outline

In this thesis, a literature review will be done in chapter 2, where background knowledge of satellite imagery is given and the state of the art models are introduced and explained in detail. Then, the process of data collection is described in chapter 3. Both in-situ data and satellite data will be discussed and the data will be preprocessed. Thereafter, the different approaches of spatio-temporal statistics are explained in chapter 4. Here, a Gaussian process regression is introduced and constructed with mathematical detail. This model is then used to analyze different kernels, parameters and variables in chapter 5. For computational purposes, approximation techniques are introduced and applied in chapter 6. In chapter 7 the results of the analysis are summed up and finally, in chapter 8 an answer is given to the main research question as well as the sub-questions. Additionally, the results will be discussed and some recommendations are given for future research.

2

Literature Review

Before diving into the spatio-temporal model that will be introduced in chapter 4, the models that are currently used will be explained and their benefits and drawbacks will be discussed in this chapter. First, some background information on satellite imagery and notation will be given. Then, some of the straightforward methods such as some polynomial models will be reviewed. These methods were introduced a few decades ago. Still, they are relevant as their strength lies in the simplicity and, as a consequence, the computational speed (O'Reilly and Werdell, 2019). Thereafter, the state of the art models are described. Due to the increase in use and interest in machine learning, algorithms such as neural networks and support-vector machines are studied and implemented to retrieve the chlorophyll-a concentration using satellite imagery (Li et al., 2018).

2.1. Satellite Imagery

Satellite imagery refers to images of the Earth, obtained by sensors on board of satellites. There are numerous commercial satellites and satellites owned by governments and therefore several sensors are used to obtain the images. Each of these sensors is specialized in its own way corresponding to its application. The usefulness of the imagery can be described in terms of resolution. Campbell and Wynne (2011) state that there are four different types of resolution to consider.

The most obvious resolution is the *spatial resolution*. This is, simply said, the size of the pixels of the image that is taken and is usually defined in meters. It is not to be confused with the number of pixels, as an image can have more pixels but have a worse spatial resolution. This is a result of having dependent pixels; though, spatial resolution increases when having more independent pixels. It is sometimes referred to as the geometric resolution which can be written in terms of ground sample distance (GSD), meaning the distance between two centers of pixels next to each other.

The capability to observe and record many levels of brightness is described as the *radiometric resolution*. When only a few levels of brightness are used, i.e. coarse radiometric resolution, the image will be of high contrast. With a fine radiometric resolution, the image will contain multiple levels of brightness and can better distinguish the difference of intensity.

The *spectral resolution* can be defined as the ability to capture the image using multiple bandwidths that covers the spectrum of colors. An ordinary camera uses three bandwidths (blue, green and red); however, sensors used for satellite imagery can contain hundreds of fine bandwidths which capture not only the visible spectrum but the non-visible spectrum as well (e.g. infra-red). Based on the spectral resolution, satellite imagering can be subdivided into three categories.

- Panchromatic (Pan) imaging is a way of acquiring data using a single bandwidth that contains multiple hundreds of nanometers, e.g. the spectral range for WorldView-2 is 450-800nm. All information reflected from each pixel is summarized into one value which is the intensity of the reflected solar radiation. As it is just one value, the information from panchromatic images is visualized using black and white images.
- Multispectral (MS) sensors, on the other hand, acquire the information using multiple smaller bandwidths. These sensors provide their data in 2 to 15 bands (Fletcher, 2012). For example, the multispectral sensor from the WorldView-2 mission provides 8 bands, each with a bandwidth of approximately 50nm. As the amount of light energy is relatively small per band in comparison to the panchromatic band, the spatial resolution of multispectral images is worse than the spatial resolution of a panchromatic image. For the WorldView-2 mission, the spatial resolution (at nadir) is 0.46m GSD (Pan) and 1.8m GSD (MS).
- Hyperspectral (HS) imaging, as the name suggests, is a way of acquiring the data using tens or even hundreds of bands with a very small bandwidth (around 1-5nm). As an example, the PRISMA (PRecursore IperSpettrale della Missione Applicativa) mission of ASI (Agenzia Spaziale Italiana) is launched for making medium-resolution hyperspectral images. It uses 66 bands in the VNIR channel and 171 bands in the SWIR channel each with bandwidths smaller than 12nm (Pignatti et al., 2013). For the spectral ranges of these channels, see Figure 2.1 (Fang et al., 2018).

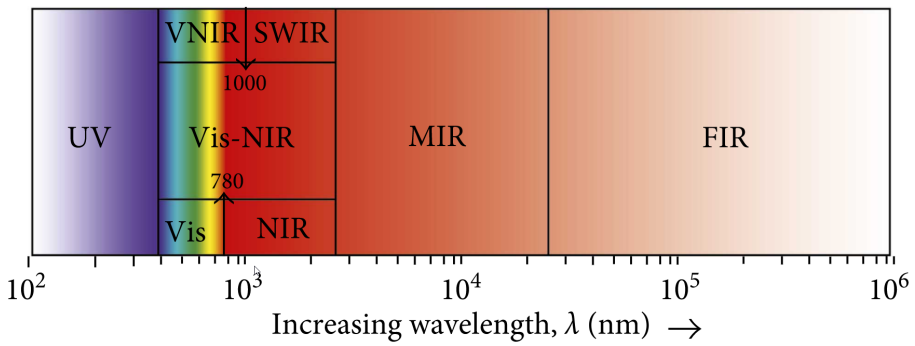


Figure 2.1: The electromagnetic spectrum of wavelength λ ranging from 10^2 to 10^6 nm. From left to right the classified classes are UltraViolet (UV), Visible and Near-Infrared (VNIR), Short-Wave Infrared (SWIR), Mid Infrared (MIR) and Far Infrared (FIR) (Fang et al., 2018).

The fourth resolution is the *temporal resolution*, which is the precision in time. For video cameras, this is usually measured in frames per second (FPS), e.g. the iPhone 8 can shoot in 240 FPS for slow-motion videos. For satellite imagery, it is very complicated to shoot in this temporal resolution considering two reasons: the satellite moves with a high velocity and due to the rotation of the Earth, the satellite has to complete multiple orbits before it reaches the same position again. For high spatial resolution images, a small distance is desired between the satellite and the Earth. Therefore, most of these satellites are placed in a sun-synchronous orbit which is located in the low Earth orbit (LEO) (Boain, 2004; Sellers et al., 2000). The altitude of sun-synchronous satellites is approximately between 150 and 900 kilometers above the surface of the Earth. Furthermore, the orbits are polar orbits which means that the satellites cross the equator multiple times per day. Because the orbit is sun-synchronous, the angle with respect to the sun remains the same which is very useful for capturing images of the Earth's surface. The period of one orbit is around 90 minutes (considering a LEO), however, the temporal resolution will be around one day because of the rotation of the Earth. Also, the swath width must be taken into account for the temporal resolution, as this is the width that

can be observed from the sensor. Furthermore, the weather conditions need to be taken into account so for places where it is regularly cloudy, the effective temporal resolution will very likely be more than one day.

The tradeoff between these resolutions has to be made for each different application. For example, satellite imagery used for studying the weather can have a relatively low spatial resolution, though a high temporal resolution is desired. Therefore, these satellites are orbiting in the geostationary orbit so the satellite is facing the same part of the Earth at any time. Generally, improving one of the resolutions leads to a decrease of one of the other resolutions (Campbell and Wynne, 2011). In our case where multispectral as well as hyperspectral sensors are possible to use, improving the spectral resolution by including more bands principally means that the spatial resolution will deteriorate (Campbell and Wynne, 2011).

The sensors measure the spectral radiance emitted from the top of the atmosphere (TOA) for different wavelengths. An atmospheric correction (AC) algorithm is applied to remove the influence of the atmosphere. Then, the remote sensing reflectances can be written as $R_{rs}(\lambda)[sr^{-1}]$. This is the light exiting the water (water-leaving radiance) normalized to the downwelling solar irradiance (Cannizzaro and Carder, 2006; O'Reilly and Werdell, 2019).

$$R_{rs}(\lambda) = \frac{L_w(\lambda)}{E_d(\lambda)}.$$

The SI-unit steradian [sr] is the dimensionless unit of a solid angle (Ω). Similar to a radian that is related to the circumference of a circle, a steradian is related to the surface of a sphere. Equivalent to the angle that is made of the radius around a circle which is exactly one radian, the solid angle that is made of an area equal to the radius squared around a sphere, is exactly one steradian. As the complete surface of a sphere is equal to $4\pi r^2$ the maximum solid angle is 4π steradian.

Although a large number of models use remote sensing reflectances to estimate the chlorophyll-a concentration and other water quality estimators, inherent optical properties (IOPs) are proposed to be used as well (Gons et al., 2008; Gons et al., 2002). An IOP is an optical property that is fixed, so it is not dependent on the changes in light fields within the water and the atmosphere. These properties, such as the absorption coefficient of water and the backscattering coefficient can be estimated using the remote sensing reflectances (Liew et al., 2011; Woźniak et al., 2019). In-situ measurements of IOPs are also used to estimate the remote sensing reflectances and therefore water quality estimators such as chlorophyll-a and colored dissolved organic matter (CDOM) (Cannizzaro and Carder, 2006; Dall'Olmo and Gitelson, 2005).

In the next section, the simple regression methods are introduced and discussed together with the different choices for ratios of remote sensing reflectances ($R_{rs}(\lambda)$) to estimate the chlorophyll-a concentration.

2.2. Simple Regression

Numerous models have been designed to model the chlorophyll-a concentration using the reflectances (O'Reilly and Werdell, 2019). The majority of the models are a modified cubic polynomial function and can be written as:

$$\log(\text{Chl-a}) = a + bR + cR^2 + dR^3 + eR^4. \tag{2.1}$$

The logarithm is used to avoid negative values for the chlorophyll-a concentration. Here, R is usually the logarithm of the ratio of two bandwidths and a, b, c, d and e are coefficients to fit the model. The most simple models set c, d, e equal to zero (a linear model) (Mishra and Mishra, 2012; Moses et al., 2009; Zhang et al., 2009). All these simple regression methods have the

advantage that they can handle large datasets easily, which is practical as the remote sensing datasets contain usually a large number of observations.

$$R = \log\left(\frac{R_{rs}(\lambda_1)}{R_{rs}(\lambda_2)}\right).$$

The reason why the ratio of remote sensing reflectances is often used is that there is a high correlation between the ratio and the chlorophyll-a concentration, when the optimal wavelengths are used (Cannizzaro and Carder, 2006; Kabbara et al., 2008; Moses et al., 2009; O'Reilly and Werdell, 2019; Tzortziou et al., 2007). The choice of λ_1 and λ_2 can be based on the type of water (i.e. oligotrophic, mesotrophic or eutrophic waters), but also heavily relies on the spectral resolution of the sensor. The maximum band ratio (MBR), introduced by O'Reilly et al. (1998), uses the maximum value of $R_{rs}(\lambda_1)$ where λ_1 is the wavelength of violet-blue ($\sim$400-510nm) and λ_2 is the green wavelength ($\sim$ 550nm). This is done because it has been observed that the chlorophyll-a concentration decreases when the ratio increases (Hooker et al., 2000). In Figure 1.2, it can be seen that the percent of reflectance of the green wavelength is considerably higher for algae-laden water in comparison with clear water and vice versa for the violet-blue wavelength.

Other ratio's of remote sensing reflectances have been used as well, such as a blue-red ratio (Cannizzaro and Carder, 2006; Kabbara et al., 2008), NIR-red ratio (Moses et al., 2009), red-green ratio (Tzortziou et al., 2007) and more complex ratio's (Dall'Olmo and Gitelson, 2005; Mishra and Mishra, 2012; O'Reilly and Werdell, 2019; Zhang et al., 2009). One of these complex ratio's is the normalized difference vegetation index (NDVI):

$$R = \log(NDVI) = \log\left(\frac{R_{rs}(\lambda_{NIR}) - R_{rs}(\lambda_{red})}{R_{rs}(\lambda_{NIR}) + R_{rs}(\lambda_{red})}\right).$$

As $R_{rs}(\lambda)$ is a ratio on itself, it takes values between 0 and 1, and thus NDVI takes values between -1 and 1. NDVI is often used to detect vegetation on land however it is shown by Mishra and Mishra (2012) that this ratio can be used to estimate the chlorophyll-a concentration. The three-banded ratio's proposed by Dall'Olmo and Gitelson (2005):

$$R = \log\left(\frac{R_{rs}(759)}{R_{rs}(690)} - \frac{R_{rs}(759)}{R_{rs}(703)}\right),$$

and Zhang et al. (2009):

$$R = \log\left(\frac{R_{rs}(753)}{R_{rs}(665)} - \frac{R_{rs}(753)}{R_{rs}(708)}\right),$$

have a similar shape, although the specific wavelengths that are used differ because of the available data. Finally, O'Reilly and Werdell (2019) derived 65 algorithms for 25 different satellite sensors of which the majority (62) use a MBR where the wavelengths are specifically chosen based on the satellite sensor. When available, six bands are used to create the MBR where the denominator is the mean of the remote sensing reflectances for the red and green wavelength. For example, for the MEdium Resolution Imaging Spectrometer (MERIS) on board of the Envisat (Environmental satellite) prior to the Sentinel satellites of ESA, the following ratio is proposed:

$$R = \log\left(\frac{\max\{R_{rs}(412), R_{rs}(442), R_{rs}(490), R_{rs}(510)\}}{\frac{1}{2}(R_{rs}(560) + R_{rs}(665))}\right).$$

As said before, the main advantage of these kind of models is that the computational complexity is low. Computing the coefficients in Equation 2.1 can be done using in-situ data of

chlorophyll-a concentration and reflectances. Consider the polynomial regression model $\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\mathbf{y}$ is the response vector, $X \in \mathbb{R}^{(n \times p)}$ the design matrix, $\boldsymbol{\beta}$ the vector containing the coefficients and $\boldsymbol{\varepsilon}$ are the (independent) random errors. The number of coefficients is p and n is the number of observations. Then, using ordinary least squares, the coefficient vector that minimizes the sum of the squared errors, $\hat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \mathbf{y}$. The number of flops (floating point operations) needed to compute $\hat{\boldsymbol{\beta}}$ is of the order: $\mathcal{O}(p^3 + np^2)$. So, it scales linearly with n and we have 5 coefficients in Equation 2.1, so $p = 5$. To compute predictions, $X_{new}\hat{\boldsymbol{\beta}}$ needs to be computed, which takes another $n(2p - 1)$ flops.

Further advantages are the simplicity of the model (i.e. coefficients are easy to interpret) and when the relationship is known to be polynomial, the model is presumed to perform well. However, the choice for the polynomial degree is ambiguous and will be based on the bias-variance tradeoff. Furthermore, outliers are able to influence the model heavily and thus requires some outlier-handling.

In the next section, the state of the art models will be discussed as well as their advantages and disadvantages.

2.3. State of the Art Models

More complex models nowadays are using machine learning algorithms to retrieve the chlorophyll-a concentration from the observed reflectances. The support-vector machine (SVM) is used (Sun et al., 2009; Wang et al., 2018) and Neural Networks (NN) as well (Lee et al., 1998; Li et al., 2018). Again, the ratio of two reflectances is used as a variable in these models. Furthermore, algorithms have been designed specifically for MERIS data (Doerffer, 2015; Doerffer and Schiller, 2007) and is redesigned and renamed into C2RCC (Case-2 Regional CoastColour) such that it is applicable to other sensors as well (Brockmann et al., 2016). This algorithm uses multiple neural networks and individual reflectances as input variables. Another complex model, however not a state of the art model, is a spatio-temporal model represented by a Gaussian process regression (GPR). This model is validated to estimate the chlorophyll-a concentration (Bazi et al., 2014; Pasolli et al., 2010; Verrelst et al., 2012).

Here, the SVM and NN algorithms, as well as the GPR model, will be explained briefly and discussed combined with relevant research.

2.3.1. Support-Vector Machine

Support-vector machine (SVM) is being used for classification problems and regression analysis (Smola and Schölkopf, 2004; Vapnik, 1995) and has been applied extensively for remote sensing research (Wang et al., 2018). Using SVM for regression analysis is also known as support-vector regression (SVR). The basic idea of SVR is to map the input vector into a high-dimensional feature space through a nonlinear mapping whereafter a linear regression problem can be solved in this feature space (Sun et al., 2009). Instead of minimizing the errors, SVR gives the possibility to specify the allowed error and has the objective to minimize the coefficients.

Consider training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}$, for weights $\mathbf{w} \in \mathbb{R}^d$ and bias $b \in \mathbb{R}$ we write the function $f : \mathbb{R}^d \to \mathbb{R}$ as:

$$f(\mathbf{x}) = \mathbf{x}^T \cdot \mathbf{w} + b.$$

Now, the objective is to minimize $\frac{1}{2}||\mathbf{w}||^2$ subject to $|y_i - f(\mathbf{x}_i)| \leq \varepsilon$ where $\varepsilon > 0$ is the allowed error ($||\mathbf{w}||^2 = \mathbf{w}^T \cdot \mathbf{w}$). The constraint of having approximations close to y_i with a maximum error of ε is not always feasible. Therefore, slack variables are introduced to manage approximations having a large error. We denote the deviation from the allowed error by ξ so the

optimization problem becomes:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \quad \frac{1}{2}||\mathbf{w}||^2 + c\sum_{i=1}^{n}|\xi_i| \tag{2.2}$$

$$\text{subject to} \quad |y_i - f(\mathbf{x}_i)| \leq \varepsilon + |\xi_i| \tag{2.3}$$

The constant $c > 0$ can be considered to be a hyperparameter and can be chosen to be small (i.e. $|\xi_i|$ is allowed to be large) or large (i.e. $|\xi_i|$ needs to be small). According to Smola and Schölkopf (2004), this problem turns out to be solved easier when considering the dual problem which ultimately leads to including kernels to make the SVR algorithm non-linear. For more details about the SVR algorithm, please refer to the tutorial on support vector regression (Smola and Schölkopf, 2004).

With the SVR algorithm, it is possible to consider different ratios of water-leaving radiance reflectances as input values and the chlorophyll-a concentration as an output value. Sun et al. (2009) compared the SVR algorithm with simple linear regression and polynomial regression techniques as described in the previous section. The area of interest is Lake Taihu (China), which is an inland turbid lake and only 47 samples are used. Despite the low number of observations, the SVR model performed better than the other regression techniques in terms of root-mean-square error. Wang et al. (2018) showed similar results, though the sample size was small, containing only 39 observations. They state that SVR has high accuracy but depend more on the observed data than the linear/polynomial regression models. Furthermore, "SVR has the advantage to solve small sample, nonlinear and high-dimensional pattern recognition problems" (Wang et al., 2018). Another advantage is that an SVR model can handle outliers by tuning the hyperparameter c. A disadvantage is that an SVR model is harder to interpret than a simple linear regression model. Most noteworthy, the computational and storage requirements scale cubic and quadratic with the number of observations, respectively. However, multiple methods exist to cope with this problem such as sampling, matrix decomposition, chunking and the use of core vector machines that showed linear time complexity and constant space complexity (Tsang et al., 2005).

2.3.2. Artificial Neural Network

An artificial neural network (ANN) is a system based on the biological neural network to solve artificial intelligence problems. ANN is using neurons and their connections between them to solve problems. An input layer and an output layer of neurons is used and if desired, multiple hidden layers are applied. Every layer consists of multiple neurons and every neuron is connected to each neuron in the layer before and after it. The input neurons can be interpreted as the explanatory variables and the output neurons as the response variables. In Figure 2.2 a simple example of an ANN with one hidden layer can be observed. As soon as the input values are known ($a_0^{(0)}, a_1^{(0)}, a_2^{(0)}$), the neurons in the first hidden layer can be computed by the following formula:

$$a_i^{(1)} = \sigma(w_{i,0}^{(1)}a_0^{(0)} + w_{i,1}^{(1)}a_1^{(0)} + w_{i,2}^{(1)}a_2^{(0)} + b_i^{(1)}) = \sigma(z_i^{(1)}).$$

Here, the superscript refers to the layer and the subscripts to the neuron in this particular layer. So $a_i^{(1)}$ is the value for neuron i in layer 1 (first hidden layer), $w_{i,1}^{(1)}$ is the weight assigned to the connection between neuron i in layer 1 and neuron 1 in the input layer. The bias $b_i^{(1)}$ is included for the occurrence that all input values are equal to zero. Finally $\sigma(\cdot)$ is a function, often the sigmoid function, such that all values are transformed into a $[0, 1]$ domain. In matrix notation we get for the first layer:

$$\mathbf{a}^{(1)} = \sigma\left(W^{(1)}\mathbf{a}^{(0)} + \mathbf{b}^{(1)}\right).$$

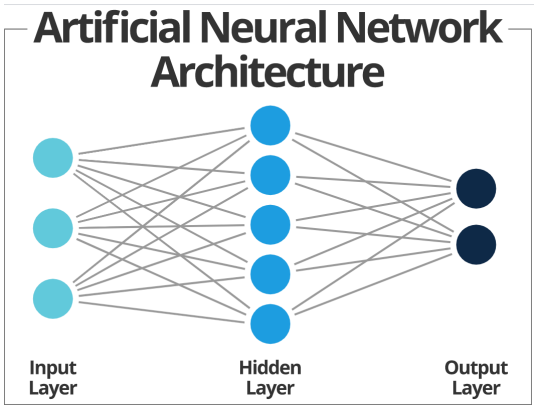


Figure 2.2: Schematic example of an artificial neural network.

For random chosen initial weights and biases, the output variables can be computed given some input values. Since the output will differ from the desired output, a cost will be assigned to this choice of weights and biases. So, learning in neural networks means updating the weights and biases such that the cost minimizes.

Define $\mathbf{y}$ as the desired output and $\mathbf{a}^{(L)}$ as the output of the model, then the cost for one trial can be defined by: $C = \sum_i (a_i^{(L)} - y_i)^2$ (quadratic loss). Now, the change to the weights can be determined by finding the gradient $\nabla C(\mathbf{w})$, therefore the following needs to be computed:

$$\begin{aligned}
\frac{\delta C}{\delta \mathbf{w}^{(L)}} &= \frac{\delta \mathbf{z}^{(L)}}{\delta \mathbf{w}^{(L)}} \frac{\delta \mathbf{a}^{(L)}}{\delta \mathbf{z}^{(L)}} \frac{\delta C}{\delta \mathbf{a}^{(L)}}, \\
&= \mathbf{a}^{(L-1)} \cdot \sigma'(\mathbf{z}^{(L)}) \cdot 2(\mathbf{a}^{(L)} - \mathbf{y}). \\
\frac{\delta C}{\delta \mathbf{b}^{(L)}} &= \frac{\delta \mathbf{z}^{(L)}}{\delta \mathbf{b}^{(L)}} \frac{\delta \mathbf{a}^{(L)}}{\delta \mathbf{z}^{(L)}} \frac{\delta C}{\delta \mathbf{a}^{(L)}}, \\
&= 1 \cdot \sigma'(\mathbf{z}^{(L)}) \cdot 2(\mathbf{a}^{(L)} - \mathbf{y}).
\end{aligned}$$

The costs will be averaged over multiple trials and the weights and biases will be updated, whereafter the procedure repeats itself. For the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$, the derivative is: $\sigma'(x) = \sigma(x)(1 - \sigma(x))$. Similar formula's apply for all weights and biases, $\mathbf{w}^{(i)}$ and $\mathbf{b}^{(i)}$. This procedure is called *backpropagation*.

The choice for the number of hidden layers and the number of neurons in the hidden layers may be arbitrary and is often based on some trial and error. Furthermore, the batch size, the number of samples to work through before updating the internal model parameters, needs to be estimated. Then the number of epochs, which is the number of times that the algorithm will work through the entire training dataset, needs to be specified as well. Finally, the procedure needs to be done a few times and averaged because the initial random weights will influence the outcome.

An obvious disadvantage of an ANN is that this model is hard to interpret and that it is close to a so-called 'black box' algorithm. The output is based on the initial weights, therefore different simulations will result in different models. Again, the hyperparameters need to be optimised which is done by trial and error. Furthermore, the training time of an ANN is time-consuming and increases by the number of neurons and hidden layers, though its production time is quite fast once trained (Doerffer and Schiller, 2007). Another advantage is that neural networks are able to detect non-linear and complex relationships between variables. It is not straightforward to estimate the uncertainty of a prediction from a neural network, however,

some techniques have been proposed to compute the uncertainty such as the dropout algorithm (Gal and Ghahramani, 2016). Lastly, to train a neural network, a relatively large dataset of training data is desired which may be hard to obtain.

The algorithm designed by Doerffer and Schiller (2007) uses two neural networks, a so-called inverse NN and a forward NN. The inverse NN has eight reflectances combined with some geometry information as input and three IOPs as output. The forward NN uses the IOPs as input with the same geometry information and eight reflectances as output. The three IOPs are (1) scattering of all particles (2) absorption of phytoplankton pigments (a_{pig}) and (3) absorption of gelbstoff and the bleached fraction of suspended matter (Doerffer and Schiller, 2007). Using the following formula:

$$\text{CHL-a} = a \cdot a_{pig}^{b},$$

the chlorophyll-a concentration can be computed, where a and b are two variables that depend on the location/type of water. The development of the neural networks is based on 550,000 simulated observations. The algorithm available through ESA's Sentinel toolbox SNAP is an improved version of the algorithm designed by Doerffer and Schiller (2007) (Brockmann et al., 2016; SNAP, 2020). It is enhanced by additional neural networks and neural networks that are specifically designed to cover extreme ranges of input parameters. The original algorithm used four hidden layers with 8, 12, 16 and 45 neurons, respectively. Unfortunately, for the modified algorithm no such details can be found, except that approximately 5 million generated observations are used to train the model. Another modification is that the reflectances in the original algorithm are the water-leaving radiance reflectances (after atmospheric correction), while the algorithm in SNAP requires data before atmospheric correction (e.g. top of atmosphere data from Sentinel 2, level 1C). To determine the uncertainty of the IOPs, again a neural network is used. This NN is trained with the errors of the NN that computes the IOPs using reflectances as input. Consequently, the error is the absolute difference between the estimated IOP and the simulated IOP. To train the uncertainty NN, the estimated IOP is used as input and error as output. Using these uncertainties for the IOPs an uncertainty for the estimated chlorophyll-a concentration can be computed. Finally, when using the C2RCC algorithm in SNAP, multiple parameters such as salinity and temperature of the water are used as input variables. The exact use of these variables as for the reflectances in the neural networks is unclear.

To conclude, the C2RCC algorithm can be used to estimate chlorophyll-a concentrations using reflectances, although, the use of neural networks makes it hard to interpret. To train these types of models a large number of observations is required. This is often a problem in EO as in-situ data is needed, which leads to the use of simulated data. The uncertainty quantification is done using a neural network as well, though this can be done more elegantly by using a Gaussian process regression.

2.3.3. Gaussian Process Regression

"A Gaussian process is a generalization of the Gaussian probability distribution" (Rasmussen and Williams, 2006). While a (univariate) probability distribution tells us something about a random variable (which is a scalar), stochastic processes tell us something about functions. Consider a regression problem with a 1-dimensional input and output. In Figure 2.3 (left) a simple example is shown, where 10 samples are drawn from a prior distribution. This prior is a distribution over functions described by a Gaussian process. In this example, there is no knowledge about the true process, so the prior mean function is equal to zero which means that at each input value, the average value over the sample functions is zero. When two data points are observed, it is possible to force the Gaussian process to draw functions that pass

these observations (no noise assumed). This can be seen in Figure 2.3 (right). The prior variance was assumed to be independent of the input value, however, the posterior variance clearly is not. One can see that the certainty is increased near the observations and decreased the further you are away from them.

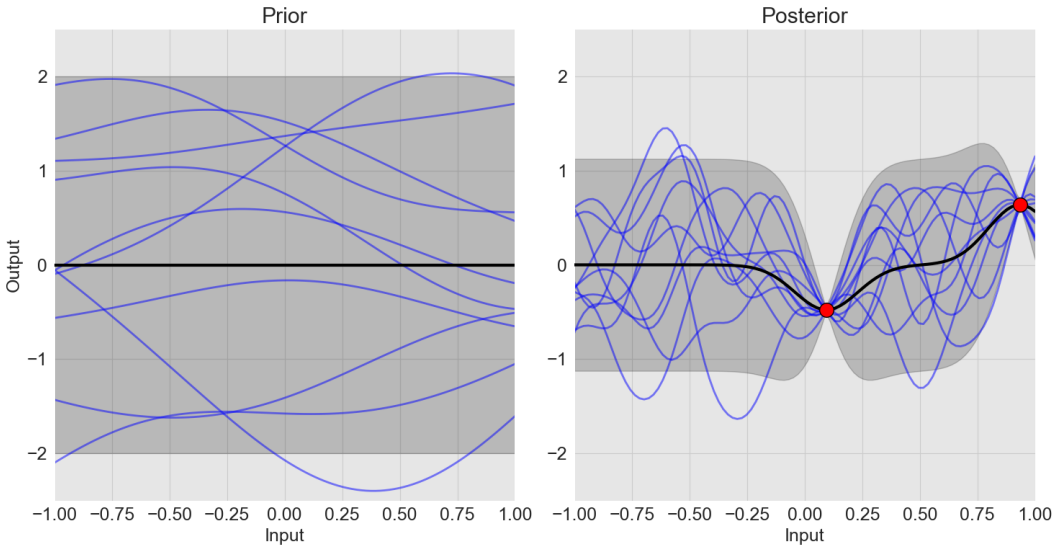


Figure 2.3: A simple example of 10 samples (blue lines) drawn from a prior distribution (left) and a posterior distribution (right). The prior mean and posterior mean are shown as a black line and the shaded region represents twice the standard deviation at each input value. The observed data points are shown as red dots.

A tacit assumption is made about the shape of the functions. This is specified by a chosen covariance function, through which smoothness and stationarity can be adjusted. A more in-depth and mathematical explanation about Gaussian process regression is given in chapter 4.

Pasolli et al. (2010) compared the Gaussian process regression model with the state-of-the-art regression methods discussed before. They did this using measurements of chlorophyll-a concentrations in both open and coastal waters and reflectances from the MERIS and SeaBAM datasets. The GPR method performed best in terms of mean-squared-error for both experiments and they conclude that “it [GPR] is a valid alternative to state-of-the-art regression methods” (Pasolli et al., 2010). Reasons for this statements are that the solution will be analytical using a GPR and that the estimation accuracies are in general better in complex chlorophyll-a concentration problems. Furthermore, the machine learning techniques described before, require a large number of observations whereas a GPR is able to perform well with limited sample availability. Finally, they state that a disadvantage of GPR is that a matrix inversion is needed which can be a problem when the number of observations increases. Verrelst et al. (2012) add that a GPR provides a confidence interval for prediction and that the hyperparameters can be interpreted better in comparison with neural networks.

Verrelst et al. (2012) and Pasolli et al. (2010) studied the estimation of the chlorophyll concentration on land and in water, respectively, using reflectances with a GPR model. However, both did not study the choice of covariance function (both used a squared exponential kernel) as well as the influence of the hyperparameters. Additionally, the matrix inversion problem is not solved and only the theoretical suggestion of applying a Cholesky decomposition is done. For these reasons, the GPR model will be analyzed extensively in this research. The performance for different kernels is tested and the influence of hyperparameters is analyzed. Finally, approximation techniques for the matrix inversion are evaluated. To do so, first in-situ as well as satellite data need to be collected carefully.

3

Data Collection

Before starting with spatio-temporal statistics and the analysis, data from the commercial satellites and in-situ data has to be collected. Commercial satellites often do not share their data for free as this is part of their business model. Deltares and therefore the HiSea project makes it possible to obtain high-resolution data, though there are multiple possibilities. The way of observing the data is different for every satellite/mission as their sensors are diverse and the goal of the commercial business may be different.

The in-situ data has to satisfy a few requirements as well. Multiple observations spatially as well as temporally are desired for doing spatio-temporal statistics. Furthermore, it has to contain the chlorophyll-a concentration including the coordinates and time of observation. Additionally, the measurements need to be recent, because the high-resolution satellite data is only available from the past few years. Therefore, various platforms have been obtained and evaluated whether it is useful for our purpose.

Finally, the in-situ data and (very-)high-resolution data need to be available in the same domain, spatially as well as temporally. This means that an in-situ dataset of cloudy days is unusable, while having only one measurement per day on one location is unusable as well. Therefore, the in-situ data as well as the high-resolution reflectance data need to be selected carefully.

3.1. In-Situ Data

First, a variety of in-situ data will be listed, whereafter the datasets will be evaluated and checked for usability.

3.1.1. MONGOOS

The Mediterranean Oceanography Network for the Global Ocean Observing System (MONGOOS) is a platform that is part of the GOOS (Global Ocean Observing System) which is administrated by the IOC (Intergovernmental Oceanographic Commission) which in turn is part of UNESCO (Sofianos et al., 2018). Concerning the Mediterranean Sea, the summarized objective of MONGOOS is to maintain and obtain oceanographical products and services. Together with improving the scientific understanding and creating awareness of the products and services.

To view the available datasets, the datacenter service via their website is used. However, only a few datasets containing chlorophyll-a concentrations are available (two fixed platforms in the Aegean sea). Moreover, there was no clear way of downloading the available datasets so this service does not seem to be useful for this project.

3.1.2. CMEMS

The Copernicus Marine Environment Monitoring Service (CMEMS) is part of the EU funded Copernicus programme which is in partnership with ESA (Le Traon et al., 2019). The purpose of the CMEMS is to provide data and services for the benefit of the safety of marine life, the environment of coastal and marine waters and for climate and weather forecasting.

Using their data archive of ocean products made it convenient to search for data. Both in-situ data and satellite data can be obtained here, however, it is not possible to filter between the two which makes it a little unstructured. Only one in-situ product with data from the past 7 years containing chlorophyll-a concentration is available. Unfortunately, when downloading this dataset, four files are available and none of them contains information about the chlorophyll-a concentration.

3.1.3. PANGAEA

"The information system PANGAEA is operated as an Open Access library aimed at archiving, publishing and distributing georeferenced data from earth system research" (Grobe et al., 2006). This system is hosted by the Alfred Wegener Institute, Helmholtz Center for Polar and Marine Research (AWI) and the Center for Marine Environmental Sciences, University of Bremen (MARUM). The data library, which is supported by the European Commission and the German Federal Ministry of Education and Research (BMBF), is deliberately an open library to promote research which makes it very accessible to retrieve data.

A dataset from global in-situ measurements of chlorophyll-a concentration can be obtained (Soppa et al., 2017) and is visualized in Figure 3.1. The data ranges from 1988 until 2012, which makes it an unpractical source. Zooming in on the Mediterranean Sea, data is only available up to 2008 and for the Aegean Sea, the data ranges from 1995 until 1999. Furthermore, the number of observations is limited, there are only 84 observations in the Aegean Sea in five years. Again, this makes it an impractical data source to use.

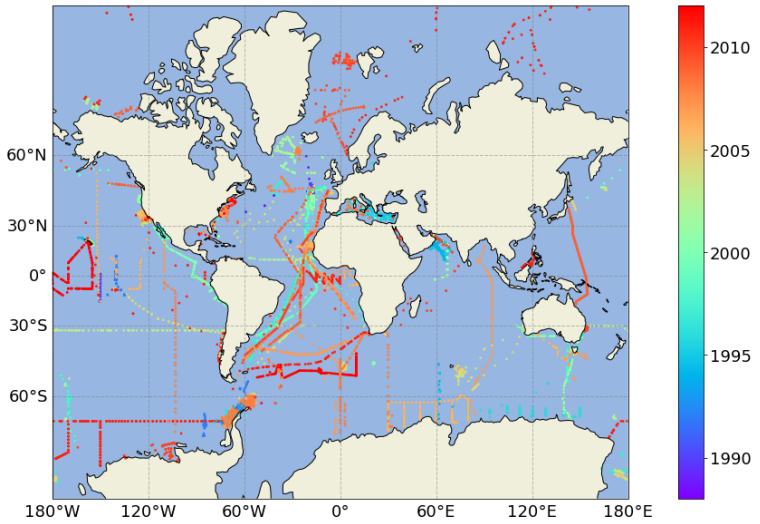


Figure 3.1: Visualization of the measurements obtained from PANGAEA (Soppa et al., 2017).

3.1.4. EMODnet

"The European Marine Observation and Data Network (EMODnet) is a network of organisations set up in 2007 by the European Commission in the framework of EU's Integrated Maritime policy to address the fragmented marine data collection, storage and access in Europe" (Calewaert et al., 2016).

Data ranging from 2014 to 2017 can be obtained containing information about the chlorophyll-a concentration. 172 stations in the Mediterranean Sea are available, see Figure 3.2. For temporal regression, this data can be useful as multiple stations measure the concentration every two weeks. For spatial statistics, however, this data is less suitable as there are not many stations in a small region collecting data on the same day.

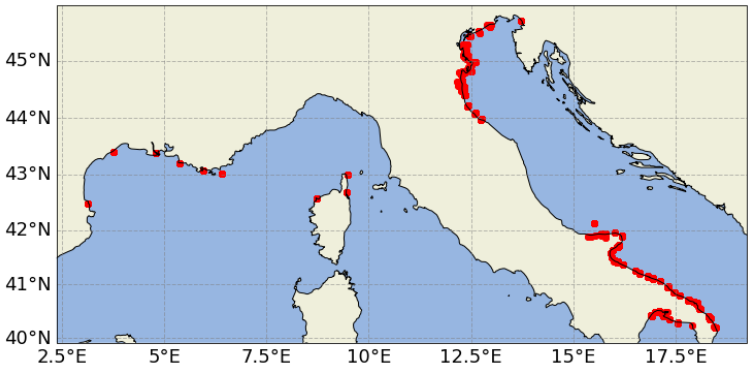


Figure 3.2: Available stations that collect information about the chlorophyll-a concentration in the Mediterranean Sea. Retrieved from EMODnet.

3.1.5. SeaDataNet

SeaDataNet is an Integrated research Infrastructure Initiative (I3), a European Commission sponsored infrastructure that provides access to marine-related data. It is doing so by using the 40 interconnected Trans National Data Access Platforms (TAP) (Schaap and Lowry, 2010). The datasets are provided by 45 different National Data Centers of 35 countries that share a coast along the European waters.

The service contains many datasets and it is easy to search for the desired region, parameter and time, see Figure 3.3. SeaDataNet provides data in NetCDF format and although it was indicated that datasets contain the chlorophyll-a concentration, no chlorophyll-a information is encountered when opening these files.

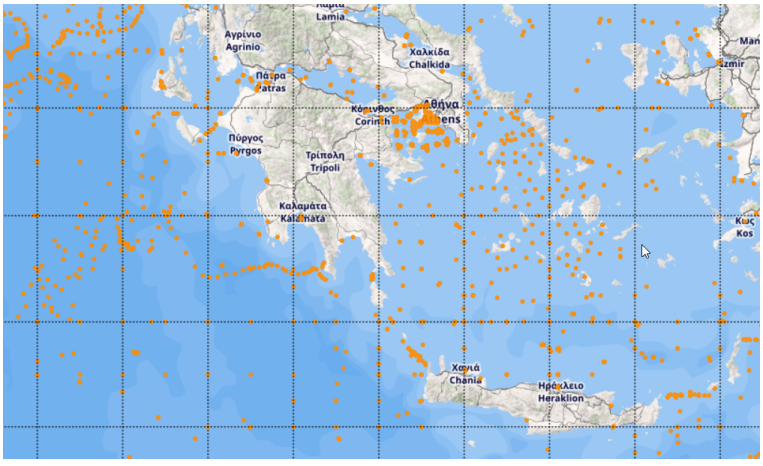


Figure 3.3: The available datasets that are indicated containing data about the chlorophyll-a concentration using the SeaDataNet service. However, for none of these datasets, information about chlorophyll-a has not been encountered.

3.1.6. ODYSSEA

ODYSSEA is a Horizon 2020 research and innovation programme of the European Commission. Horizon 2020 is one of the Framework Programmes (FP8) funded by the European Union (Spanoudaki et al., 2020). The project aims to make marine-related data available for a wide range of consumers.

The service it provides is very user-friendly and it is clear how to access the datasets. However, the dataset stated here is obtained via Deltares and is not available on their platform yet. The datasets contains chlorophyll-a concentration and is situated in the Aegean Sea within the time range: 29-7-2019 to 21-8-2019, see Figure 3.4. The measurements are taken by a glider that dives underwater and measures the depth, chlorophyll-a concentration, time and location twice every minute. Therefore, this dataset makes it possible to do spatio-temporal statistics as it contains multiple observations on the same day of different locations and doing so for multiple days.

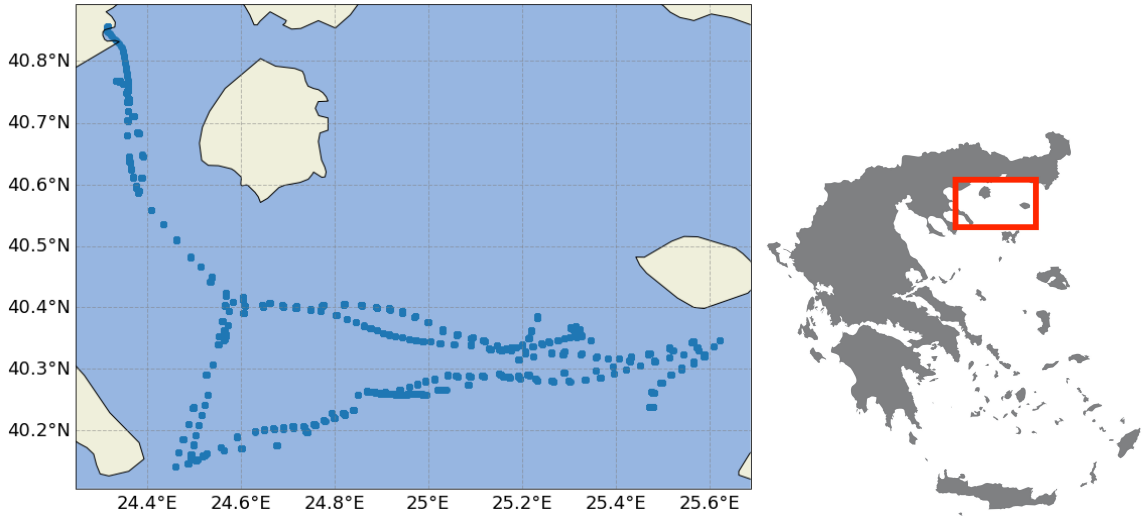


Figure 3.4: Measurements of the glider containing information about the depth and chlorophyll-a concentration. This visualization covers 24 days of measurements, containing approximately 2800 observations per day. The dataset is from the ODYSSEA programme and is located in the Thracian Sea which is the northern part of the Aegean Sea.

After some extensive search for in-situ data, the ODYSSEA data seems ideal for this study. Containing both in space as well as in time multiple observations makes it possible to do spatio-temporal statistics and thus will be used throughout this research.

3.2. High-Resolution Reflectance Data

For the high-resolution data which contains the reflectances, multiple options are available, see Table 3.1. These commercial missions/sensors have a very high spatial resolution of approximately 2 meters and provide the obtained data in several bandwidths. These bandwidths are the explanatory variables in the regression analysis and usually, a higher number of explanatory variables is desired to find a relationship with the response variable. Therefore, Worldview-2 from DigitalGlobe (USA) will be the first commercial satellite to evaluate. It is not necessary to use the data from all these satellites, as the idea of the problem remains similar. The only constraint is that the data needs to match with the in-situ data in terms of time and location, the satellite data needs to be from the same day and of the same location as the in-situ data.

In Table 3.2 the available spectral ranges of each mission can be seen. The bandwidth

Mission/Sensor	Agency	Year of Launch	# Bands	Spatial Resolution
Pleiades	CNES, France	2012	4	2.8 m
Superview-1	Beijing Space View Tech Co Ltd.	2016	4	2.0 m
WorldView-2	DigitalGlobe, USA	2009	8	1.8 m
TripleSat	21AT, China	2015	4	3.2 m
Spot-6	CNES, France	2012	4	6.0 m
KOMPSAT-3	KARI, S-Korea	2012	4	2.8 m

Table 3.1: Information about the 6 commercial missions/sensors for retrieving the high-resolution reflectance data. The spatial resolution is for the multispectral bands at nadir.

is the difference between the two wavelengths and the average of the two wavelengths is the central wavelength which is denoted by λ in the equations. For example, the Pleiades satellites use a spectral range of 430-550 nm for the blue band. The bandwidth is 80 nm and the central wavelength is 490 nm. The spectral ranges can be overlapping with each other and sometimes leave gaps. Using more bands usually indicate that the spectral ranges are smaller (i.e. the bandwidth is smaller). A sensor is then able to measure the radiance more specifically which can lead to a lower spatial resolution. All bands are measured with the same spatial resolution for each sensor, see Table 3.1 for this spatial resolution.

Mission/Sensor	Coastal	Blue	Green	Yellow	Red	Red Edge	NIR1	NIR2
Pleiades		430-550	490-610		600-720		750-950	
SuperView-1		450-520	520-590		630-690		770-890	
WorldView-2	400-450	450-580	510-580	585-625	630-690	705-745	770-895	860-1040
TripleSat		440-510	510-590		600-670		760-910	
SPOT-6		450-525	530-590		625-695		760-890	
KOMPSAT-3		450-520	520-600		630-690		760-900	

Table 3.2: The spectral ranges for the 6 commercial missions/sensors. All in nanometers.

3.2.1. WorldView-2

WorldView-2 is a high-resolution commercial imaging satellite and was launched on October 8, 2009 (DigitalGlobe, 2010). The satellite is in a sun-synchronous orbit at an altitude of 770km and thus has an orbit period of around 100 minutes. The time before it returns at the same location can take up to 1.1 days (the temporal resolution). The instrument onboard is called the WorlView-110 camera (WV110) and can obtain the images using a Panchromatic mode and a Multispectral mode, where 8 bands can be used (the spectral resolution). The spatial resolution is different per area, e.g. for European cities, the resolution is 40cm (Pan and Pan-sharpened) and 1.8m (MS).

Two data products can be obtained using the ESA website, a full archive and European cities. Access to the full archive is only possible by submission of a project proposal. For the European cities product, only selected regions can be chosen to obtain data from that area, see Figure 3.5. As one can see, mostly land data is available and for this study data on the sea is desired. For cities in the coastal area, some images do contain the sea as well, however, the images are very close to the coast and the ODYSSEA in-situ data does not overlap with these images. Also, the temporal resolution is very low as only four images are available for the Greek city Kavala all from April 2011. Therefore, this data product does not seem practical for this study.

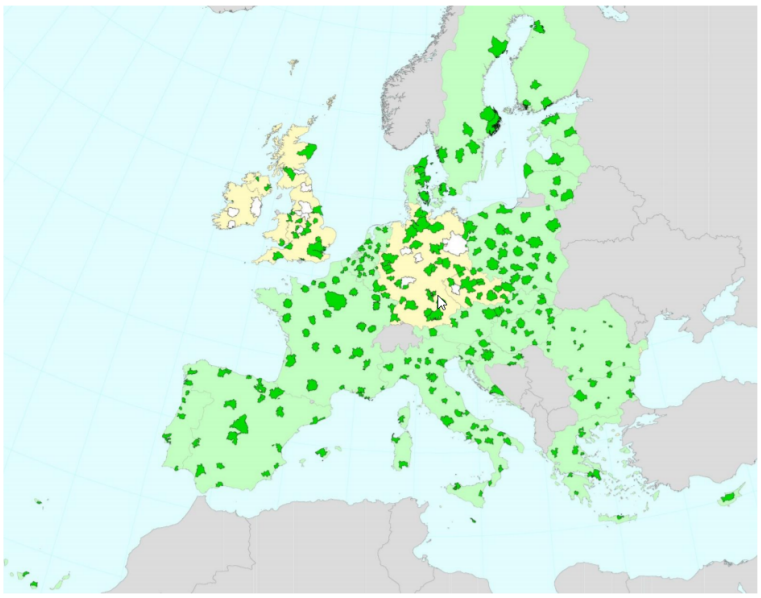


Figure 3.5: Coverage of large urban zones (LUZ) of the European cities product of the WorldView-2 mission.

3.2.2. Pleiades

The Pleiades constellation consists of two very-high-resolution satellites: Pleiades-HR 1A and B, launched on December 17, 2011 and December 2, 2011 respectively. The satellites are in a sun-synchronous orbit at an altitude of 695km at opposite sides of the earth which makes it possible to access any place of the Earth each day and rapid imaging for applications as civil security and defense missions. The sensor can obtain images in 4 bands with a spatial resolution of 2.8 meters (Lebègue et al., 2012).

By using the ESA data service, data collected by the Pleiades mission can be found. By setting the timeframe equal to the timeframe of the in-situ data by ODYSSEA there are only 10 datasets available scattered over the Earth. By selecting the datasets individually it becomes clear that the areas of interest are very specific. For example, three of the ten available datasets are images of glaciers in Iceland. Therefore, this data product can only be used when in-situ data is found in these areas or when a specific order can be done to create images of the area of interest.

3.2.3. SuperView-1

The SuperView-1 or GaoJing-1 constellation composed of four EO satellites is a mission by the Beijing Space View Tech Co Ltd. Again, its objective is defense and security as well as gathering information concerning land and forestry management. The program started in 2016 with the launch of the first two satellites, followed by the other two satellites in 2018. The satellites are in a sun-synchronous orbit at an altitude of 530km. The sensors collect images in four bands with a 2-meter spatial resolution and the temporal resolution is about 2 days.

Unfortunately, the data is not freely available and only a few samples can be obtained. Just as the high-resolution data from WorldView-2 and Pleiades, the data from SuperView-1 can be bought using the Apollo Mapping service. Using the Image Hunter search engine by Apollo Mapping, another problem arises, there is not one satellite that creates an image every day of the same area. A solution to this problem may be to use images from several satellites, however, this creates the problem that sensors obtain their data in different wavelengths. Ideally, in-situ data from a considerable amount of days around a year with matching satellite images can be gathered to create a model that is as complete as possible.

3.2.4. TripleSat

TripleSat consists, as the name suggests, of three satellites operating together at an altitude of 651km in a sun-synchronous orbit. Launched in 2015, the operational time will be approximately 7 years and the costs are fully covered by 21AT (Twenty-First Century Aerospace Technology) based in China. The sensors onboard the three satellites obtain the images in 4 bands with a spatial resolution of 3.2 meters.

Again, only a few data samples are freely available. Via Apollo Mapping and their search engine Image Hunter, it is possible to purchase the desired data. Similar to SuperView-1, this data is not used in this research and other satellite data sources are considered.

3.2.5. SPOT-6

SPOT (Satellite Pour l'Observation de la Terre) is a space program of the French space agency CNENS. It contains seven missions, starting in 1986 with the SPOT-1 that was able to obtain images with 20-meter spatial resolution. Nowadays, only two satellites are in operation, SPOT-6 and 7, launched in 2012 and 2014 respectively. Both satellites are put in the same orbit as the Pleiades constellation and have similar sensors that obtain their images in 4 bands with a spatial resolution of 6 meters.

The data from both satellites can be obtained in a similar way as for the Pleiades constellation. Again, the Apollo Mapping service can be used and the ESA data service can be accessed. The ESA data service does not contain any matching data products with the in-situ data from ODYSSEA. Searching without the constraints of matching the in-situ data, showed that there is limited data available via this service.

3.2.6. KOMPSAT-3

KOMPSAT (Korean Multi-Purpose Satellite) is a Korean space program that has many purposes: military observations, communication and environmental monitoring, just to name a few. The satellite KOMPSAT-3 is in a sun-synchronous orbit at an altitude of 685km. The sensor onboard uses 4 spectral bands with a spatial resolution of 2.8 meters.

The data obtained by the KOMPSAT satellites can be obtained from the Apollo Mapping service via the Image Hunter search engine. Again, only a few samples can be obtained freely.

Since all of the high-resolution data did not match with the in-situ data from ODYSSEA, medium-resolution data (that can be obtained freely) will be considered to be used in this research. The advantage is that it is very likely to find a match between the in-situ data and the medium-resolution data in terms of time and place. The disadvantage is that it is medium-resolution (10 meters), instead of high-resolution (approximately 2 meters). The study was to estimate the chlorophyll-a concentration using high-resolution data, however, doing this with medium-resolution does not alter the essence of the study much. We need to consider the increase of satellite data, nonetheless. A reasonable option for medium-resolution satellite data is the Sentinel-2 satellite.

3.2.7. Sentinel-2

The Sentinel-2 constellation consists of two satellites (A and B) which are part of the Copernicus space program from ESA. The satellites are launched in a sun-synchronous orbit at an altitude of 786km in 2015 and 2017. The sensors onboard both satellites obtains images in 13 spectral bands with different spatial resolutions. The blue, green, red and NIR bands are available in 10-meter resolution, other bands such as SWIR and Vegetation red edge are available in either 20 or 60 meters resolution. Some of the main application of this mission is to monitor agriculture, land ecosystems and inland/coastal water quality; management of forests

and civil security (Drusch et al., 2012). The temporal resolution is 5 days (instead of 10 days because of two satellites being used).

With the Copernicus Open Access Hub service, the data from both Sentinel-2A and B can be obtained freely. By selecting the area of interest and the sensing period based on the in-situ data, 20 products satisfy the requirements. Four products per available day are accessible because two images cover the area of interest and there are two types of products available for each image. Firstly, the level-1C product consists of the Top-Of-Atmosphere (TOA) reflectances and secondly, the level-2A product consists of the Bottom-Of-Atmosphere reflectances. The connection between these products and the algorithm applied will be discussed in the next section. Finally, some of the data products will contain a significant amount of clouds such that the data is impractical. This can be solved by filtering on the cloud cover percentage. For example, when we take into account the datasets with a cloud coverage between 0 and 5%, six data products are available for three different days.

For these reasons, the Sentinel-2 satellite will be used to retrieve data from throughout this research.

3.3. Data Preprocessing

Satellite images have to be preprocessed as several factors will disturb the data. This can be done manually, but for some higher level data products, preprocessing is already been done. Also, the in-situ data needs to be preprocessed before analyzing as there may be observations that are impossible (negative concentrations) or outliers (e.g. coordinates out of region).

3.3.1. Atmospheric Correction

The main part of this is the algorithm that derives the bottom of atmosphere (BOA) reflectances (or surface reflectances) from the top of atmosphere (TOA) reflectances. Furthermore, the different product types and classification algorithms will be discussed here. The information in this section is retrieved from the Sentinel-2 technical guide (“Sentinel-2 MSI - Technical Guide”, n.d.).

The first product, which is the basis of the other products, is called ‘Level-0’. This is the compressed raw data obtained by one of the sensors and is then used to create the ‘Level-1’ products. This is subdivided into three categories: Level-1A, B and C. The Level-1A product contains the uncompressed raw data and a start has been made with registering the spectral bands and some ancillary data is used to process the data. This additional data consists of information about the orbit of the satellite. Further processing of the data includes radiometric processing and defining the geometry of the grid, after which the Level-1B product is obtained. Radiometric processing consists of multiple actions, such as equalizing corrections and dark signal corrections. For missing values, an interpolation technique is used to fill in the gaps. Then, the first data product available for consumers is the Level-1C product. This is the data containing orthorectified reflectances from the top of the atmosphere. Furthermore, cloud masks are included. The incoming solar radiance is accounted for by computing the direction of the radiance which is defined as the zenith angle. Additionally, the distance from the Earth to the sun is used, because the irradiance is proportional to the distance squared. This follows the inverse-square law of irradiance. The cloud mask calculations are done to identify dense and cirrus clouds and is done using the 60-meter resolution spectral bands. The simplified version of identifying dense clouds is to tag pixels as dense clouds when a high value in the blue band is observed. A more detailed description of the algorithm will be given later, where a distinction between snow and clouds is made as well. Cirrus clouds are harder to identify than dense clouds as they are thin and almost transparent. Again, the simplified algorithm to detect cirrus clouds is by tagging pixels as cirrus clouds when a low value in the blue band

and a high value in band 10 is obtained. This band is in the SWIR region and is sometimes referred to as the cirrus band.

The final product, the Level-2A product, contains the BOA reflectances including all sorts of classification (cloud, cloud shadows, water, snow etc.). Along with the TOA reflectances in the Level-1C product, this Level-2A product is also available for consumers. The process of getting TOA reflectances from BOA reflectances is called atmospheric correction (AC). Sensor specific information is used to calibrate formulas that explain atmospheric conditions as well as solar geometries and ground elevations. One can imagine that objects at different heights reflect differently and thus a map of ground elevations is needed for this process. The aerosol optical thickness can be calculated using a specific reference area where the behaviour of reflectances is known. Other atmospheric interferences such as water vapour, haze and cirrus are dealt with by detecting them and sometimes removing them. The classification is done using neural networks and thresholds based on single radiances, ratios and slightly more complex ratios such as the NDVI.

The classification of cloud and snow is done by following multiple steps, where at each step more cloud-free pixels are tagged and the remaining pixels are classified as cloud or snow. First, the red band and the normalized difference snow index (NDSI) are used to identify cloud-free, potential cloud and cloudy pixels. For example, if the reflectance in the red band is lower than 0.07, the pixel is considered to be cloud-free. If the value exceeds 0.25 the pixel is considered cloudy. When $0.07 \leq R_{rs}(\lambda_{Red}) \leq 0.25$ the pixel is potentially cloudy and a cloud probability is assigned to these pixels (scaling linearly from 0 to 1).

$$NDSI = \frac{R_{rs}(\lambda_{Green}) - R_{rs}(\lambda_{SWIR} = 1610)}{R_{rs}(\lambda_{Green}) + R_{rs}(\lambda_{SWIR} = 1610)}.$$

After this first step, more specific thresholds for detecting snow are applied to the pixels with a probability larger than zero. The pixels having a snow probability larger than a certain threshold are then combined to create a snow mask. Pixels with a snow probability below this threshold are passed to the next step where vegetation, soils/desert, water and clouds are classified. Pixels are classified as water when the ratio of band 2 and band 11 ($R_{rs}(\lambda_{Blue})/R_{rs}(\lambda_{SWIR} = 1610)$) is higher than a certain threshold and band 12 ($R_{rs}(\lambda_{SWIR} = 2190)$) is lower than a certain threshold. This only applies to pixels that are not classified as snow, vegetation or soil previously in the process.

Pixels that have not yet been classified are identified as thin cirrus cloud, cloud medium probability and cloud high probability based on the final probabilities. Finally, the shadow created by the clouds is classified by using the position of the sun, cloud mask and the cloud height distribution. By applying a self-organizing map (also known as a Kohonen map) the 'dark areas' are classified.

In our area of interest (the Thracian sea) and our study, we are interested in water leaving reflectances. This means that the Level-2A product will be used in this study and as we are only interested in images of water, pixels containing land and clouds need to be detected.

3.3.2. Land Detection

The classification masks available in the Level-2A product from the Sentinel-2 satellites makes it possible to easily create a mask array for pixels containing water. However, when only four bandwidths are available it is still possible to detect land from water as shown in the following example.

The satellite images that are used in this study will focus on the Thracian sea where numerous islands are located. The models and interpolation techniques that will be used need the

water leaving reflectances to estimate the chlorophyll-a concentration. Therefore, the reflected light that comes from land needs to be removed. The island Limnos (Λήμνος), for example, is located in the Thracian sea and the reflected data in four different bands (blue, green, red and NIR) can be seen in Figure 3.6. Because the near-infrared radiation is caused by heat, the land is easily detected by the higher near-infrared radiation. The land will have a higher temperature than the water, so the near-infrared wavelength will be used to detect the land and remove these observations from the dataset when predicting.

To be precise, the data used for these examples are obtained from the Sentinel-2B satellite on 9 August 2019 at 9:05:59 a.m. and has the Level-2A product level.

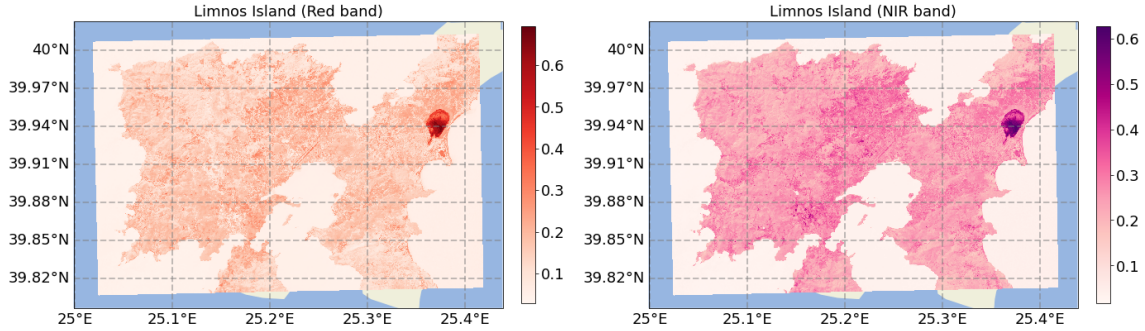


Figure 3.6: Images of Limnos Island in two different bands: red and near-infrared. It is visually able to see that the contrast is highest using the near-infrared image between the land and water. Near-infrared radiation is emitted by objects that have a high temperature, therefore the land will radiate more near-infrared than the water around it. The high ratios in the east of the island are caused by Aliki Lake, a salt lagoon that has a white color and thus high ratios of reflectances.

From the histogram (Figure 3.7) containing the near-infrared reflectances of Figure 3.6, one can see a peak for a low percentage of NIR reflectance and then a second peak around 22% of reflectance. The scale of the y-axis is logarithmic so one can split the two groups by selecting a value in the valley between those two peaks. Naturally, the border between land and water is not perfect in the sense that the pixels from the image contain either water or land. Therefore, some pixels will contain both land and water and as a result, the reflectance ratio will be in between those peaks. To verify this, a range of values in the valley is selected, see the orange colored observations in Figure 3.7. In the figure on the right, one can see the land detection, wherein orange the coastal observations are highlighted. These coastal observations are all on the border between land and water. Note that some of the observations seem to be on land, however, some small lakes causes these observations. Therefore, the reflectance ratio of 0.1 is set to be the value to split water and land observations. This is in line with the threshold used in the SNAP tool which is also 0.1 by default.

By comparing this result with the classification file from the Level-2A product, some differences are observed. In Figure 3.8, the classification performed by the AC from Sentinel can be observed. The pixels classified as water seem to be similar as in Figure 3.7 (right), with minor differences at the coast and some inland waters. Besides having vegetation and non-vegetation on the island Limnos, some pixels are classified differently. Along the coast and in two interesting regions some pixels are classified as clouds, snow or even unclassified. The region in the middle of the island is the airport of Limnos and two salt lakes are located on the east coast.

In Figure 3.9 a zoomed plot of the salt lake can be seen. The distinction between land and water is quite clear, however, the salt lake itself is wrongly classified. Note that this image does not contain any clouds (visually observed from the constructed RGB image), whilst at the boundary of the lake medium and high cloud probability is classified. The salt lake itself is

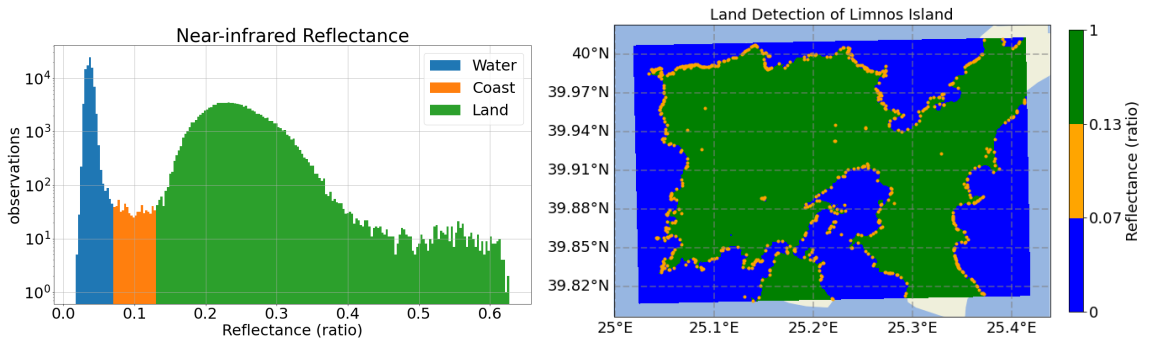


Figure 3.7: Histogram of NIR reflectances (left) and the visualisation of land detection of Limnos Island (right). Note the logarithmic scale in the histogram. The coastal area, as confirmation, is set to have a ratio between 0.07 and 0.13. In the visualisation, one can see the corresponding observations and the three classes (water (blue), coast (orange), land (green)). All orange observations are on the border of water and land as predicted.

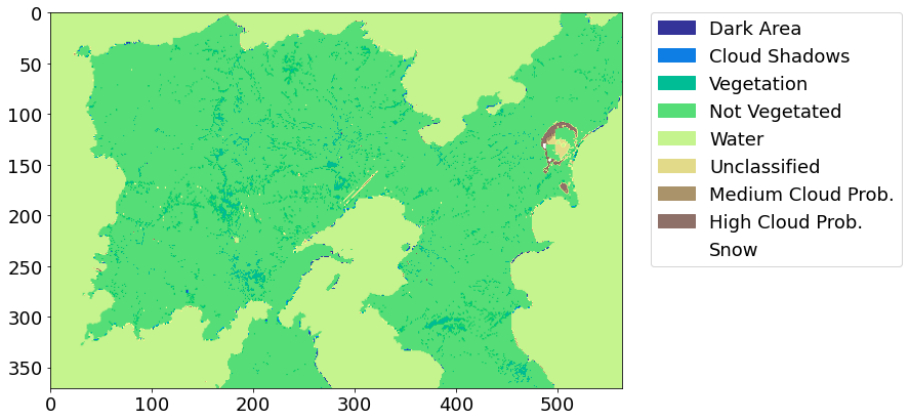


Figure 3.8: Classification map of Limnos island which can be retrieved from the Level-2A product from both Sentinel satellites. One can see a clear distinction between land and water, though at the boundary, airport (center of the island) and salt lakes (east coast) some misclassification is observed.

classified as not vegetated (correct), water (incorrect) and unclassified. The simple method to classify pixels land for NIR reflectances bigger than 0.1 does not have any problems with the salt lakes and airport.

Both methods can be used to obtain a mask array for the non-water pixels, though the classification method from Sentinel is likely to make some small errors due to special inland properties.

3.3.3. Cloud Detection

When the pixels containing land are omitted (masked), there is mainly one obstacle that prevents us from observing true water leaving reflectances, clouds. The detection of clouds for satellite imagery is a complete study on its own, as there are numerous kinds of clouds and the detection is also area related (Mahajan and Fataniya, 2019, Amato et al., 2008, Cutillo et al., 2004). Therefore, only the classification algorithm from Sentinel will be used to detect these clouds to prevent outliers in the estimations of the models.

In Figure 3.10, the classification map of the Thracian sea can be seen. Thin cirrus, medium and high cloud probability are brown colored and present mostly in the top-left part of the image. Visually, from the RGB image and the plots of the individual bands, only a part of the identified clouds can be seen. This makes it hard to verify that this algorithm is correct, though the shapes of the observed clouds seem to be in order. As there is no reason to reject this

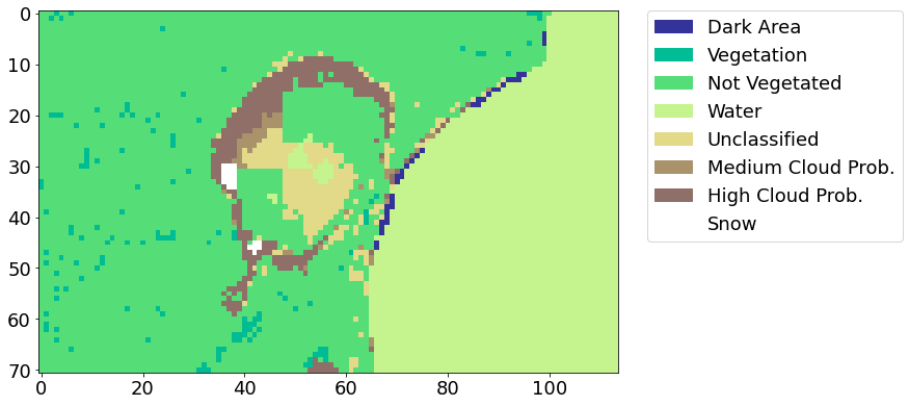


Figure 3.9: A zoomed in classification map focused on the salt lake of Limnos. Some pixels are wrongly identified as clouds or water and some are unclassified.

method, other than the misclassifications near the salt lake, this method will be used to mask the non-water pixels.

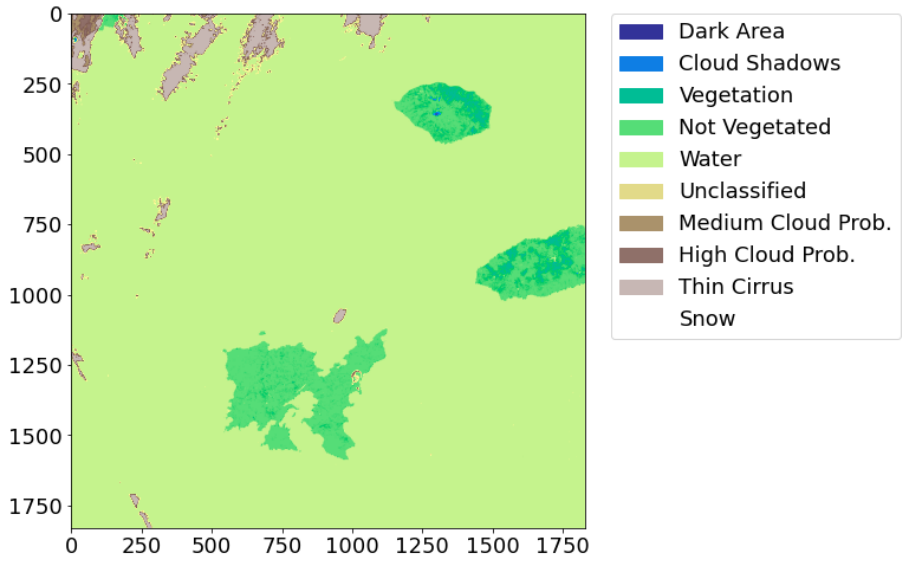


Figure 3.10: Classification map of the Thracian sea. The presence of dense clouds and especially cirrus clouds can be seen in the north-western part of this map.

4

Spatio-Temporal Statistics

Nowadays, with remote sensing using satellites, measurements are taken from anywhere on Earth. As a consequence of the rotation of the satellites around the Earth, measurements are taken with a regular time-step. This approach to acquire information from Earth is used in several disciplines, such as ecology, meteorology and geology. However, there are numerous non-scientific applications for remote sensing, for instance, military and human rights (Parks, 2009). To link remote sensing data with variables on Earth (e.g. type of tree, oxygen etc.) in-situ data is used. Where satellite data contains a large number of spatial observations every few days, in-situ observations are often taken at a single location but more frequently in time. Taking measurements in this manner, there will be locations and points in time where there are no measurements. To obtain these measurements in the gaps of time and space, spatio-temporal statistics is often used (Wikle et al., 2019).

A spatio-temporal model is a model that incorporates the relationship in space and time simultaneously. Theoretically, one can make a spatio-temporal model to model a physical process. Statistics is usually applied in these fields, despite of deterministic physical rules that may apply here. For example, predicting the exact wind speed at every location requires a very complex, deterministic model. To create such a model, one needs to incorporate atmospheric circulation, jet streams, temperature, etcetera. Doing this on a large scale, it requires an immense amount of computational power, which, in combination with the complex physical behaviour, makes it infeasible to do so. Therefore, randomness and uncertainty are used to create a *statistical* spatio-temporal model where the wind speed and direction are predicted for an area. Another reason to design a statistical spatio-temporal model is that observations will be made with an error. Some of the challenges of these models are handling the variables both in time and space, time is one-dimensional whereas space is often two- and sometimes three-dimensional. Furthermore, do you treat a time difference of size k and a spatial difference of size k similarly? Spatial data may be assessed by zip-code, state or country. Time, on the other hand, is often assessed by hour, day, month or year.

Analysis of spatio-temporal data is done for several reasons, one might be interested in understanding the relationship between two variables. Visualizing the data to understand the process can be done as well, however, this may be inconvenient when the spatial data is three dimensional. Furthermore, as shown by Kim et al. (2019), spatio-temporal statistics can be used to forecast the trajectory of hurricanes. Performing spatio-temporal statistics is broadly done because of three reasons (Wikle et al., 2019):

- To estimate a certain variable in time and/or space.
- To understand the behaviour of a certain variable.

- To forecast a certain variable in time.

Mainly the first reason applies to this research, estimating the chlorophyll-a concentration in space. Also, the relationship between the reflectances and the chlorophyll-a concentration will be investigated. To go further into depth, the analysis of this can be done in two different ways: the *descriptive* and *dynamic* approach.

4.1. Dynamic Approach

The dynamic approach, in contrast to the descriptive approach, can use the scientific knowledge of the processes more easily. As the name suggests, the dynamic approach uses the fact that many spatio-temporal problems in nature are some dynamic movements. There are differential equations that describe these movements, such as the Navier-Stokes equations, so it is a waste if these are not used. Naturally, the assumption here is that one has some scientific knowledge of the problem.

The dynamic approach tries to model the spatial process that changes through time. So, we try to model the present state of the process conditional on the data of the past. In general, these models will be a hierarchical model, where the conditional process can be described with parameters and these parameters are based on scientific knowledge.

Using a mapping that glues the data to the unknown process ($\mathcal{H}_t$), the model can be written as (Wikle et al., 2019):

$$Z_t(s) = \mathcal{H}_t(Y_t(s), \theta_{d,t}, \varepsilon_t(s)),$$

where $Z_t(s)$ is the observed measurement at spatial location s and time t. Similarly for $Y_t(s)$ which is the unknown true process and the error $\varepsilon_t(s)$. Then, the parameters used for this model are denoted by $\theta_{d,t}$. Now, the assumption is that $Z_t(s)$ is independent in time when it is conditioned on the true process and the parameters (Wikle et al., 2019). Using this assumption, we can write the joint distribution of the observed measurements conditioned on the true process and the parameter as:

$$[\{Z_t(s)\}_{t=1}^T | \{Y_t(s)\}_{t=1}^T, \{\theta_{d,t}\}_{t=1}^T] = \Pi_{t=1}^T [Z_t(s) | Y_t(s), \theta_{d,t}].$$

The component distributions $[Z_t(s)|Y_t(s), \theta_{d,t}]$ can be considered to be Gaussian or non-Gaussian depending on the specific problem.

This can be continued by making the Markov assumption that only the recent past is relevant for defining the present state of the true process. For a more detailed description of this approach, the reader is referred to *Spatio-Temporal Statistics with R* (Wikle et al., 2019, Chapter 5). In this study, we will use the descriptive approach to design a statistical spatio-temporal model for the chlorophyll-a concentration.

4.2. Descriptive Approach

Briefly explained, the descriptive approach uses a mean function and a covariance function to describe the process. However, the exact functions of variables having nonlinear relationships can be very complex. So, estimates need to be made of these functions, which is done by using a kernel. The kernel often assumes that nearby observations tend to be alike and that observations further away can be more diverse.

To put this into mathematical equations, the observations $Z(s,t)$ at location s and time t are modelled as (Wikle et al., 2019) :

$$Z(s,t) = Y(s,t) + \varepsilon(s,t).$$

where $s \in D_s$ and $t \in D_t$. D_s and D_t are the spatial and temporal domains respectively and are usually of the form: $D_s \subseteq \mathbb{R}^2$ and $D_t \subseteq \mathbb{R}$. $Y(s,t)$ is the notation for the true process,

the process that is desired to find a model for. Finally, the stochastic process for the errors, $\{\varepsilon(s,t)\}$, is assumed to have mean zero, fixed variance and independently distributed (i.e. i.i.d.) and independent from the observation $Y(\cdot,\cdot)$. Then, the true process Y is assumed to be able to decompose into two processes: a mean process and a mean zero random process which incorporates the spatial and temporal statistical dependencies.

$$Y(s,t) = \mu(s,t) + \eta(s,t).$$

The objective is to make predictions $\hat{Y}(s,t)$ using a linear predictor such that the mean squared prediction error is as low as possible. One approach, to give this problem more structure, is to assume that the true process $Y(s,t)$ is a Gaussian process.

4.3. Gaussian Process Regression

Gaussian process regression (GPR) is a nonparametric Bayesian approach of performing regression. This means that no assumptions are made about the distribution of the predictor and no particular form of relationship between the explanatory variables and the dependent variable is taken. To understand what a GPR is, we first need to define what a Gaussian process is (Rasmussen and Williams, 2006).

Definition 4.3.1 *A **Gaussian process** is a collection of random variables such that any (finite) subset of these variables follows a joint Gaussian distribution.*

A Gaussian process can be fully specified by a mean function and a covariance function. Take a real process $\{f(\mathbf{x}) : \mathbf{x} \in S \subset \mathbb{R}^d\}$ where $\mathbf{x}$ can be either a spatial, temporal or spatio-temporal location in S which is a subset of the d-dimensional space. Then, the Gaussian process can be written as:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \tag{4.1}$$

where the mean function is denoted by $m(\mathbf{x})$ and the (symmetric) covariance function $k(\mathbf{x}, \mathbf{x}') = cov(f(\mathbf{x}), f(\mathbf{x}'))$ for any location $\{\mathbf{x}, \mathbf{x}'\} \in S$.

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

Often, the mean function is set equal to zero (Rasmussen and Williams, 2006), though we will discuss the general case. The difference between common distributions such as the Gaussian and exponential distribution is that a sample from the Gaussian Process (Equation 4.1) is a function instead of a single value. Since the input of the mean and covariance function can be anything from the subset S, the true process f can be described on the entire subset S. In practice only a finite number of input variables can be used and by definition 4.3.1, it follows a multivariate normal distribution. Therefore, sampling from the Gaussian process at a finite set of points $X = \{x_1, \ldots, x_n\}$ can be done by drawing from $\mathcal{N}(m(X), k(X,X))$, where $m(X) = [m(x_1), \ldots, m(x_n)]^T$ and $k(X,X) \in \mathbb{R}^{n \times n}$ is the $n \times n$ covariance matrix. An element in the covariance matrix on row i and column j is equal to $k(X,X)_{i,j} = k(x_i, x_j) = cov(f(x_i), f(x_j))$ for $1 \leq i, j \leq n$.

For doing regression, we can treat the Gaussian process as a prior and condition on the observed data to find the posterior predictive distribution. First, let us introduce some notation for the data. Denote the training data by $\mathscr{D} = (X, \mathbf{y}) = \{\mathbf{x}_i, y_i\}_{i=1}^n$, where every input $\mathbf{x}_i \in \mathbb{R}^D$ is D dimensional and $y_i \in \mathbb{R}$ is one dimensional. As an example, one can think of $\mathbf{x}_i$ being a coordinate on Earth (2-dimensional) and y_i being the temperature (1-dimensional). The unlabeled data or test data can be written as $\mathscr{T} = X_* = \{\mathbf{x}_i\}_{i=n+1}^{n+k}$ where $\mathbf{x}_i \in \mathbb{R}^D$ and we wish to estimate $f(\mathbf{x}_i)$ for $i = n+1, \ldots, n+k$.

4.3.1. Noise-free Observations

When we assume the observations to be noise-free we have $f(\mathbf{x}_i) = y_i$, i.e. there is no error. For convenience, we denote $\mathbf{f} \in \mathbb{R}^n$ to be the vector of function values $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)$ which in this case is equal to $\mathbf{y}$. Furthermore, we denote $\mathbf{f}_* \in \mathbb{R}^k$ to be the vector of function values $f(\mathbf{x}_{n+1}), \ldots, f(\mathbf{x}_{n+k})$. Now, we can write the Gaussian process prior to be:

$$\mathbf{f} \sim \mathcal{N}(m(X), \Sigma_{ff}). \tag{4.2}$$

Here, $m(X) = [m(\mathbf{x_1}), \ldots, m(\mathbf{x_n})]^T$ and Σ_{ff} is the covariance matrix where every (i, j) element corresponds to $k(\mathbf{x}_i, \mathbf{x}_j) = cov(f(\mathbf{x}_i), f(\mathbf{x}_j))$ for $1 \leq i, j \leq n$. To predict $f(\mathbf{x}_i)$ for $\mathbf{x}_i \in \mathscr{T}$, we consider the following joint multivariate Gaussian distribution:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \text{Gau}\left(\begin{bmatrix} m(X) \\ m(X_*) \end{bmatrix}, \Sigma \right), \qquad \Sigma = \begin{bmatrix} \Sigma_{ff} & \Sigma_{ff_*} \\ \Sigma_{f_*f} & \Sigma_{f_*f_*} \end{bmatrix}. \tag{4.3}$$

Analogous to $m(X)$, we have $m(X_*) = [m(\mathbf{x}_{n+1}), \ldots, m(\mathbf{x}_{n+k})]^T$. The covariance matrix $\Sigma \in \mathbb{R}^{(k+n)\times(k+n)}$ is divided into four smaller matrices: $\Sigma_{ff} \in \mathbb{R}^{n\times n}$, $\Sigma_{ff_*} = \Sigma_{f_*f}^T \in \mathbb{R}^{n\times k}$ and $\Sigma_{f_*f_*} \in \mathbb{R}^{k\times k}$. Similarly to Σ_{ff}, every element of Σ_{ff_*} corresponds to $k(\mathbf{x}_i, \mathbf{x}_j) = cov(f(\mathbf{x}_i), f(\mathbf{x}_j))$ where $\mathbf{x}_i \in \mathscr{D}$ and $\mathbf{x}_j \in \mathscr{T}$ and likewise for the elements of $\Sigma_{f_*f_*}$ with $\mathbf{x}_i, \mathbf{x}_j \in \mathscr{T}$. It is now possible to derive the distribution of $\mathbf{f}_*$ conditional on $\mathbf{f}$ and as a result we get the posterior:

$$\mathbf{f}_*|\mathbf{f} \sim \mathcal{N}(m(X_*) + \Sigma_{f_*f}\Sigma_{ff}^{-1}(\mathbf{f} - m(X)), \Sigma_{f_*f_*} - \Sigma_{f_*f}\Sigma_{ff}^{-1}\Sigma_{ff_*}). \tag{4.4}$$

A noteworthy remark is that the computational complexity is dominated by computing the inverse of the matrix $\Sigma_{ff} \in \mathbb{R}^{n\times n}$. The number of flops to invert a matrix of size $n \times n$ using Gauss-Jordan elimination scales cubically ($\mathcal{O}(n^3)$). n is the number of in-situ observations and remember that, for the ODYSSEA dataset, roughly 67,000 observations are obtained. For the distribution of $\mathbf{f}_*|\mathbf{f}$ this inversion needs to be done once, as the result can be used for both the mean and covariance matrix. Techniques to speed up this inversion will be discussed later on.

4.3.2. Noisy Observations

In practice, observations will contain some noise and will not be perfect. This noise is often assumed to be modeled by the following equation:

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma_{noise}^2). \tag{4.5}$$

Similar notation as before can be used, however, instead of conditioning on $\mathbf{f}$ we need to condition on $\mathbf{y} = [y_1, \ldots, y_n]^T$. Note that it is still possible to find the conditional distribution of $\mathbf{f}_*|\mathbf{y}$ which is the true process of interest. The Gaussian process now induces the following prior on $\mathbf{y}$:

$$\mathbf{y} \sim \mathcal{N}(m(X), \Sigma_{ff} + \sigma_{noise}^2 I_n).$$

Then the joint multivariate Gaussian distribution of $\mathbf{y}$ and $\mathbf{f}_*$ is:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \text{Gau}\left(\begin{bmatrix} m(X) \\ m(X_*) \end{bmatrix}, \Sigma \right), \qquad \Sigma = \begin{bmatrix} \Sigma_{ff} + \sigma_{noise}^2 I_n & \Sigma_{ff_*} \\ \Sigma_{f_*f} & \Sigma_{f_*f_*} \end{bmatrix}. \tag{4.6}$$

Note that the only difference between Equation 4.3 and Equation 4.6 is the added noise in the covariance matrix and the replacement of $\mathbf{y}$ for $\mathbf{f}$. Note that the covariance matrices only depend on the input variables $\mathbf{x}_i$. Again, the conditional distribution can be derived:

$$\mathbf{f}_*|\mathbf{y} \sim \mathcal{N}(m(X_*) + \Sigma_{f_*f}(\Sigma_{ff} + \sigma_{noise}^2 I_n)^{-1}(\mathbf{y} - m(X)), \Sigma_{f_*f_*} - \Sigma_{f_*f}(\Sigma_{ff} + \sigma_{noise}^2)^{-1}\Sigma_{ff_*}). \tag{4.7}$$

The computational complexity is once more dominated by computing the inverse of a matrix with size $n \times n$ and is of order $\mathcal{O}(n^3)$.

4.3.3. Log Marginal Likelihood

Although the GPR is called a nonparametric regression method, there are some parameters in this model that we will call hyperparameters. Up to now, the standard deviation for the noise, σ_{noise}, is the only hyperparameter that we have encountered. The remainder of the hyperparameters are present in the covariance function. Later on, the marginal likelihood and also the log marginal likelihood will become of practice for optimizing these hyperparameters. The marginal likelihood, or sometimes referred as evidence, $p(\mathbf{y}|X)$, can be written in terms of prior and likelihood (Rasmussen and Williams, 2006). First, the following rule for marginalization is used. Given two random variables A and B the marginal distribution of A can be written as the joint distribution function of A and B by integrating over all possible outcomes of B:

$$p_A(a) = \int_b p_{A,B}(a,b)db. \tag{4.8}$$

Now, the marginal likelihood can be written as:

$$\begin{aligned} p(\mathbf{y}|X) &= \int_{\mathbf{f}} p(\mathbf{y},\mathbf{f}|X)d\mathbf{f}, \\ &= \int_{\mathbf{f}} p(\mathbf{y}|\mathbf{f},X)\cdot p(\mathbf{f}|X)d\mathbf{f}. \end{aligned}$$

The first equal sign uses Equation 4.8 and the second equal sign is because of the product rule in probability. The prior distribution is known and equal to $p(\mathbf{f}|X) = \mathcal{N}(m(X), \Sigma_{ff})$, see Equation 4.2. The likelihood can be written using Equation 4.5, because of independent noise we get again a Gaussian distribution:

$$p(\mathbf{y}|\mathbf{f},X) = p(\mathbf{y}|\mathbf{f}) = \mathcal{N}(\mathbf{f}, \sigma^2_{noise}I_n).$$

Then, the marginal likelihood can be written as (Rasmussen and Williams, 2006; Dunson et al., 2020; Murphy, 2012):

$$p(\mathbf{y}|X) = (2\pi)^{-n/2}|\Sigma_{ff} + \sigma^2_{noise}I_n|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{y}-m(X))^T(\Sigma_{ff}+\sigma^2_{noise}I_n)^{-1}(\mathbf{y}-m(X))\right). \tag{4.9}$$

For convenience, the log marginal likelihood is commonly used and can be written as:

$$\log p(\mathbf{y}|X) = -\frac{n}{2}\log(2\pi) - \frac{1}{2}\log\left|\Sigma_{ff}+\sigma^2_{noise}I_n\right| - \frac{1}{2}(\mathbf{y}-m(X))^T(\Sigma_{ff}+\sigma^2_{noise}I_n)^{-1}(\mathbf{y}-m(X)). \tag{4.10}$$

This result can also be derived from Equation 4.6 where it can be seen that $\mathbf{y} \sim \mathcal{N}(m(X), \Sigma_{ff} + \sigma^2_{noise}I_n)$. Now, the hyperparameters used in computing the covariance matrix and the noise can be estimated by maximizing the marginal likelihood (type 2 maximum likelihood). Since the logarithm is a monotone increasing function, this method is similar to maximizing the log marginal likelihood in Equation 4.10.

4.3.4. Hyperparameters

The variable for the noise in Equation 4.5 and hyperparameters that are used in computing Σ_{ff} are estimated by maximizing the log marginal likelihood. Usually, these parameters are denoted by $\boldsymbol{\theta}$ and an algorithm is used to maximise the log marginal likelihood, such as a gradient descent/ascent algorithm (Rasmussen and Williams, 2006). Using $\Sigma_n = \Sigma_{ff} + \sigma^2_{noise}I_n$ and Equation 4.10, the partial derivative of the log marginal likelihood with respect to $\boldsymbol{\theta}$ can be written as:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{y}|X, \boldsymbol{\theta}) = -\frac{1}{2}\frac{\partial \log |\Sigma_n|}{\partial \boldsymbol{\theta}} - \frac{1}{2}(\mathbf{y} - m(X))^T \frac{\partial \Sigma_n^{-1}}{\partial \boldsymbol{\theta}}(\mathbf{y} - m(X)) \tag{4.11}$$

Σ_n is said to be differentiable if every element is differentiable with respect to $\boldsymbol{\theta}$, therefore the choice for the covariance function (discussed in section 5.4) takes this into account. Since the determinant is a polynomial function of all the elements of the matrix Σ_n, $|\Sigma_n|$ is also differentiable. Furthermore, Σ_n is positive definite for all $\boldsymbol{\theta}$, so the inverse Σ_n^{-1} is also differentiable. The partial derivative of an inverse matrix can be computed using the following reasoning.

$$0 = \frac{\partial I}{\partial \boldsymbol{\theta}} = \frac{\partial \Sigma_n^{-1}\Sigma_n}{\partial \boldsymbol{\theta}} = \Sigma_n^{-1}\frac{\partial \Sigma_n}{\partial \boldsymbol{\theta}} + \frac{\partial \Sigma_n^{-1}}{\partial \boldsymbol{\theta}}\Sigma_n, \qquad \Rightarrow \tag{4.12}$$

$$\frac{\partial \Sigma_n^{-1}}{\partial \boldsymbol{\theta}}\Sigma_n = -\Sigma_n^{-1}\frac{\partial \Sigma_n}{\partial \boldsymbol{\theta}}, \qquad \Rightarrow \tag{4.13}$$

$$\frac{\partial \Sigma_n^{-1}}{\partial \boldsymbol{\theta}} = -\Sigma_n^{-1}\frac{\partial \Sigma_n}{\partial \boldsymbol{\theta}}\Sigma_n^{-1}. \tag{4.14}$$

And secondly the partial derivative of $\log |\Sigma_n|$ can be computed using Jacobi's formula (Bhatia and Jain, 2009), for any differentiable positive definite matrix Σ_n:

$$\frac{\partial |\Sigma_n|}{\partial \boldsymbol{\theta}} = \mathrm{tr}\left(\mathrm{adj}(\Sigma_n)\frac{\partial \Sigma_n}{\partial \boldsymbol{\theta}}\right), \tag{4.15}$$

$$= |\Sigma_n| \mathrm{tr}\left(\Sigma_n^{-1}\frac{\partial \Sigma_n}{\partial \boldsymbol{\theta}}\right). \tag{4.16}$$

Here $\mathrm{adj}(\Sigma_n) = |\Sigma_n|\Sigma_n^{-1}$ is used. Using the chain rule combining with Equation 4.16 we get:

$$\frac{\partial \log |\Sigma_n|}{\partial \boldsymbol{\theta}} = \mathrm{tr}\left(\Sigma_n^{-1}\frac{\partial \Sigma_n}{\partial \boldsymbol{\theta}}\right). \tag{4.17}$$

Substituting Equations 4.14 and 4.17 into Equation 4.11, the partial derivative of the log marginal likelihood can be written as:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \log p(\mathbf{y}|X, \boldsymbol{\theta}) = -\frac{1}{2}\mathrm{tr}\left(\Sigma_n^{-1}\frac{\partial \Sigma_n}{\partial \boldsymbol{\theta}}\right) + \frac{1}{2}(\mathbf{y} - m(X))^T\Sigma_n^{-1}\frac{\partial \Sigma_n}{\partial \boldsymbol{\theta}}\Sigma_n^{-1}(\mathbf{y} - m(X)), \tag{4.18}$$

$$= \frac{1}{2}\mathrm{tr}\left((\boldsymbol{\alpha}\boldsymbol{\alpha}^T - \Sigma_n^{-1})\frac{\partial \Sigma_n}{\partial \boldsymbol{\theta}}\right), \qquad \boldsymbol{\alpha} = \Sigma_n^{-1}(\mathbf{y} - m(X)). \tag{4.19}$$

Using a gradient ascent algorithm, one can find a local maximum by taking the following steps. Take an initial guess $\boldsymbol{\theta}^{(0)}$ and compute the gradient $\nabla \log p(\mathbf{y}|X, \boldsymbol{\theta}^{(0)})$. Then, the updated estimate for your hyperparameters ($\boldsymbol{\theta}^{(1)}$) is the gradient times a step-size $\gamma \in \mathbb{R}^+$ plus the initial guess. So in general, the i-th iteration is equal to:

$$\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)} + \gamma \nabla \log p(\mathbf{y}|X, \boldsymbol{\theta}^{(i-1)}), \qquad i = 1, 2, \ldots. \tag{4.20}$$

Just as in Equations 4.4 and 4.7, the inverse of the matrix Σ_n is the most dominant (computationally) in this process. However, using a gradient ascent algorithm, the inverse needs to be computed several times as the matrix Σ_n depends on the hyperparameters $\boldsymbol{\theta}$. The model will be trained on n in-situ observations and as n increases (more observations are done spatially and/or temporally), it is preferred to include these observations in training the model. Therefore, it is desired to find a method to speed up this process.

Note that the test data X_* is only used to compute the conditional distribution $\mathbf{f}_*|\mathbf{y}$ and not in updating the hyperparameters. So, when the number of satellite data k increases, the extra

time complexity depends on simple matrix-vector multiplications and matrix-matrix multiplications. The number of flops to compute the posterior mean (depending on k, so ignoring the inverse) in Equation 4.7 is $k(2n-1)+k = 2kn$, which scales linearly with k. Similarly, for the covariance matrix, the number of flops is equal to $2n^2k - nk + 2nk^2 = 2nk(n+k+\frac{1}{2})$, which scales quadratically with the number of satellite observations. Usually, one is only interested in the diagonal of the covariance matrix, so roughly $\frac{1}{k}$ part of the covariance matrix needs to be computed. The number of flops necessary to compute the diagonal of the covariance matrix (depending on k) is equal to $k(n+1)(2n-1)$, which scales linearly with k. Thus, we can conclude that the time complexity problem relies upon computing the inverse of a matrix size $n \times n$. To speed up this process a few methods are considered, which are analyzed in chapter 6. In the next chapter the variables, the choice for covariance functions and parameters are analyzed for the GPR model using the ODYSSEA in-situ data and Sentinel-2 satellite data.

5 Analysis

In this chapter, an analysis will be carried out concerning the Gaussian process regression discussed in the previous chapter. It is important to note that our data is incomplete because no observations are containing the information of all variables. The in-situ data records the chlorophyll-a concentration, location and depth, but no reflectances. On the other hand, the satellite data contains the reflectances on multiple locations, though the depth and chlorophyll-a concentration are not recorded. Ideally, we would like to estimate the chlorophyll-a concentration based on the location, reflectances and depth, whilst incorporating the relationship between these variables and the chlorophyll-a concentration. So we need to have an expression for the unobserved variables conditioned on the observed variables. The first idea is to create a GPR model to find estimations for the reflectances at the in-situ locations. Then, for a specific depth, another GPR model can be made to estimate the chlorophyll-a concentrations based on the relationship between the reflectances by the first GPR model and the chlorophyll-a concentration from the in-situ data. The problem is to integrate the uncertainty of the reflectances in the model. This has been attempted, however, due to impractical distributions, this was not feasible. Moreover, as we are dealing with high-resolution satellite data the uncertainty will be small and a reasonable estimate for the reflectances can be obtained. For simplicity, the GPR model to find estimations for the reflectances will be replaced by an interpolation method. After this, a GPR model can be constructed to find the chlorophyll-a concentration, where the estimation for the reflectances is assumed to be the true value.

First, the interpolation methods to find the reflectances will be explained. Then, based on these results an exploratory analysis is carried out. An explanatory analysis is done by discussing and applying different covariance functions and investigating the parameters of the GPR model.

5.1. Interpolation Reflectances

To find a statistical relationship between the chlorophyll-a concentration and the explanatory variables (e.g. the reflectances), data is needed that contains both variables at the same location and time. As said before, the data is incomplete: the measured reflectances will not be in the same location as the chlorophyll-a concentration. Thus, some interpolation is needed to estimate these reflectances. These estimates will then later be used to train the model, where it is assumed that the interpolated values are the true values. This assumption can be substantiated by the fact that the satellite data has a very high resolution.

There are many ways to interpolate this 2-dimensional problem, such as with inverse distance weighting (IDW) and bilinear or even bicubic interpolation.

5.1.1. Nearest-Neighbour Interpolation

The simplest technique which can be used for interpolation is the nearest-neighbour interpolation (NNI). This method assigns the value of the closest known point value to the unknown point. Thus a two-dimensional piecewise-constant function. In Figure 5.1, an example is shown where one can see a grid of 25 known observations and in color the assigned values to the unknown points in between them.

The nearest-neighbour method does not consider values of other neighbouring observations so no averaging or smoothing takes place.

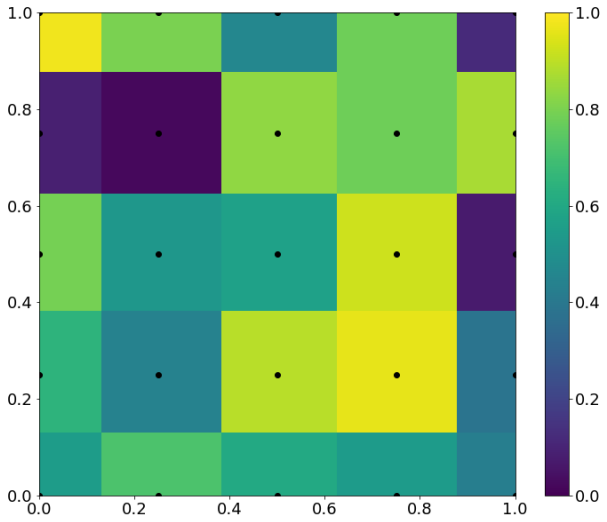


Figure 5.1: Nearest-neighbour interpolation by using a 5x5 grid of known observations (black). In color: the interpolated values can be observed and a check pattern appears because only the nearest observation is used to interpolate.

5.1.2. Inverse Distance Weighting

Inverse distance weighting (IDW) is a method that follows Tobler's first law of geography (Tobler, 1970): "Everything is related to everything else, but near things are more related than distant things" (p. 236). IDW does this, as the name suggests, by giving a higher weight to observations nearby and a lower weight to observations far away (Wikle et al., 2019).

Suppose the spatial data that is given by:

$$\{Z(s_1), Z(s_2), \ldots, Z(s_n)\},$$

where the spatial locations are $\{s_i, i = 1, \ldots, n\}$, so $Z(s_i)$ represents the observation at location s_i. To find the value for the unobserved point s_0, the weights need to be calculated by taking the inverse of the distance between the observed locations and the unobserved location, which is usually done using the Euclidean norm.

$$\tilde{w}_i(s_0) = \frac{1}{d(s_i, s_0)^p}, \qquad i = 1, \ldots, n.$$

$d(\cdot, \cdot)$ represents the Euclidean norm since we will be using spatial data, but any norm can be used here. Furthermore, the parameter p, the power coefficient, is a positive real number (i.e. $p \in \mathbb{R}^+$) that controls the smoothness of the method. Higher values for p result in higher weights for points close to the interpolated point. Low values for p, on the other hand, assign higher weights to points further away and results in a more smooth interpolation. The weights

need to be normalized after which they will be combined to compute the estimate for $Z(s_0)$.

$$w_i(s_0) = \tilde{w}_i(s_i)/\Sigma_{j=1}^{n}\tilde{w}_j(s_0), \qquad i = 1, \ldots, n.$$

$$\hat{Z}(s_0) = \Sigma_{i=1}^{n} w_i(s_0) Z(s_i).$$

In Figure 5.2 the interpolation for $p = 1$ (left) and $p = 4$ (right) can be seen, where the same observations are used as in Figure 5.1. One can indeed observe the smoothness and the importance of observations increasing for higher values of p. To estimate the power coefficient p a cross-validation technique can be used to optimize this parameter.

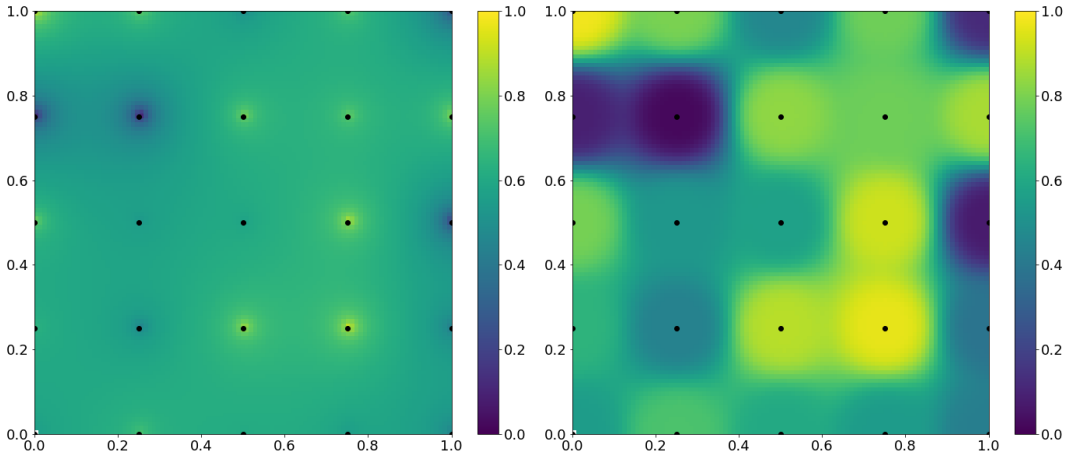


Figure 5.2: Inverse Distance Weighting (IDW) by using a 5x5 grid of known observations (black). In color the interpolated values can be observed using power coefficient $p = 1$ (left) and $p = 4$ (right). One can see that for lower values of p the interpolation is more smooth, whilst for higher values of p the importance of observations that are nearby is increasing.

5.1.3. Bilinear and Bicubic Interpolation

What in one dimension is called linear and cubic interpolation is in two dimensions *bilinear interpolation* and *bicubic interpolation*. Both methods only use a select number of observations nearby to compute the interpolated value. Furthermore, the data observations that they can handle must be a regular grid. As this is often the case in images (pixels of a picture are in a regular grid), the methods are often used in image processing (Hwang and Lee, 2004).

Bilinear interpolation uses the four closest (2x2 grid) observations and applies a linear interpolation in both dimensions. The order in which direction the interpolation is done does not influence the outcome. The function $f(x, y)$ in equation 5.1 is fitted in each grid between four points where the function value is known. Using the four function values, the four coefficients $\{a_{ij} | i, j \in \{0, 1\}\}$ can be computed.

$$f(x, y) = \sum_{i=0}^{1}\sum_{j=0}^{1} a_{ij} x^i y^j. \tag{5.1}$$

Bicubic interpolation uses the 16 closest (4x4 grid) observations and the problem uses 16 coefficients ($\{a_{ij} | i, j \in \{0, 1, 2, 3\}\}$) for each interpolation for matching the function values at each point. The function $f(x, y)$ in equation 5.2 is fitted in the center 2x2 grid. There are 16 coefficients, therefore the surrounding 12 observations are used to compute these.

$$f(x, y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij} x^i y^j. \tag{5.2}$$

At the edge of the domain, no 4x4 grid can be made and a bilinear interpolation is done instead.

In Figure 5.3 the interpolation using both methods can be seen. The same observations have been used as in Figure 5.2 and 5.1. One can observe that the bicubic interpolation is more smooth than the bilinear interpolation which is the result of using more pixels. Because the bicubic interpolation is using more pixels, the computation time is longer. For large datasets, this may become a practical problem and needs to be taken into account.

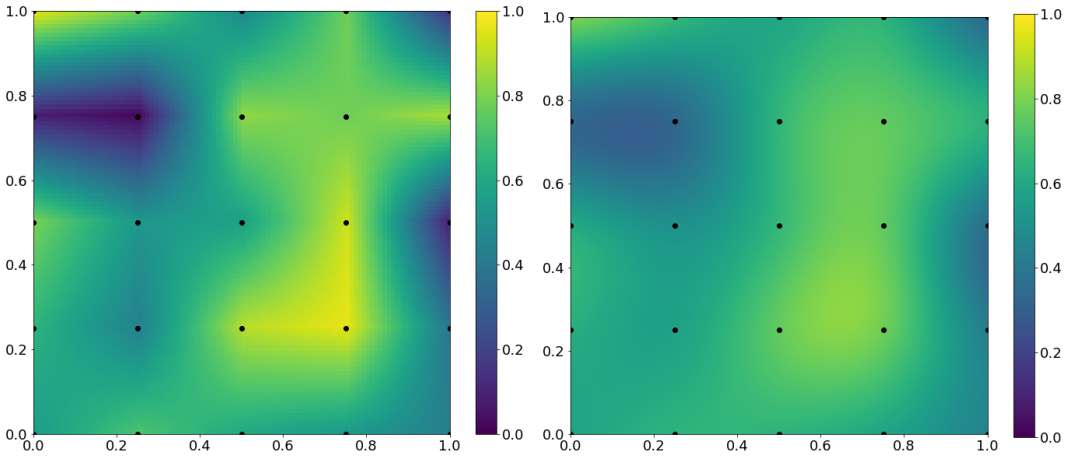


Figure 5.3: Bilinear (left) and bicubic (right) interpolation by using a 5x5 grid of known observations (black). In color the interpolated values can be observed. Because the bicubic interpolation is using more pixels than the bilinear interpolation it is more smooth.

5.1.4. Spline Interpolation

Spline interpolation, as the name suggests uses splines which are piecewise polynomials that are connected in the nodes in such a way that they ensure smoothness (Vuik et al., 2007). In a one dimensional example, a polynomial function of a certain degree p will be defined on each interval such that the function values and the $p-1$ derivatives are connected in the nodes.

For two dimensions the method is similar, however, also partial derivatives need to be computed. To compute these derivatives, finite differences or so-called spectral derivatives can be used (Enomoto, 2008).

For a cubic spline interpolation ($p=3$), the function in equation 5.2 is fitted on a 2x2 grid. The 16 coefficients are computed by using the function values of the 4 grid points and the 3 partial derivatives in each of those 4 grid points. To compute the partial derivatives, the surrounding grid points are needed. So a 4x4 grid is used to interpolate in the center 2x2 grid. In Figure 5.4 the interpolation can be seen using a 5x5 grid of observations. The advantage of spline interpolation in comparison with bicubic interpolation is that this technique does not require a (structured) grid. Instead, it will use the adjacent points to compute the coefficients.

5.1.5. Radial Basis Function Interpolation

The technique called radial basis function (RBF) interpolation uses a weighted sum of RBFs to interpolate. Thus an RBF interpolation can be performed on unstructured data (points need not lie on a grid).

A radial function is a function ϕ such that the output only depends on the distance between the input and some certain value (i.e. $\phi(x) = \phi(||x-c||)$). To find the value for the unobserved point s_0, one needs to compute $\hat{Z}(s_0) = \sum_{i=1}^{n} w_i \phi(||s_0 - s_i||)$ where we have n observations s_i. $w_1, \ldots, w_n$ are chosen (calculated) such that $\hat{Z}(s_i) = Z(s_i)$ for every observation s_i, this is done by solving $\Phi\mathbf{w} = \mathbf{Z}$ for $\mathbf{w} = [w_1, \ldots, w_n]^T$. Here, the matrix Φ has on the i-th row and

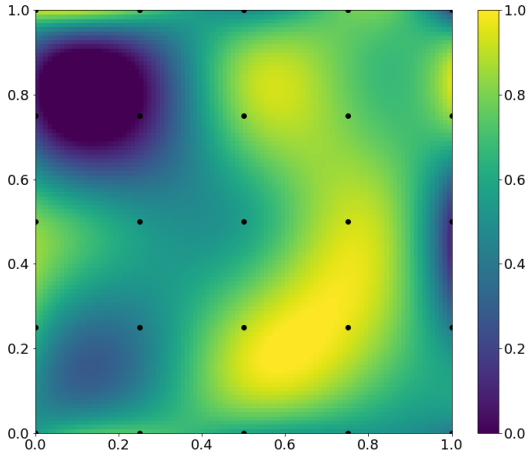


Figure 5.4: Cubic spline interpolation 5x5 grid of known observations (black). In color the interpolated values can be observed.

j-th column $\Phi_{i,j} = \phi(||x_i - x_j||)$ and $\mathbf{Z} = [Z(s_1), \ldots, Z(s_n)]^T$. There are a number of common used functions as radial function such as a Gaussian and the multiquadric function (Harpham and Dawson, 2006). These functions use a shape parameter $\epsilon \in \mathbb{R}^+$ that we are able to tune.

In Figure 5.5 the interpolation can be seen using the multiquadric function on the left and the Gaussian function on the right with the shape parameter $\epsilon = 1$. Visually the interpolations are very similar to the cubic spline interpolation, however, the possibility to modify the shape parameter and the radial basis function provide the RBF interpolation more freedom.

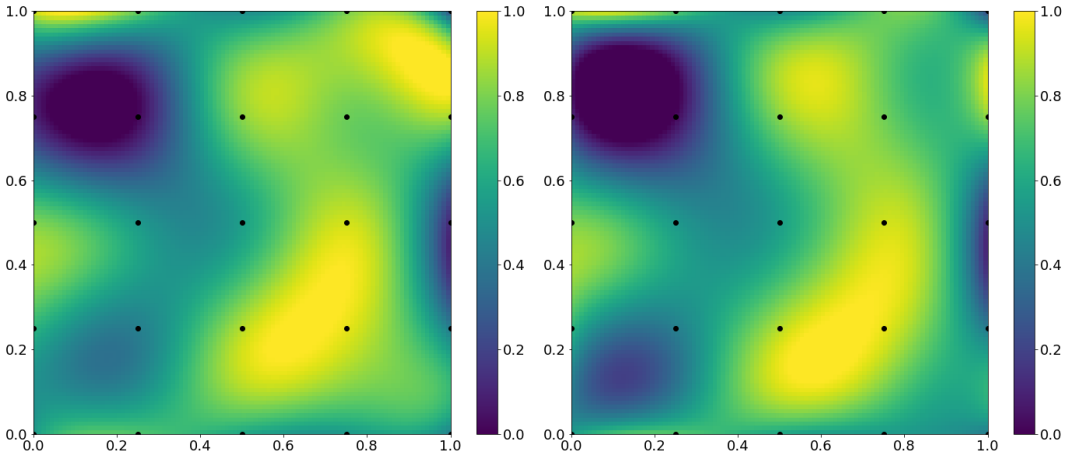


Figure 5.5: Radial basis function interpolation on a 5x5 grid of known observations (black). The type of radial basis function is multiquadric (left) and Gaussian (right) with the shape parameter $\epsilon = 1$. In color the interpolated values can be observed.

5.1.6. Results

To compare the performance of the interpolation methods, cross-validation will be applied to all methods and the mean squared error will be computed. First, a large region from the Sentinel-2 image is selected carefully, i.e. without any land and clouds containing the in-situ measurements that will be used later in the project.

Next, K-fold cross-validation is carried out. The number of groups (or folds) that the dataset is split into, K, need to be chosen carefully. There is a bias-variance trade-off related to the choice of K. As the number of observations n in satellite imagery gets large quite fast, a leave-

one-out cross-validation (LOOCV) is ruled out because of the computation time. A common choice to prevent a high bias or a high variance is $K = 10$ or $K = 5$ (James et al., 2013, Wikle et al., 2019). For this study, $K = 10$ is chosen as it seems to be able to represent the data quite well.

Let us use the same notation as before and let Z_i be the observations from the satellite image, where $i = 1, \ldots, n$ (n being the number of observations). The interpolation on the kth fold is denoted by $\hat{Z}_i^k$ where $k = 1, \ldots, K$ and $i = 1, \ldots, n_k$ with n_k being the number of observations in the kth fold. The mean squared error is then computed for the four bandwidths for every method.

$$MSE_k = \frac{1}{n_k} \sum_{i=1}^{n_k} (Z_i - \hat{Z}_i^k)^2.$$

The total mean squared error for each bandwidth is the average over the number of folds, this will be denoted by $MSE_{(K)}$.

$$MSE_{(K)} = \frac{1}{K} \sum_{k=0}^{K} MSE_k.$$

This metric is computed for every method, if possible, and is used to estimate the parameters of the methods for which the total mean squared error is minimal.

The area used is in the northern part of the Aegean Sea, the Thracian Sea. The number of observations is only $n = 522$. Comparing with the number of observations that will be used later on, this is a small number because of the high computational time for some methods that we will see later on.

For the inverse distance weighting, the power parameter p needs to be investigated. For every bandwidth, the total mean squared error is computed using $K = 10$ and the results can be obtained from Figure 5.6. Here the mean squared error is computed for different values of p and it is done for every bandwidth. This suggests that a choice for $p = 2.0$ will result in the best performance in terms of mean squared error. Note that the standard deviation (as this is an average of 10 interpolations) is approximately $4 \cdot 10^{-7}$ for every bandwidth.

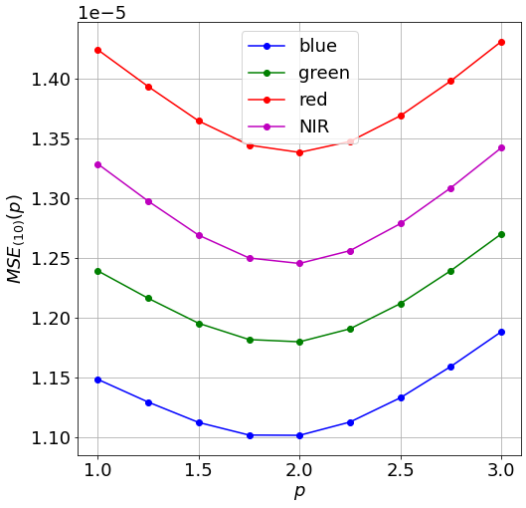


Figure 5.6: The mean squared error for different values of p doing IDW. In color the four different bandwidths that are obtained from the Sentinel-2 satellite. The subset of the image, taken from the Thracian Sea on August 9th 2019, contains 18330 observations.

For the radial basis function, there are multiple things to choose before comparing with the other methods. The type of function used and the shape parameter ϵ needs to be determined.

For different values of ϵ, we perform the cross-validation method and compute their mean squared error. This is done for the following functions: *multiquadric*, *inverse* and *Gaussian*. Other functions that are considered, but do not contain a shape parameter are: *linear*, *cubic*, *quintic* and *thin plate*. A first observation was that a relative high choice ($\epsilon > 1$) for ϵ resulted in unstable solutions because of ill-conditioned matrices. A good choice for the shape parameter is close to the minimum distance between the points that are observed (Mongillo, 2011). In our case that is approximately $7 \cdot 10^{-4}$ and 10^{-5} decimal degrees in the longitude and latitude direction respectively. In Figure 5.7, one can see the mean-squared-error for the three different basis functions that depend on the shape parameter. Note that the log-scale is sometimes used to visualize the results better. For the multiquadric function, a shape parameter $\epsilon = 10^{-5}$ is suggested, as the mean squared error is approximately constant for lower values. For the inverse function, the blow-up of the mean squared error is visible because of an ill-conditioned matrix. Here, a shape parameter $\epsilon = 4 \cdot 10^{-4}$ is chosen. Again, for the Gaussian function, a blow-up of the mean squared error can be seen. The choice $\epsilon = 8.2 \cdot 10^{-4}$ is here suggested. Note that for all these estimations $MSE_{(K)}$ is computed, the corresponding standard deviation is approximately 10^{-6}.

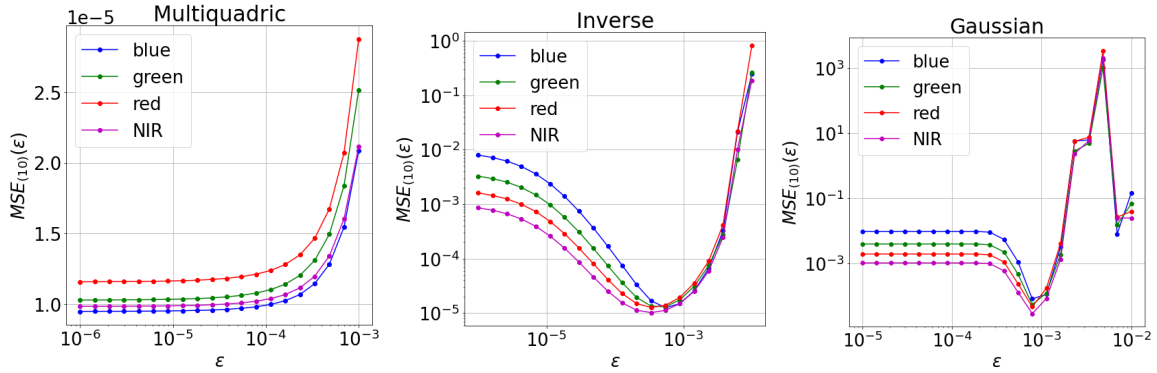


Figure 5.7: The mean squared error for different values of ϵ doing RBF interpolation. The multiquadric function (left), the inverse function (middle) and the Gaussian function (right) is used. Note the log scaled x-axes and y-axes (except for the y-axis using the multiquadric function).

In Table 5.1 the mean squared error $MSE_{(K=10)}$ can be obtained for each radial basis function and each bandwidth. The Gaussian function is performing worst for all four bandwidths, whereas the linear and multiquadric functions have the lowest total mean squared error.

Function	blue	green	red	NIR
Multiquadric	9.53e-06	1.04e-05	1.16e-05	9.90e-06
Inverse	1.44e-05	1.28e-05	1.28e-05	1.03e-05
Gaussian	6.40e-05	4.96e-05	4.74e-05	2.92e-05
Linear	9.50e-06	1.03e-05	1.16e-05	9.85e-06
Cubic	1.37e-05	1.61e-05	1.83e-05	1.47e-05
Quintic	2.19e-05	2.51e-05	2.98e-05	2.32e-05
Thin Plate	1.14e-05	1.30e-05	1.46e-05	1.20e-05

Table 5.1: $MSE_{(K=10)}$ for each function and bandwidth using RBF interpolation. The shape parameters used are: Gaussian $\epsilon = 8.2e-4$, Inverse $\epsilon = 4e-4$, Multiquadric $\epsilon = 1e-5$.

Finally, the results for nearest-neighbour interpolation (NNI), IDW, cubic spline and the RBF interpolation are summarized in Table 5.2. One can see that the total mean squared error is minimal using IDW for every bandwidth. Note that the simplest interpolation method, NNI, has

a similar performance as the more advanced interpolation methods. In terms of computational time, NNI and IDW are significantly faster than the cubic spline and RBF interpolation. Based on these results, IDW is used to interpolate the reflectances on the in-situ locations. It is now possible to use a Gaussian process regression as observations are containing both the chlorophyll-a concentration as well as the (interpolated) reflectances, location, depth and time.

Method	blue	green	red	NIR
NNI	1.36e-05	1.56e-05	1.74e-05	1.52e-05
IDW	7.70e-06	7.90e-06	8.80e-06	7.90e-06
Spline	1.29e-05	1.51e-05	1.58e-05	1.36e-05
RBF	9.50e-06	1.03e-05	1.16e-05	9.85e-06

Table 5.2: The total mean squared error ($MSE_{(K=10)}$) for the nearest-neighbour interpolation, inverse distance weighting, cubic spline and radial basis function using the linear function. The power parameter $p = 2$ is used doing IDW.

5.2. Exploratory Analysis

The 10-meter resolution satellite data obtained by the Sentinel-2B satellite is used. The data product is from 9 August 2019 and the product level is 2A. Using SNAP, a NetCDF file can be saved containing the four bands (blue, green, red and NIR) and the classification map. The images contain 10980 by 10980 pixels for each bandwidth. As this is quite a lot of data, considering the memory capacities, a subset of this data is used in the analysis. The in-situ data from 9 August 2019 contains 2866 observations in the area of the satellite image. Though, temporally these observations are taken approximately every 30 seconds. In this analysis, they are assumed to be obtained at the same time as the satellite image is taken. Because of the result of the last section, IDW is applied to obtain the reflectances at the in-situ locations. In Figure 5.8, the green reflectance ratio is shown (satellite data) as well as the in-situ data (circles). The color represents the value of the reflectance ratio, so visually one can see that the interpolated values match with the surrounding values. Furthermore, it can be seen that the location of the in-situ observations can be categorized into 7 clusters. For each of these clusters, observations are done for different depths.

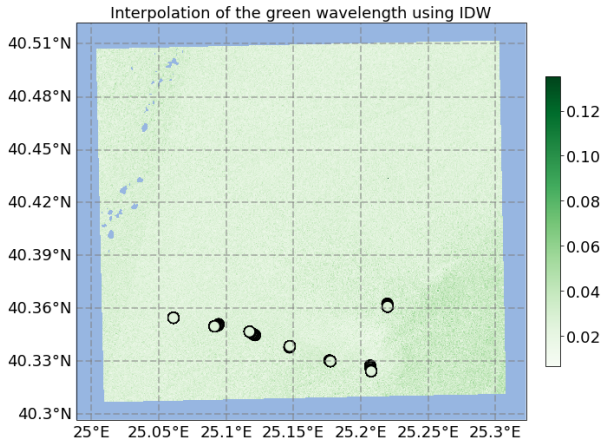


Figure 5.8: Interpolation of the green wavelength using IDW. The color represents the value of the reflectance ratio. The in-situ locations are denoted by circles, for which the color represents the interpolated value. It can be seen that there are seven clusters of observations that are near each other (location wise).

The clusters have been given a number ranging from 1 to 7 from left to right. Then, the

number of observations at different depths for each of these clusters is visualized by a histogram in Figure 5.9. From this histogram, it can be seen that there are many observations done for depths 0-20 meters compared to the rest. By making the bins smaller, it becomes clear that there are approximately 200 observations done in the first two meters. Observations are quite well spread between a depth of 40 and 500 meters and observations deeper than 500 meters are (almost) only done in cluster 6. By the use of colors, one can see that in every range, multiple observations are done in every cluster.

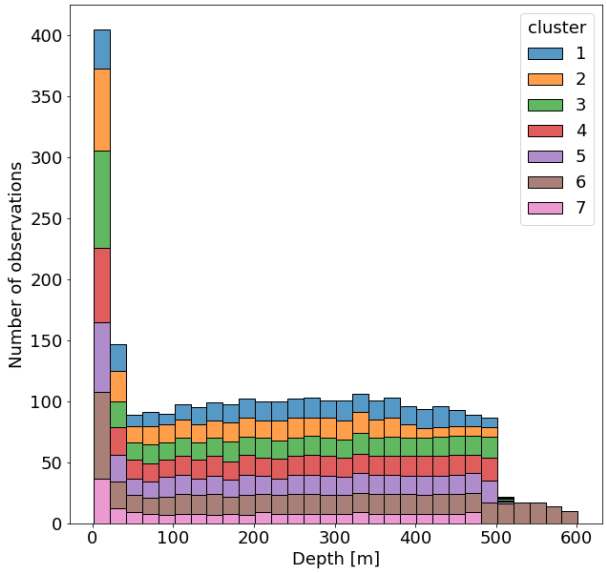


Figure 5.9: Histogram of the depth of the 2866 in-situ observations. In color the seven clusters are represented that can be seen in Figure 5.8. Many observations are done close to the surface.

With the interpolated reflectances, Spearman's rank correlation coefficient can be calculated, see Figure 5.10 (left). The correlation between each variable is visualized by a colored square: the brighter/larger the color/square the more correlated the variables are (positively or negatively). It can be seen that the reflectances are positively correlated with each other and that chlorophyll-a is only correlated with the depth (negatively). The longitude and latitude are negatively correlated with each other and the correlation is close to zero for the other variables, except for the red wavelength. An interesting thought is: what will happen when only the observations are used that are close to the surface (say depth is smaller than 2 meters)? Light can penetrate the water for only a few meters, so naturally, the reflectance and the chlorophyll-a concentration at a depth of 500 meters will not have any significant relationship.

In Figure 5.10 on the right, the correlation between the chlorophyll-a concentration and the wavelengths can be seen as a function of increasing depth. For every meter, the observations above this depth level are used, so at a depth of fifty meters, all observations with a depth ranging from zero to fifty are used to compute the correlation coefficient. One can see that the correlation is high (negatively) for observations close to the surface and gets lower (closer to zero) when deeper observations are included. The red wavelength tends to be the least informative, whilst the green wavelength seems to be the most informative which is in accordance with Figure 1.2.

Further research into the relationship between the chlorophyll-a concentration and the reflectances (and the ratio of reflectances) is done later on when the linear and polynomial models from the literature are compared with the GPR model. Additionally, it is of interest to investigate the influence of the depth (measured in meters below sea level) on the chlorophyll-a measurement. A scatter plot of the depth versus the chlorophyll-a concentration has been

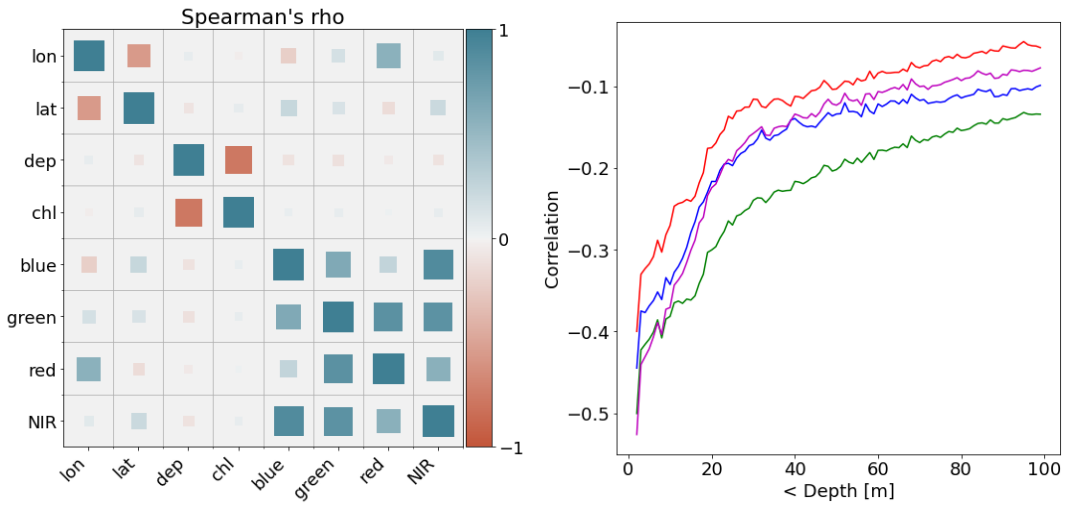


Figure 5.10: The spearman rank correlation coefficient between every two variables (left). The value is represented by color (blue: positive, red: negative) and size/brightness (larger/brighter: strong correlation, small/transparent: low correlation). On the right: the correlation between the chlorophyll-a concentration and the wavelengths for increasing depth. For a certain depth, the observations with a lower depth are used to compute the correlation. Clearly, for large depths, low correlation are computed (close to zero). Each color is the corresponding wavelength, with purple for the NIR wavelength.

made as well as a Locally Weighted Scatter plot Smoothing curve (LOWESS), see Figure 5.11. LOWESS is a non-parametric regression technique, so no assumptions are made about the distribution of the data. From this figure, it is clear that there is a relationship between the chlorophyll-a concentration and the depth of the measurement. The concentration seems to increase until a depth of approximately 90 meters. After this, the concentration decreases until a depth of approximately 250 meters is reached. Once this is reached, the concentration stays relatively constant and close to zero.

This peak of chlorophyll-a concentration is also known as the deep chlorophyll maximum (DCM), which is the region below the surface of the water that contains the most chlorophyll concentration (Cullen, 1982; Huisman et al., 2006). This is a common feature in oligotrophic waters where the available nutrients in the surface region are depleted. Therefore, depth is an important variable in the analysis of this research and has to be taken into account when doing so.

5.3. Temporal Modelling

GPR was introduced as a spatio-temporal model, so observations from multiple days can be included to estimate eventually the chlorophyll-a concentrations at a certain point in time. The in-situ data supports temporal modelling because measurements are taken 24 days, twice every minute. In chapter 3, we established that only three data-products from Sentinel-2 are available in the time range of the ODYSSEA dataset. The in-situ data from the corresponding days can be included and the reflectances interpolated on the in-situ locations. Again, it is assumed here that the in-situ measurements are taken at the same time as the satellite measures the reflectances.

Ideally, for the analysis of the variable time, the measurements by the glider are done at similar coordinates for different days. However, in our problem, the glider is in a different area of the Thracian sea for each of these days. As there is not a lot of time difference between the observations (maximum a few days), there is not a lot of change in chlorophyll-a concentration present (for a similar depth). To capture the structure of seasonal change, at least some data

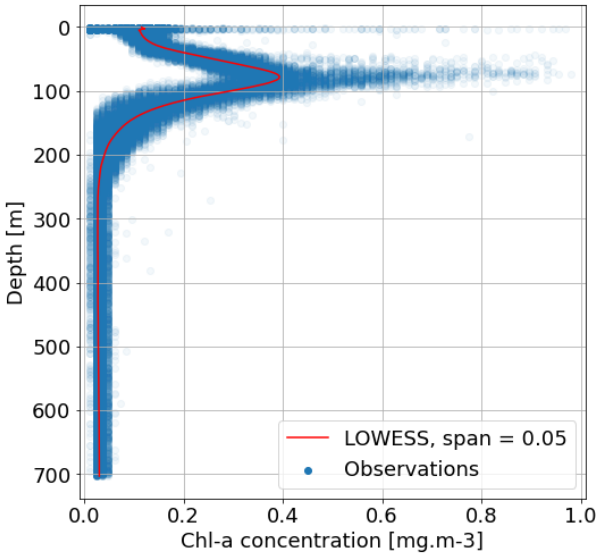


Figure 5.11: Nonparametric regression by local averaging of the chlorophyll-a concentration on the depth. Each local average is based on approximately 3300 observations (i.e., a span of 0.05). It can be seen that the concentration of chlorophyll-a peaks at a depth of 90 meters and stays relatively constant for measurements taken deeper than 250 meters. All measurements from the in-situ data are used here.

from every season is needed. In our problem, three days from mid-August are available which is clearly not enough to capture this seasonal effect. For these reasons, the focus of this study is on spatial statistics. Though, in the discussion some ideas and suggestions are done for future research to include the variable time as well.

5.4. Covariance Functions

In Section 4.3, details of the Gaussian process regression has been discussed. Here, we glanced at the covariance matrix that is created using a covariance function. The choice for this function has a crucial impact on the final results (Dunson et al., 2020). Usually, a covariance function is chosen based on expertise and knowledge about the application, though a clear answer when specific functions are preferable is unavailable (Kang et al., 2017).

The choice for the covariance function is the main assumption of the Gaussian process regression that incorporates the idea that observations close to each other will have similar behaviour. So, a test observation will predominantly use the information from close observations. The word *close* suggests that a location in two- or three-dimensional space of some sort is meant. However, in our case, an observation contains the location as well as the time, depth and reflectances. So, two observations can be close in time and reflectances but far apart (spatially) nonetheless.

The function $k(x, x') = cov(f(x), f(x'))$ is defined as a covariance function when it solely depends on the input variables, is symmetric and is positive semi-definite. The first statement about the input variables is done because one wishes to compute the covariance between observed and unobserved data. The symmetric property comes by definition from the covariance itself ($cov(f(x), f(x') = cov(f(x'), f(x))$). Finally, positive semi-definiteness is needed as the covariance matrix is defined to be positive semi-definite. The covariance matrix of $X = [x_1, \ldots, x_n] \in \mathbb{R}^n$ can be written in terms of the covariance function, namely, every i, j-th

element of the matrix is the covariance function evaluated at x_i and x_j.

$$K(X,X) := \begin{bmatrix} k(x_1,x_1) & \dots & k(x_1,x_n) \\ \vdots & \ddots & \vdots \\ k(x_n,x_1) & \dots & k(x_n,x_n) \end{bmatrix}. \tag{5.3}$$

A covariance function $k(x,x')$ that only depends on $r = x - x'$ is called *stationary*. The evaluation does solely depend on the difference between the observations and does not depend on absolute position. Furthermore, if the covariance function depends on $r = |x - x'|$, the covariance function is called *isotropic*. Now, the function depends only on the absolute difference between the two observations. Often, a different norm is used such as the euclidean norm, however, this will indirectly treat the variables such as time and space the same. Finally, a covariance function that only depends on $x \cdot x'$ is called a *dot product* covariance function.

Another convenient property of covariance function is that it is possible to multiply and sum covariance functions while the result will still be a covariance function. This way it is possible to treat variables differently in the covariance function. For example, let x denote the position in space and t the position in time, a covariance function k of two observations (x,t) and (x',t') can be written as

$$k((x,t),(x',t')) = k^{(s)}(x,x')k^{(\tau)}(t,t'), \tag{5.4}$$

where $k^{(s)}$ is the covariance function specified for the space and $k^{(\tau)}$ for the time.

In the following subsections, several covariance functions will be discussed including their characteristics and performance.

5.4.1. Squared Exponential Kernel

Probably the most commonly used covariance function is the squared exponential kernel, also known as radial basis function kernel or Gaussian kernel (Rasmussen and Williams, 2006; Wikle et al., 2019). The kernel function of two observations x and x' is given by:

$$k_{SE}(x,x') = \sigma_f^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right). \tag{5.5}$$

Here, the hyperparameter $\sigma_f^2 > 0$ denotes the process variance which is used in all covariance functions and operates as a scale factor. This will influence the variation of the function from the mean. So, if the process variance is taken too small, the covariance function will be close to zero, no matter the input, which results in a modelled function very close to the mean. However, if the process variance is taken too large, it will deviate from the mean easily which makes it possible to capture outliers.

The hyperparameter l denotes the (characteristic) lengthscale and determines the smoothness of the modelled function. When l is small, the covariance functions depends more heavily on $r = (x-x')^2$ thus it will be less smooth. Meanwhile, when l is large, the influence of r is less and as a result, the modelled function will be smoother.

For now, we treated x as a one dimensional variable, however, in our case, there are 7 variables that are considered as input values and we will write them as $\mathbf{x} = [x_1, \dots, x_7]^T$. Now, the covariance function can be written as:

$$k(\mathbf{x},\mathbf{x}') = \sigma_f^2 \exp(-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^T\Lambda^{-1}(\mathbf{x}-\mathbf{x}')), \qquad \Lambda = \text{diag}(l_1^2,\dots,l_7^2).$$

This way there is a lengthscale parameter for each variable. This is essentially multiplying 7 squared exponential kernels, one for each variable but having one scaling parameter σ_f^2.

5.4.2. Matérn Kernel

A generalization of the squared exponential kernel is the Matérn kernel, named after the Swedish statistician Bertil Matérn.

$$k_M(x,x') = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{|x-x'|}{l} \right)^{\nu} K_\nu \left(\sqrt{2\nu} \frac{|x-x'|}{l} \right). \tag{5.6}$$

Where $K_\nu(\cdot)$ is the modified Bessel function, $\Gamma(\cdot)$ is the Gamma function and the variables ν and l are two positive hyperparameters. Again, σ_f^2 is included as the process variance. The realisations from the Gaussian process are almost surely $\lceil \nu \rceil - 1$ times differentiable using the Matérn kernel (Santner et al., 2003), therefore ν can be referred as the smoothness parameter. This function can look quite complex at first, though using Formula 10.2.15 from (Abramowitz and Stegun, 1964) we can rewrite the modified Bessel function using $\nu = p + \frac{1}{2}$ for $p \in \mathbb{N}_0$.

$$K_{\nu=p+\frac{1}{2}}(z) = \sqrt{\frac{\pi}{2z}} \exp(-z) \sum_{k=0}^{p} \frac{(p+k)!}{k!(p-k)!} (2z)^{-k}, \qquad z \in \mathbb{R} \backslash \{0\}. \tag{5.7}$$

To rewrite the complete kernel the following equality is used for the Gamma function:

$$\frac{1}{\Gamma(p+\frac{1}{2})} = \frac{2^{2p}}{\sqrt{\pi}} \frac{\Gamma(p+1)}{\Gamma(p+2)}.$$

The square root of π cancels out and all exponents of base 2 and $z = \sqrt{2\nu}\frac{|x-x'|}{l}$ (outside the sum) end up to be p. Rewriting everything in terms of p gives us:

$$k_{\nu=p+\frac{1}{2}}(x,x') = \sigma_f^2 \frac{\Gamma(p+1)}{\Gamma(p+2)} \exp\left(-\sqrt{2p+1} \frac{|x-x'|}{l} \right) \sum_{k=0}^{p} \frac{(p+k)!}{k!(p-k)!} \left(2\sqrt{2p+1} \frac{|x-x'|}{l} \right)^{p-k}. \tag{5.8}$$

Though the equation may still look messy, choosing small numbers for p give much simpler expressions and are often used as kernel (Rasmussen and Williams, 2006). For $p = 0, 1$ and 2 the Matérn kernel is implemented in the GPy package for Python (GPy, since 2012).

$$k_{\nu=\frac{1}{2}}(x,x') = \sigma_f^2 \exp\left(-\frac{|x-x'|}{l} \right), \tag{5.9}$$

$$k_{\nu=\frac{3}{2}}(x,x') = \sigma_f^2 \left(1 + \sqrt{3} \frac{|x-x'|}{l} \right) \exp\left(-\sqrt{3} \frac{|x-x'|}{l} \right), \tag{5.10}$$

$$k_{\nu=\frac{5}{2}}(x,x') = \sigma_f^2 \left(1 + \sqrt{5} \frac{|x-x'|}{l} + \frac{5(x-x')^2}{3l^2} \right) \exp\left(-\sqrt{5} \frac{|x-x'|}{l} \right), \tag{5.11}$$

$$\lim_{\nu \to \infty} k_\nu(x,x') = \sigma_f^2 \exp\left(-\frac{(x-x')^2}{2l^2} \right). \tag{5.12}$$

In the last equation, the squared exponential kernel shows up when ν approaches infinity. This confirms that the Gaussian process is infinitely differentiable using a squared exponential kernel. According to Stein (2012), an infinitely differentiable Gaussian process is an unrealistic property for something that models a physical process. Therefore, Stein suggested the Matérn kernel instead of the squared exponential kernel, to model physical processes such as the chlorophyll-a concentration.

In Figure 5.12 (left), the covariance functions in Equations 5.9 - 5.12 are plotted versus the absolute distance. Here, the influence of the smoothness parameter ν can be observed, for higher values of ν the covariance function decreases less rapidly (for small absolute distances). The process variance is set to $\sigma_f^2 = 1$. On the right, a sample is plotted using the four different covariance functions using Formula 4.2. The difference is smoothness is well visible, note that these are random samples from a multivariate normal distribution. So, the exact path of the samples will differ, however, the shape of the sample will remain similar.

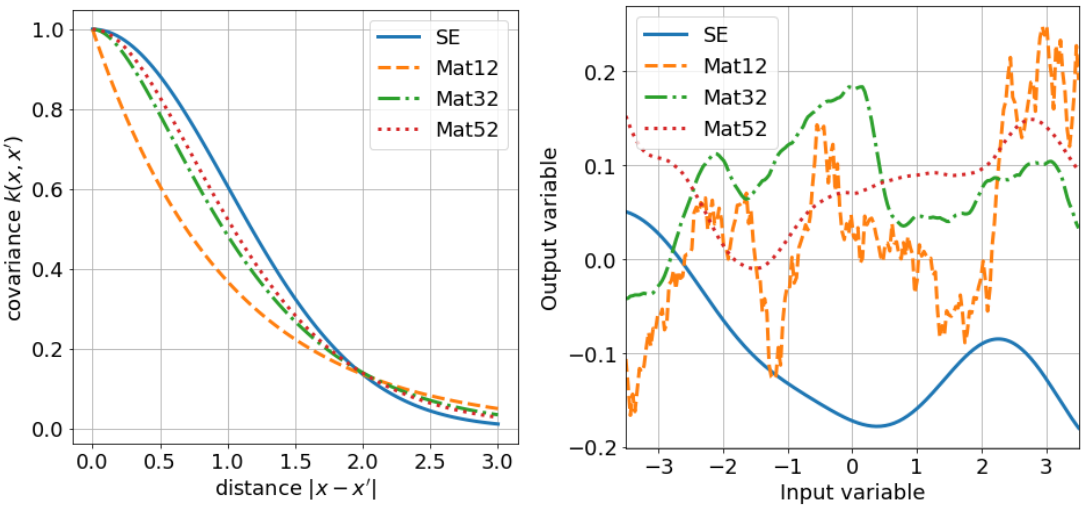


Figure 5.12: Matérn covariance functions for different values of ν while $\sigma_f^2 = 1$ (left). Samples of the prior using the different covariance functions (right). Mat12 is the Matérn kernel using $\nu = 1/2$, similar for the other functions. One can see that the squared exponential (SE) kernel is the most smooth kernel of all Matérn kernels.

5.4.3. Linear Kernel

The simplest example of a dot product kernel is the linear kernel which can be written as:

$$k_{Lin}(x, x') = \sigma_0^2 + \sigma_f^2(x \cdot x'). \tag{5.13}$$

Where σ_0^2 functions as a bias, σ_f^2 operates again as the process variance and the (dot) product is simply the multiplication of the variables. The idea of this kernel is completely different than for the Matérn kernels. Before, the kernels depended on the difference between two points, the smaller the difference the higher the covariance is. For the linear kernel, the covariance depends on the inner product of two points, so when the inner product is high, the covariance is high. This also makes it possible to retrieve negative values for the covariance function. Furthermore, this product makes the linear kernel invariant under rotation around the origin, despite that it is not invariant under translation. Positive observations close to zero have a much lower covariance than the observations far away from the origin. Using a linear kernel boils down to doing Bayesian linear regression, so samples from the prior using a linear kernel will be linear, hence the name of the kernel.

When the relationship of the variables is expected to be linear, this kernel can be used. However, often this kernel is used in combination with other kernels (Rasmussen and Williams, 2006).

5.4.4. Multilayer Perceptron Kernel

The multilayer perceptron kernel (MLP kernel), also called as the neural network kernel, is another example of a dot product kernel. The covariance function can be written as:

$$k_{MLP}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \frac{2}{\pi} \arcsin\left(\frac{\sigma_v^2 \mathbf{x}^T \mathbf{x}' + \sigma_0^2}{\sqrt{\sigma_v^2 \mathbf{x}^T \mathbf{x} + \sigma_0^2 + 1}\sqrt{\sigma_v^2 \mathbf{x}'^T \mathbf{x}' + \sigma_0^2}}\right). \tag{5.14}$$

To explain the variables, first the process to derive this kernel is explained. The derivation of this formula starts by considering a single (hidden) layer neural network. By following the procedure of Neal (2012) and Williams (1998), we start with an input x, a bias b and weights w_i for each neuron i in the hidden layer to the output neuron and weights v_i for the input neuron to each neuron i in the hidden layer. So, w_1 is the weight from the first neuron in the hidden

layer to the output neuron. Having $N \in \mathbb{N}$ number of neurons in the hidden layer we obtain $f(x)$ by computing:

$$f(x) = b + \sum_{i=1}^{N} w_i h(x, v_i). \tag{5.15}$$

Here, the function $h(x, v_i)$ is the transfer function which is assumed to be bounded. The core idea now, according to Neal (2012), is to let the number of neurons in the hidden layer go to infinity. As a result, this network can approximate any continuous function arbitrarily closely assuming a compact domain (Hornik et al., 1989). We set a prior to the bias and weights w_i independently with a zero mean and standard deviations σ_b and σ_w respectively. Furthermore, we let the weights v_i be independent and identically distributed. We now wish to compute $cov(f(x), f(x')) = \mathbb{E}(f(x)f(x')) - \mathbb{E}(f(x))\mathbb{E}(f(x'))$.

First, let us compute the expectation of $f(x)$ by computing the expectation of each element in the sum first.

$$\mathbb{E}(w_i h(x, v_i)) = \mathbb{E}(w_i)\mathbb{E}(h(x, v_i)) = 0,$$

for all i since w_i is independent and its expectation is zero by assumption. The variance of each element in the sum can be calculated by again using independence and the assumptions:

$$\mathbb{E}((w_i h(x, v_i))^2) = \mathbb{E}(w_i^2 h(x, v_i)^2) = \mathbb{E}(w_i^2)\mathbb{E}(h(x, v_i)^2) = \sigma_w^2 V(x).$$

We can write $\mathbb{E}(h(x, v_i)^2) = V(x)$ for all i since v_i is iid by assumption and the transfer function is assumed to be bounded. Then, by the Central Limit Theorem, for large N we can write the prior for $f(x)$ to be Gaussian with zero mean and variance $\sigma_b^2 + N\sigma_w^2 V(x)$. Using $\sigma_w = \omega N^{-1/2}$ we can let $N \to \infty$ and the variance of the prior converges to $\sigma_b^2 + \omega^2 V(x)$. Finally, for two different input variables x and x' we get:

$$\begin{aligned}
\mathbb{E}[f(x)f(x')] &= \mathbb{E}\left[\left(b + \Sigma_{i=1}^{N} w_i h(x, v_i)\right)\left(b + \Sigma_{i=1}^{N} w_i h(x', v_i)\right)\right], && (5.16)\\
&= \mathbb{E}[b^2 + b(\Sigma_{i=1}^{N} w_i h(x', v_i) + \Sigma_{i=1}^{N} w_i h(x, v_i)) && (5.17)\\
&+ \Sigma_{i=1}^{N} w_i h(x, v_i) \times \Sigma_{i=1}^{N} w_i h(x', v_i)], && (5.18)\\
&= \sigma_b^2 + 0 + \Sigma_{i=1}^{N} \sigma_w^2 \mathbb{E}(h(x, v_i)h(x', v_i)), && (5.19)\\
&= \sigma_b^2 + \omega^2 C(h(x, v_i), h(x', v_i)). && (5.20)
\end{aligned}$$

Where in the final equality $C(x, x') = \mathbb{E}(h(x, v_i)h(x', v_i))$ is used (which does not depend on i). Also, in Equation 5.19 it is used that the expectation of $w_i h(x, v_i) w_j h(x, v_j)$ for $i \neq j$ is equal to zero. We generalize this approach by having n input variables by writing x as a vector $\mathbf{x}$ and weights from the input variables to neuron i, v_i, as vector $\mathbf{v}_i$. Choosing the error function as the transfer function, i.e. $h(\mathbf{x}, \mathbf{v}_i) = \text{erf}(v_0 + \Sigma_{j=1}^{n} v_j x_j)$ with $\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$, we obtain Formula 5.14 (Williams, 1998). Here, the assumption is made that any $\mathbf{v}_i \sim \mathcal{N}(0, \Sigma)$ with $\Sigma = \text{diag}(\sigma_0^2/2, \sigma_v^2/2, \ldots, \sigma_v^2/2)$.

This procedure of Neal (2012) and Williams (1998) makes it clear how the variables in the covariance function originate. σ_v^2 is twice the variance of each weight from the input neurons to the neurons in the hidden layer. Furthermore, σ_0^2 is twice the variance of v_0 which operates as a bias. In Figure 5.13, one can see the effect of changing the variance of the weights, as this will produce samples that are able to vary more. Moreover, for large values of the input variable x and $-x$ the samples tend to go to equilibrium. This can be explained by the covariance matrix, for positive and negative input values the covariance is positive and as the input values x ($-x$) gets larger (smaller), the covariance increases with the surrounding input values.

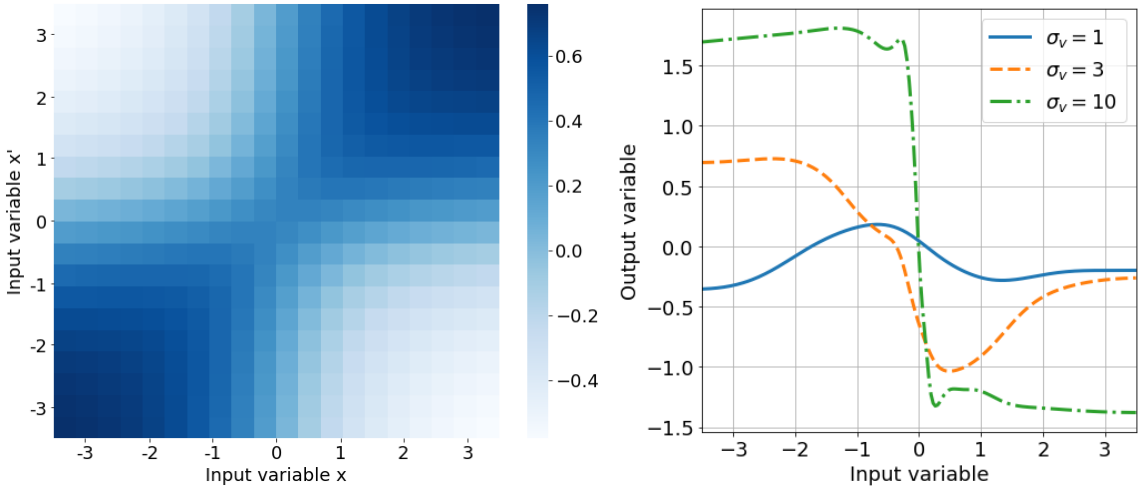


Figure 5.13: Covariance matrix (left) using the MLP kernel with parameters $\sigma_f^2, \sigma_v^2, \sigma_0^2$ equal to 1. Samples of the prior using the different values for σ_v while σ_f^2, σ_0^2 are set equal to 1 (right). One can see that the samples with a higher variance of weights vary more quickly.

Other options for the choice of the transfer function are possible, such as the modulated squared exponential (Rasmussen and Williams, 2006). By the constraint of having a positive definite kernel, the hyperbolic tangent function which can operate as a sigmoid function is not an option to be a valid kernel. This can be shown by considering two observations, $X = \{x_1, x_2\}$, and the hyperbolic tangent function as kernel (i.e. $k(x, x') = \tanh(a + x \cdot x')$). The covariance matrix is then constructed in terms of the covariance function:

$$K(X, X) = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) \\ k(x_2, x_1) & k(x_2, x_2) \end{bmatrix}. \tag{5.21}$$

Then the eigenvalues can be calculated by solving the following equation (note that $k(x_1, x_2) = k(x_2, x_1)$ by definition).

$$\lambda_{1,2} = \frac{k(x_1, x_1) + k(x_2, x_2) \pm \sqrt{(k(x_1, x_1) - k(x_2, x_2))^2 + 4k(x_1, x_2)^2}}{2}. \tag{5.22}$$

For a positive definite kernel the eigenvalues of matrix $K(X, X)$ should have strict positive eigenvalues for any choice of observations x_1, x_2 and parameter a. Choosing $(a, x_1, x_2) = (-2, 1, 3)$ (for example) results in the eigenvalues $\lambda_{1,2} \approx (1.28, -1.05)$. Hence, we can conclude that the hyperbolic tangent kernel is not a valid kernel.

5.4.5. Results

To decide which kernel will be used to create the covariance matrices, the cross-validation technique is used again (Mohammed and Cawley, 2017). The hyperparameters of the GPR model will be optimized with a certain part of the data and then predicted on the rest of the data. Then, the predicted value can be compared with the observed value and this is done for all kernels that are described. One thing to mention is that the part of the data that will be used to train/test can not be selected randomly as we did before with the interpolation methods. This is due to the fact that some of the observations in the in-situ data are very close in space. One prefers to know the error that you make at locations where no observations are done. So target-oriented cross-validation will be carried out to compute the performance for each of the kernels. This is done by splitting the data into seven groups (the seven clusters that are observed in Figure 5.8) and letting the test set be one of those groups.

With the training data, the optimal hyperparameters are computed by maximizing the log marginal likelihood. This is done by doing 100 iterations with the L-BFGS-B algorithm within the GPy package. The mean squared error and log marginal likelihood including standard error are computed as well as the coefficient of determination (R^2), AIC and BIC (see Table 5.3).

Kernel	MSE (s.e.)	LML (s.e.)	R^2	AIC	BIC
SE	$5.7 \cdot 10^{-4}$ $(7 \cdot 10^{-5})$	-20.43 (9.32)	0.943	-21,402	-21,348
Mat52	$4.5 \cdot 10^{-4}$ $(3 \cdot 10^{-5})$	-13.66 (8.98)	0.955	-22,090	-22,036
Mat32	$\mathbf{4.3 \cdot 10^{-4}}$ $\mathbf{(2 \cdot 10^{-5})}$	$\mathbf{-10.79}$ $\mathbf{(8.89)}$	**0.957**	**-22,180**	-22,126
Mat12	$4.5 \cdot 10^{-4}$ $(5 \cdot 10^{-5})$	-17.46 (8.75)	**0.957**	-22,073	-22,020
MLP	$4.4 \cdot 10^{-4}$ $(3 \cdot 10^{-5})$	-90.17 (8.58)	0.956	-22,166	**-22,149**
Lin	$6.1 \cdot 10^{-3}$ $(5 \cdot 10^{-4})$	-390.9 (34.0)	0.377	-14,625	-14,614

Table 5.3: Performance for choosing a different kernel. The mean squared error (MSE) and log marginal likelihood (LML) including the standard deviation are computed as well as the coefficient of determination (R^2), AIC and the BIC.

From this table, we can conclude that the linear model performs worst for each of the metrics (high MSE, AIC and BIC, low MLL and R^2). This was to be expected as the linear kernel can be used when the relationship between the variables is expected to be linear, which in this case is not. When focusing on the AIC and BIC values, the MLP and Matérn kernel with $\nu = \frac{3}{2}$ are suggested to be the best models as these values are the lowest for all kernels. However, the log marginal likelihood for the MLP kernel is considerably lower in comparison to the Matérn kernels. Also, considering R^2, the Matérn kernels with $\nu = \frac{1}{2}$ and $\nu = \frac{3}{2}$ perform best, though for the other kernels the values for R^2 are very similar.

Looking at the MSE, the Matérn kernel with $\nu = \frac{3}{2}$ scores best, while the squared exponential has a significant higher MSE. The results for the Matérn kernels with $\nu = \frac{1}{2}, \frac{3}{2}, \frac{5}{2}$ are very similar (note the standard error for the MLL) so either one of these kernels can be picked for further analysis. Based on the results for the MSE, MLL and the slightly lower values for AIC and BIC, the Matérn kernel with $\nu = \frac{3}{2}$ is chosen for further analysis.

5.5. Parameter and Variable Analysis

In this section, the parameters of the model (noise) and kernels such as the lengthscale will be investigated. From the previous section, the Matérn kernel with $\nu = \frac{3}{2}$ is suggested and will be used in the parameter analysis. There are several things to analyze here. First, for different values of hyperparameters, the posterior mean and twice the standard deviation are shown in Figure 5.14. Data is generated by the sine function and adding some noise to the observations (red dots). Then, for three different lengthscales, the hyperparameters are set by optimizing the MLL. For a small lengthscale ($l = 0.2$), the prediction can be seen in the figure on the left. Notice that the error bars are small when it is close to an observation and large when it is away from the data points. The prediction is more flexible in comparison to larger lengthscales, which explains the lower value for the noise σ_n. As the lengthscale increases, there is less flexibility so the noise parameter will increase.

From the figures, it can be seen that for $l = 0.2$ and $l = 1$ the predictions tend to go to the prior mean (0) very quickly. For the larger lengthscale $l = 10$, the square root of the process variance is $\sigma_f = 4.62$. So it deviates from the mean easier than for the lower lengthscales.

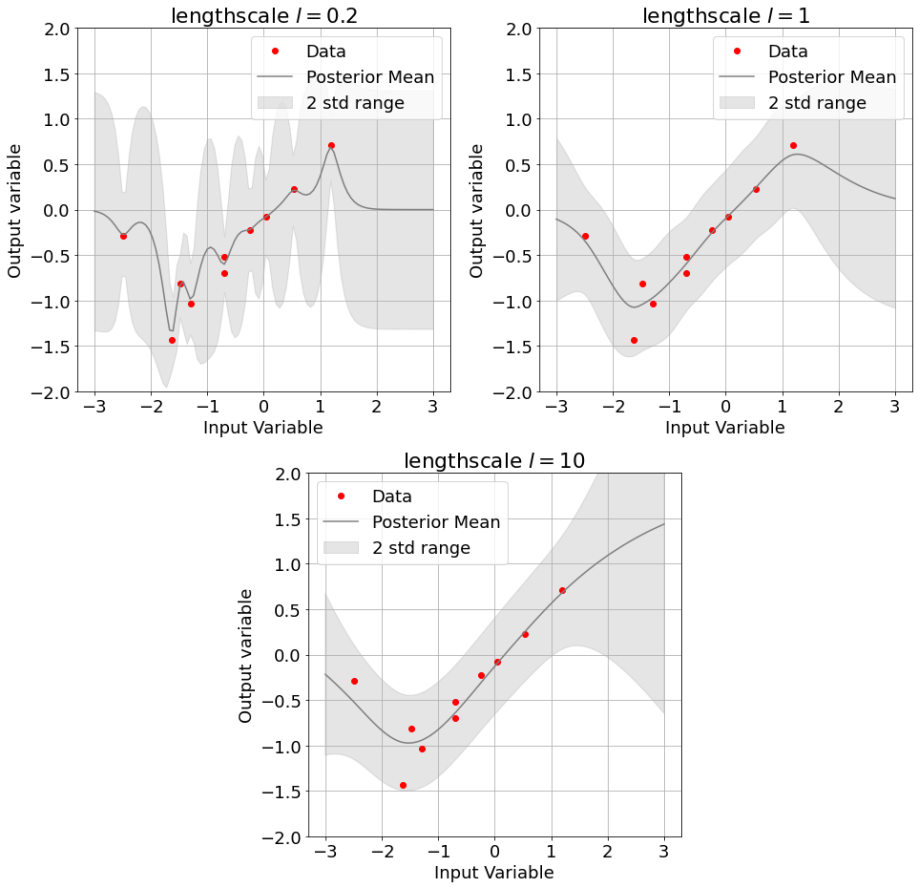


Figure 5.14: GPR using the Matérn kernel with $\nu = \frac{3}{2}$ for different lengthscales. The data is generated by using the sinus function and adding some noise. For lengthscale equal to $0.2, 1$ and 10 the hyperparameters σ_n and σ_f are set by optimizing the MLL. From left to right, the hyperparameters (σ_n, σ_f, l) are: (0.12, 0.64, 0.2), (0.22, 0.59, 1) and (0.23, 4.62, 10).

5.5.1. Log Marginal Likelihood versus the Hyperparameters

The log marginal likelihood is often used to analyze the hyperparameters (Rasmussen and Williams, 2006), because it is the distribution of **y** given the data and hyperparameters. Therefore, plotting the hyperparameters versus the log marginal likelihood gives an idea of what value it should take. To do this, all hyperparameters are fixed except for one. First, all the hyperparameters are determined by maximizing the MLL, then one of the hyperparameters is altered and a new MLL is computed. Finding the maximum MLL is done with the L-BFGS-B algorithm until convergence is met, i.e. the gradient is under a certain threshold.

```
1 kern = GPy.kern.sde_Matern32(input_dim = 7, variance = sigma_f**2,
      lengthscale = l, ARD = True)
2 gpr = GPy.models.GPRegression(X, Y, kern, normalizer = True)
3 gpr.Gaussian_noise.variance = sigma_n**2
4
5 gpr.optimize(messages = True)
```

Listing 5.1: Python code to maximize the MLL using the Matérn kernel with $\nu = \frac{3}{2}$.

In Listing 5.1, the few lines can be seen that are needed to perform the GPR with the Matérn kernel in Python. First, the kernel is created with seven input dimensions (four bands, two coordinates, one depth), the process variance and the lengthscales are assigned to initial values. In line 2, the GPR model is made with in-situ data (X, Y) where X is the input data

and Y is the output data. In line 3 the noise is added to the model with an initial value and then the MLL is maximized in line 5. In Table 5.4, the output is returned from the optimization. The runtime, number of iterations (i), -MLL (f) and the length of the gradient vector (|g|). Note that L-BFGS-B is minimizing the negative MLL, so the obtained MLL for this optimization is ≈ 14.7.

Running L-BFGS-B (Scipy implementation) Code:

runtime	i	f	|g|
12s50	0004	3.971185e+02	2.717315e+05
47s43	0015	2.484520e+02	9.843419e+01
01m41s61	0032	4.001635e+01	6.506783e+02
02m06s66	0040	1.151571e+01	9.703510e+03
03m16s78	0062	-1.103986e+01	3.764353e+03
04m10s83	0079	-1.415438e+01	2.947423e+02
04m54s91	0093	-1.473873e+01	1.562309e-08

Runtime: 04m54s91
Optimization status: Converged

Table 5.4: Output of the optimization with the L-BFGS-B algorithm in line 5 of Listings 5.1. The runtime after i number of iterations along with the negative MLL (f) and the length of the gradient vector (|g|) are shown. After 4 minutes and 54.91 seconds the optimization is converged.

The parameters that give this result are $\sigma_n = 0.22, \sigma_f = 0.67$ and

$$l = [99.96, 100.00, 75.51, 0.009, 0.40, 0.0011, 0.009],$$

where the first two lengthscales are for the longitude and latitude, the third is for the depth and the final four are the lengthscales for the blue, green, red and NIR reflectances, respectively. For each of these 9 hyperparameters, a plot has been made of the log marginal likelihood versus a single hyperparameter. In Figure 5.15, the plots can be seen for σ_f^2 and the lengthscales of the variables longitude, depth and the blue wavelength. For all other figures, the reader is referred to Appendix B. For l_{depth}, l_{blue} and σ_f^2 a clear maximum can be seen at the values that were determined before. For $l_{longitude}$ it seems that the log marginal likelihood is increasing and converging to the maximum 14.7 as $l_{longitude}$ increases.

To get a better understanding of what these figures mean, the formula in Equation 5.10 for the Matérn kernel with $\nu = \frac{3}{2}$ is used. In our model, a lengthscale parameter is included for each variable and the kernel can be written as:

$$k_{\nu=\frac{3}{2}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \left(1 + \sqrt{3}\phi(\mathbf{x}, \mathbf{x}')\right) \exp\left(-\sqrt{3}\phi(\mathbf{x}, \mathbf{x}')\right),$$
$$\phi(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{7} \frac{|x_i - x_i'|}{l_i}.$$

From this equation, it can be derived what happens when l_i tends to infinity. As $l_i \to \infty$ for a certain i, the fraction $\frac{|x_i - x_i'|}{l_i}$ goes to zero. This means that the difference in value for this variable does not influence the covariance function. So a type of variable selection can be done by analyzing these hyperparameters. The value for the MLL when $l_i \to \infty$ can be interpreted as the value for the MLL when variable i is removed from the regression. When a clear peak appears in the log marginal likelihood versus l_i plot, the variable is of some importance for the model. The height of the peak and the value for MLL determine the amount of influence a parameter has. So the figures here and in the appendix show that the variables longitude, latitude and the green wavelength can be removed from the regression to obtain a similar log

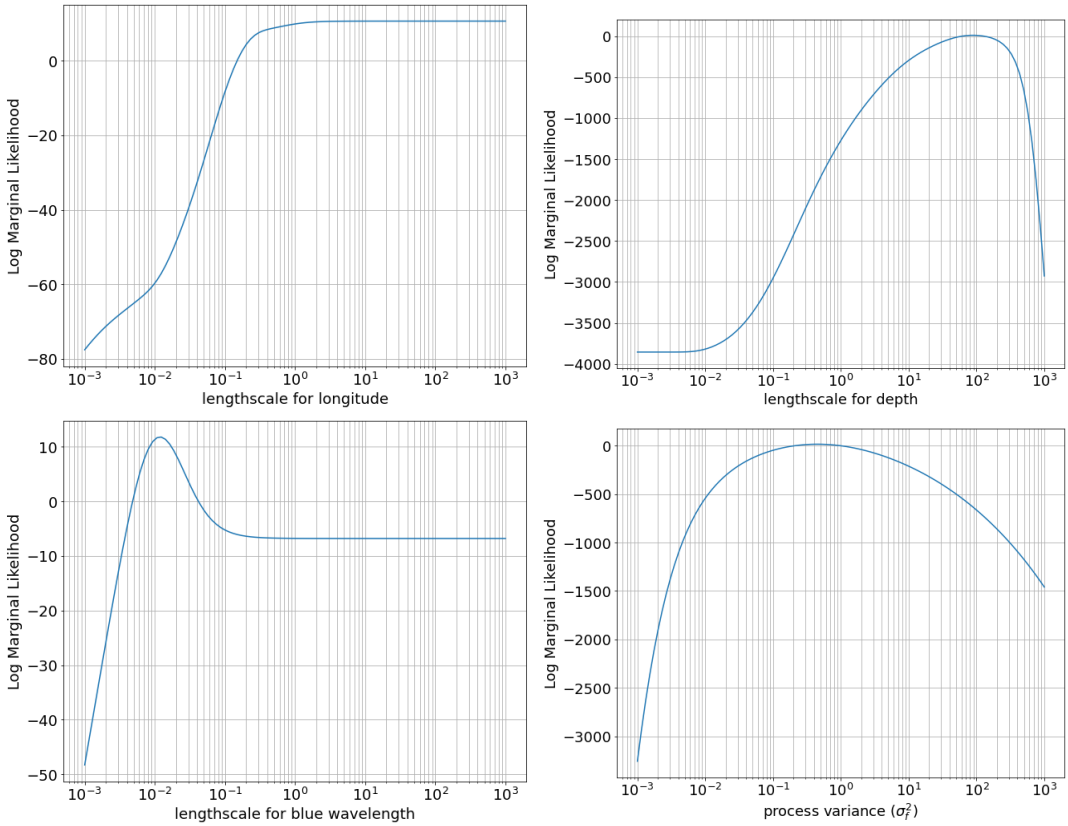


Figure 5.15: Log marginal likelihood plotted versus four hyperparameters: lengthscale for longitude (top-left), depth (top-right) and blue wavelength (bottom-left) and the process variance σ_f^2 (bottom-right).

marginal likelihood. For σ_f^2 and σ_n^2 a similar analysis can be done here as the peak shows you the value that is suggested to use.

Another way of analyzing the hyperparameters is by considering the change of log marginal likelihood as a result of changing the hyperparameters. Remember that $\log p(\mathbf{y}|X)$ is maximized to derive the hyperparameters. In Figure 5.16 (left) a contour plot is shown for different values of the lengthscale for the NIR wavelength l_{NIR} and the longitude $l_{longitude}$. A logarithmic scale is used for both axes and there is a sort of ridge containing the maximum values for the log marginal likelihood. For the plot on the right, a clear peak can be seen for l_{blue} the process variance σ_f^2.

From these plots, it is possible to derive the influence of the two hyperparameters. For example, the choice for l_{NIR} does matter for the log marginal likelihood when a high value for $l_{longitude}$ (close to 100) is set. Additionally, for l_{NIR} close to 10^{-2} the lengthscale for the longitude has little effect on the log marginal likelihood ($l_{longitude}$ in range $10^0 - 10^3$).

It is possible that there are multiple peaks in the contour plot. Whether the optimization algorithm ends up in one of those peaks depend on the initial values and the step size. It is therefore advised to run the optimization algorithm for different initial values and step sizes, although, there is no guarantee that the maximum log marginal likelihood is achieved. These plots can help to identify multiple peaks, though only two variables are altered.

5.5.2. Continuous Plots

Instead of computing the log marginal likelihood for different values of hyperparameters, the posterior mean and twice the standard deviation can be calculated for different values of a single variable. Consider a location (longitude and latitude) where an observation is made

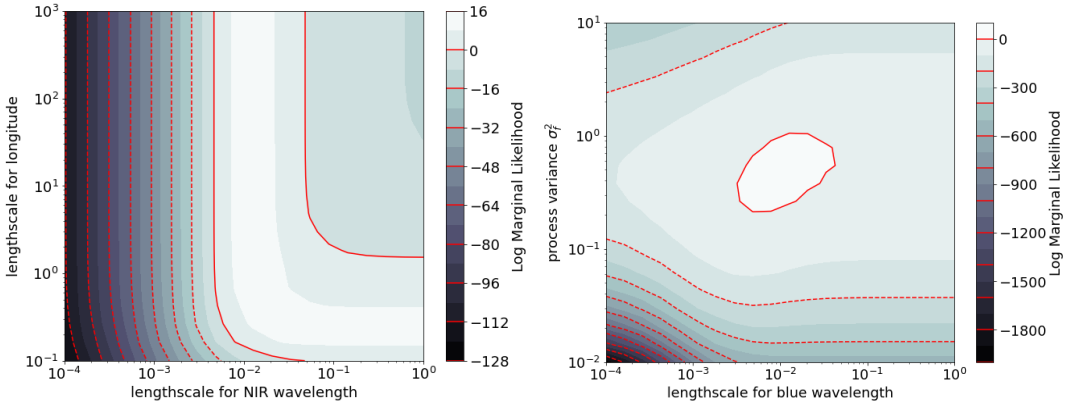


Figure 5.16: Contour plot of the log marginal likelihood for different values of the lengthscale for the NIR wavelength l_{NIR} and the longitude $l_{longitude}$ (left) and lengthscale for the blue wavelength l_{blue} and the process variance σ_f^2 (right).

by the glider from the ODYSSEA dataset. It is possible to derive the predictions at different depths for the same location and reflectances.

CHL-a	Latitude	Longitude	Depth	Blue	Green	Red	NIR
0.121	25.0915	40.3498	8.402	0.0388	0.0267	0.0228	0.0219

Table 5.5: An example of a single observation containing the chlorophyll-a concentration $[mg.m^{-3}]$, longitude and latitude in decimal degrees, depth $[m]$ and the interpolated ratios of reflectances.

First, a dataset is created containing 500 rows of the same observation (see Table 5.5 for the values). Then, the column containing the values for the depth is changed into 500 evenly spread values between 0 and 500 meters. This will be the dataset for which the predictions are made and the result for different Matérn kernels can be seen in Figure 5.17. From left to right and top to bottom the plots are made with $\nu = \frac{1}{2}, \nu = \frac{3}{2}, \nu = \frac{5}{2}$ and with the squared-exponential kernel. A similar shape can be observed in Figure 5.11, where the depth from all observations of the ODYSSEA dataset are plotted against the chlorophyll-a concentration. From the figures below, a clear DCM can be spotted around 90 meters. Furthermore, the confidence range is smallest for high values of depth and largest around the DCM. For the purpose of comparing the kernels, this plot has been made for all Matérn kernels that are discussed in this research. Interestingly, the plots are qualitatively similar but have differences in smoothness. As explained previously, the smoothness parameter ν affects the curve of the estimations, which is now shown in practice.

These kinds of plots can be made for any selection of variables, so for any location and any combination of reflectances. Moreover, instead of shifting the depth, any variable can be picked to estimate the chlorophyll-a concentrations. For example, for the same observation as before, the longitude is shifted and an estimation is made for the chlorophyll-a concentration in Figure 5.18. Here, it is distinctly visible that the error bounds are smallest around the observed value of longitude (≈ 25) and deviate when the value is further away from the observed value.

5.6. Comparison with state-of-the-art models

In this section, the GPR model will be compared with the neural networks that can be used with the SNAP software and with the polynomial regression techniques described in chapter 2.

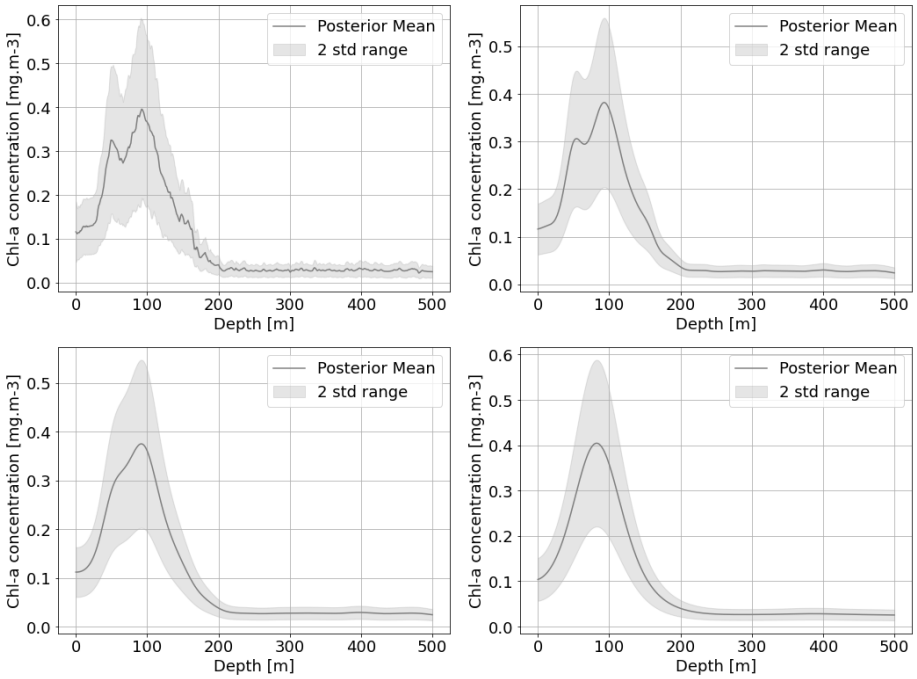


Figure 5.17: Posterior mean and twice the standard deviation for estimating the chlorophyll-a concentration at different depths. On top, the Matérn kernel with $\nu = \frac{3}{2}$ is used. The bottom row from left to right, the Matérn kernel with $\nu = \frac{1}{2}$, $\nu = \frac{5}{2}$ and the squared-exponential kernel is used.

5.6.1. C2RCC

The GPR model using the Matérn kernel with $\nu = \frac{3}{2}$ will be compared with the state of the art algorithm implemented in the SNAP tool. As mentioned in chapter 2, the C2RCC algorithm computes the absorption of phytoplankton pigments (a_{pig}) using neural networks based on the TOA reflectances. After approximately 45 minutes, the variable a_{pig} is calculated on an 1830 by 1830 grid with the same resolution that has been used before (10 meters) and saved into a netCDF file, which can be imported in Python. Then, a_{pig} is computed on the location of the in-situ data using IDW interpolation. A scatter plot of the chlorophyll-a concentration against the inherent optical property can be made, see Figure 5.19. A clear distinction between observations close to the surface and observations deep in the sea can be observed. In this example, a threshold of 5 meters is chosen, though the conclusion is similar for a threshold close to 5 meters. If the threshold is too high, the distinction is less clear and the difference in performance is smaller. Choosing the threshold too low results in a very small dataset, e.g. for 1.5 meters there are only 2 observations and no proper analysis can be done. As there are measurements done by the glider for different depths along a similar coordinate, the interpolated IOP is approximately the same, which causes these columns of data points.

Using non-linear least squares the parameters a and b can be computed. Remember that:

$$\text{CHL-a} = a \cdot a_{pig}^{b}. \tag{5.23}$$

Then, this formula can be applied to estimate the chlorophyll-a concentration with the estimated values for a and b. In Figure 5.19 the blue line is created by fitting all data (blue and red data-points): $(a, b) = (0.135, 0.113)$ and the red line is created by only fitting the red data-points, $(a, b) = (0.086, -0.069)$.

In combination with 10-Fold cross-validation, the mean-squared-error can be computed for all observations and the ones close to the surface. Every iteration, the parameters a and b are estimated, after which the estimations can be done for the test set. The resulting MSE for

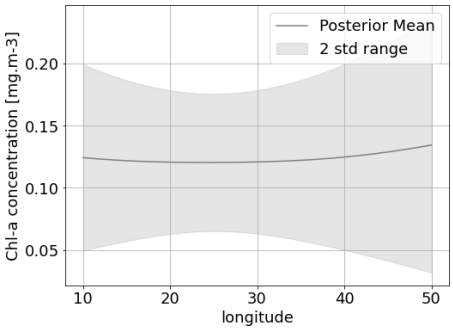


Figure 5.18: Posterior mean and twice the standard deviation for estimating the chlorophyll-a concentration for different values of longitude (decimal degrees).

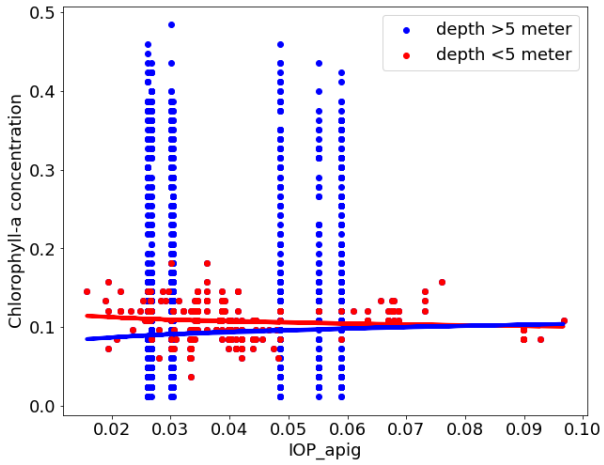


Figure 5.19: Scatter plot of the chlorophyll-a concentration against the IOP a_{pig}. Two groups are distinguished by color based on their depth and a fitted regression using Equation 5.23 is included. The blue line is created by using all observations, whilst the red line is created with only the red observations.

using all observations is $9.9 \cdot 10^{-3}$ with a standard error of $(4 \cdot 10^{-4})$. As expected, the MSE for only the data close to the surface is lower and is equal to $5.9 \cdot 10^{-4}$ with a standard error of $5 \cdot 10^{-5}$, more details in Table 5.6. The mean-squared-error for the Matérn kernel is included as a comparison and is lower for both the C2RCC method applied to the shallow observations and for all observations. Moreover, the coefficient of determination is very close to 1 for the Matérn kernel and close to zero for the C2RCC methods.

Method	MSE (s.e.)	R^2
C2RCC (all)	$9.9 \cdot 10^{-3}$ $(4 \cdot 10^{-4})$	-0.001
C2RCC (shallow)	$5.9 \cdot 10^{-4}$ $(5 \cdot 10^{-5})$	-0.003
Matérn32	$4.3 \cdot 10^{-4}$ $(2 \cdot 10^{-5})$	0.957

Table 5.6: The mean-squared-error (MSE) and coefficient of determination R^2 for the C2RCC method using all observations and only the shallow observations. The MSE and R^2 for the Matérn kernel with $\nu = \frac{3}{2}$ (using all observations) is included as comparison.

The reason for the low coefficient of determination for the C2RCC algorithm using all observations can be explained by the fact that there is no relationship between reflectances and observations taken deep in the sea. For the shallow observations only, the coefficient of determination is still low. This could be explained by having a small sample size ($n = 206$), moreover, the computation of a_{pig} is unclear. So the relationship between the reflectances and

the concentration chlorophyll-a is ambiguous. From Figure 5.20, the observed chlorophyll-a concentrations are plotted versus the predicted chlorophyll-a concentrations for all three methods. Here it can be seen that the Matérn kernel performs better than the C2RCC models.

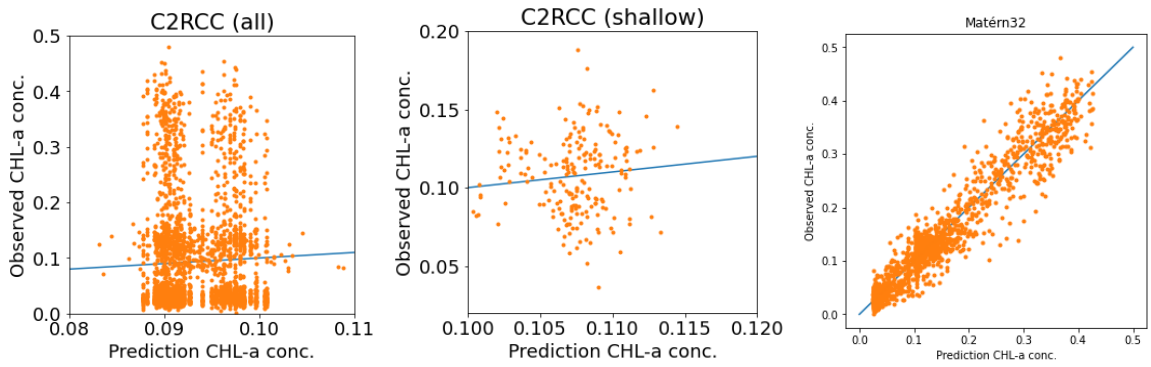


Figure 5.20: Scatter plot of the observed versus predicted chlorophyll-a concentration. The C2RCC algorithm is used with all in-situ observations (left) and only the shallow observations (middle). On the right a GPR model is applied with a Matérn kernel with $\nu = \frac{3}{2}$. The corresponding MSE and R^2 can be obtained from Table 5.6.

5.6.2. Polynomial Regression

In chapter 2, the polynomial regression techniques that are used were discussed. Here, the ratio between two reflectances is used as an explanatory variable. Therefore, the correlation between the ratio of reflectances with the chlorophyll-a concentration is analyzed first. In Figure 5.21, the Spearman's rank correlation coefficient can be seen. Again a threshold of 5 meters is chosen for the distinction between shallow observations and other observations. Every element of the correlation matrices shows the correlation between the chlorophyll-a concentration and the ratio between the row color and column color. On the diagonal, only the single reflectance values are used to compute the correlation. It is visible that the correlation increased for only the shallow observations. The ratio with the absolute highest correlation for all observations is the blue-NIR ratio ($\rho = 0.05$). Also for the shallow observations, the blue-NIR ratio has the highest correlation: $\rho = 0.38$. The MBR introduced by O'Reilly et al. (1998) was the ratio of the blue band and green band, for which $\rho = 0.27$ with the shallow observations.

For every ratio and single reflectance, a polynomial model of order 4 is created. Using 10-Fold cross-validation the MSE and R^2 are computed and compared with the previous models. This procedure is done for both the shallow observations and all observations. In Table 5.7 the metrics can be obtained for a few ratios using the shallow or all observations. The blue-NIR ratio stood out in the correlation matrices and is also plotted versus the chlorophyll-a concentration in Figure 5.22. As expected, the MSE is relatively high when all observations are used and drops when only the shallow observations are used. Note that the R^2 is very low both times in comparison with the R^2 for the C2RCC algorithm and especially in comparison with the GPR model.

The MSE for both the polynomial model and C2RCC algorithm are quite low when only using the shallow observations. This is because there is less variation in the observed chlorophyll-a concentrations. Therefore, the R^2 metric is included in this analysis. In the left figure below, it can be seen that no clear relationship is present between the variables. There is some correlation between the ratio and chlorophyll-a concentration, but not enough to estimate the concentration correctly. It is expected to have less performance when all observations are used as this model is completely based on the reflectances and these are mostly dependent on the area close to the surface. However, when only the shallow observations are consid-

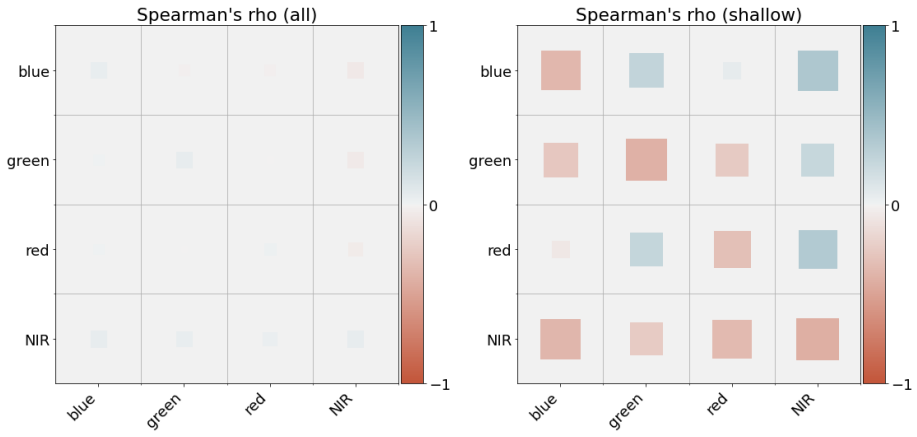


Figure 5.21: Spearman's rank correlation coefficient between the ratio of two reflectances and the chlorophyll-a concentration for all observations (left) and the shallow observations (right). The ratio of reflectances is the row divided by column, e.g. the bottom left correlation is for the ratio $R_{rs}(\lambda_{NIR})/R_{rs}(\lambda_{blue})$. On the diagonal, the single reflectance is used, so the diagonal values for all observations coincide with the correlation values in Figure 5.10.

Method	MSE (s.e.)	R^2
blue-NIR (all)	$1.1 \cdot 10^{-2}$ $(6 \cdot 10^{-4})$	-3234
blue-NIR (shallow)	$6.6 \cdot 10^{-4}$ $(5 \cdot 10^{-5})$	-222
NIR-red (shallow)	$6.7 \cdot 10^{-4}$ $(5 \cdot 10^{-5})$	-225
green (shallow)	$6.5 \cdot 10^{-4}$ $(5 \cdot 10^{-5})$	-220
Matérn32	$4.3 \cdot 10^{-4}$ $(2 \cdot 10^{-5})$	0.957

Table 5.7: The mean-squared-error (MSE) and coefficient of determination R^2 for some of the polynomial methods using all observations and only the shallow observations. The MSE and R^2 for the Matérn kernel with $\nu = \frac{3}{2}$ (using all observations) is included as comparison.

ered, there is almost no relation visible which results in the low R^2.

From the comparison with both the C2RCC algorithm and the polynomial models, it becomes clear that estimation the concentration for observations with a high value for depth is hard for these models. For shallow observations there is a higher correlation between the ratios and the chlorophyll-a concentration, however, the value for R^2 suggests that it is not a very good method. The MSE is low when only the shallow observations are considered, but this is due to the fact that there is little variation in the observed concentration. The GPR model using the Matérn kernel with $\nu = \frac{3}{2}$ performs better in terms of MSE and R^2 when all in-situ observations are used.

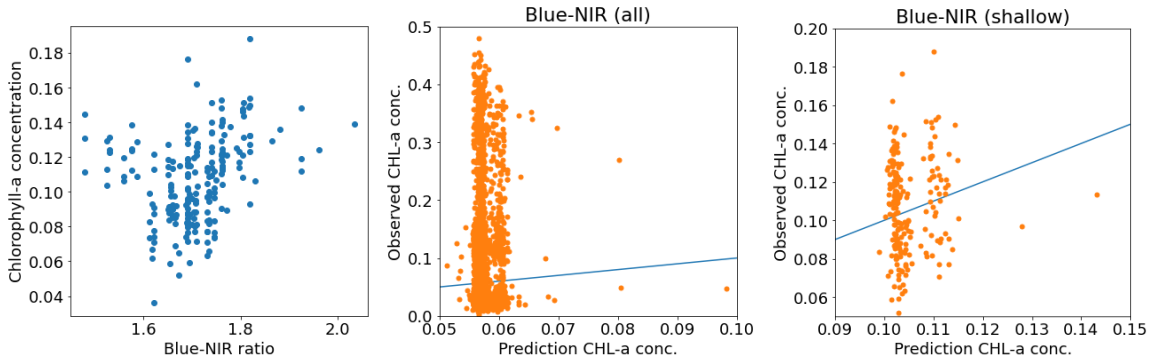


Figure 5.22: Scatter plot of the chlorophyll-a concentration versus the blue-NIR ratio (left) and the observed versus predicted chlorophyll-a concentration (middle and right). All observations are used in the middle plot and only the shallow observations are used in the plot on the right. The corresponding MSE and R^2 can be obtained from Table 5.7.

6

Approximation Methods

In chapter 4 we have determined that GPR scales cubically with the number of in-situ observations. In this chapter, we will discuss a few approximation methods to speed up this process. First, the Cholesky decomposition is explained as this is an exact method that does speed up the inversion of the matrix and is used in standard packages to invert a matrix. Thereafter, the approximation methods will be explained and tested for some toy problems and our chlorophyll-a problem.

6.1. Cholesky Decomposition

The Cholesky-decomposition of a real, symmetric, positive definite matrix A is of the form: $A = LL^T$ where L is a lower triangular matrix. Furthermore, L has only strictly positive values on the diagonal and as a result, can be inverted. Because of the particular shape of L, the inverse L^{-1} can be computed relatively quickly. The inverse for A is equal to:

$$A^{-1} = (LL^T)^{-1} = (L^T)^{-1}L^{-1} \tag{6.1}$$

So instead of directly inverting matrix A, L needs to be calculated after which the inverse L^{-1} needs to be computed. The computational cost of the Cholesky factorization equals $\frac{1}{3}n^3 + \mathcal{O}(n^2)$, then computing the inverse costs $\frac{1}{3}n^3 + \mathcal{O}(n)$ flops. So in total, the computational cost to compute L^{-1} is equal to $\frac{2}{3}n^3 + \mathcal{O}(n^2)$. An ordinary matrix-matrix multiplication of two $n \times n$ sized matrices costs $2n^3 - n^2$ flops. As $(L^T)^{-1} = (L^{-1})^T$ and we know that A^{-1} is symmetric, the required number of flops reduces. The total number of flops required to compute the matrix-matrix multiplication is $\frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n$. To compute A^{-1}, a total of $n^3 + \mathcal{O}(n^2)$ flops is required.

In modern packages for Python, NumPy (Harris et al., 2020), the default algorithm to invert a matrix is based on the LU decomposition. The computational cost of the LU-factorization is $\frac{2}{3}n^3 + \mathcal{O}(n^2)$. Then, again, the inverse of L and U need to be computed which costs twice the computational cost to compute L^{-1} which is $\frac{4}{3}n^3 + \mathcal{O}(n^2)$. The required cost for the matrix-matrix multiplication is similar as before as we are dealing with an upper and lower triangular matrix resulting in a symmetric matrix. The total amount of flops required to compute A^{-1} is: $2\frac{1}{3}n^3 + \mathcal{O}(n^2)$. For various values of n, the average computational time is recorded and shown in Figure 6.1. We can see that indeed the time needed to compute the inverse with a Cholesky decomposition is less than with a LU decomposition. On average, Cholesky is twice as fast, however for the two largest matrices, Cholesky is three times faster ($n = 2560$) and almost five times faster ($n = 5120$).

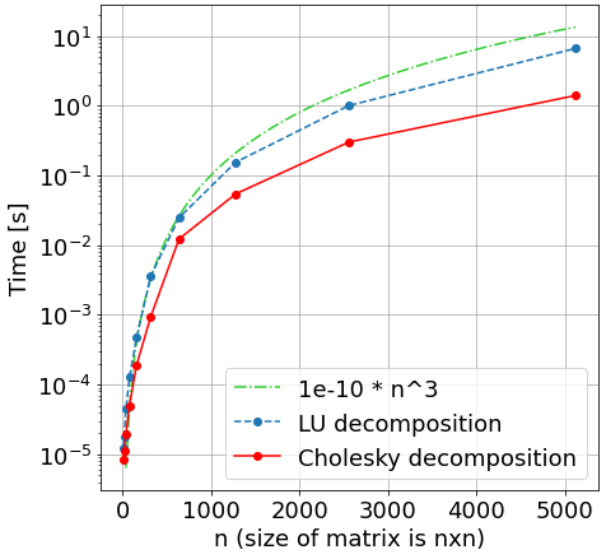


Figure 6.1: Computation time needed to compute the inverse of a matrix of size $n \times n$ using a LU decomposition (blue) and a Cholesky decomposition (red). In green the function $y = 10^{-10}n^3$ is plotted. With a Cholesky decomposition, the time needed to invert a matrix is less than with a LU decomposition.

6.2. Expectation Propagation

Rasmussen and Williams (2006) showed that expectation propagation (EP) can be used to approximate analytically intractable integrals for Gaussian process classification (GPC). This idea of approximating an integral with EP leads to the idea to use EP to approximate the posterior distribution. In this section, the steps taken in the EP algorithm will be explained.

Approximating a probability distribution can be done using a technique called expectation propagation (EP) (Minka, 2001). EP uses an iterative scheme to minimize the Kullback-Leibler divergence $D_{KL}(p||q) = \int p(x) \log \frac{p(x)}{q(x)} dx$ where $q(x)$ approximates the distribution $p(x)$. Alternatively, one can minimize $D_{KL}(q||p)$ which is called variational inference (Rasmussen and Williams, 2006).

When the distribution for $q(x)$ is assumed to be from the exponential family, we can write:

$$q(x;\eta) = c(\eta)h(x) \exp\left(\sum_{i=1}^{k} \pi_i(\eta)\tau_i(x)\right), \tag{6.2}$$

where η are called the natural parameters of the distribution and the function $c(\eta)$ is the function such that the integral over all possible values of x is equal to 1 (required to be a probability density function). Furthermore, $\pi_i(\eta)$ and $\tau_i(x)$ are measurable functions for $i = 1, \ldots, k$. By following the steps from Bishop (2006), the KL divergence is minimized when the expected sufficient statistics are matched. When $q(x)$ is chosen to be Gaussian, the minimization of the KL divergence is optimal when the mean of $q(x)$ is equal to the mean of the distribution of $p(x)$ and the variance (covariance matrix) is equal to the variance (covariance matrix) of $p(x)$. This is called *moment matching* (Rasmussen and Williams, 2006).

Using Bayes rule, the posterior $p(\theta|\mathcal{D})$ for latent parameters θ conditioned on the observed data $\mathcal{D}$, can be written as the product of the likelihood and the prior distribution divided by the marginal likelihood (a normalization constant).

$$p(\theta|\mathcal{D}) = \frac{p(\theta)p(\mathcal{D}|\theta)}{\int p(\theta)p(\mathcal{D}|\theta)d\theta}. \tag{6.3}$$

By following the steps from Bishop (2006), we assume that the likelihood can be factorized into n independent observations.

$$p(\theta|\mathcal{D}) \propto p(\theta)\Pi_{i=1}^{n} p(y_i|\theta) = p(\theta)\Pi_{i=1}^{n} p_i(\theta). \tag{6.4}$$

Here, we used $p(y_i|\theta) = p_i(\theta)$ for convenience in writing the rest of the procedure. Now, the approximation used in EP is

$$q(\theta) = \frac{1}{Z} p(\theta)\Pi_{i=1}^{n} q_i(\theta), \tag{6.5}$$

where Z is the approximation for the marginal likelihood and $q_i(\theta)$ the approximation for $p_i(\theta)$ (Rasmussen and Williams, 2006; Bishop, 2006). Next, the approximations for the likelihood are initialized and sequentially updated by removing and storing individual factors $q_i(\theta)$. In more detail, if we want to update the factor $q_j(\theta)$, firstly the factor is removed from the product such that we end up with the so-called *cavity distribution*. The product can either be obtained by multiplying each individual term except $q_j(\theta)$ or by using the approximated likelihood and dividing by $q_j(\theta)$.

$$q_{-j}(\theta) \propto p(\theta)\Pi_{i=1, i\neq j}^{n} q_i(\theta). \tag{6.6}$$

Subsequently, the factor is updated by minimizing the Kullback-Leibler divergence which is now a tractable problem. While directly minimizing $D_{KL}(p(\theta|\mathcal{D}), q(\theta))$ leads to an untractable problem as the unknown posterior needs to be calculated. Instead, using EP, each individual factor is approximated to find an approximation for the likelihood.

$$q_j^{new}(\theta) = \underset{q_j(\theta)}{\text{argmin}}\, D_{KL}\left(p_j(\theta)q_{-j}(\theta)\middle\|q_j(\theta)q_{-j}(\theta)\right). \tag{6.7}$$

This minimization can be simplified by choosing convenient distributions for our approximations. As a result of the above, a distribution from the exponential family (e.g. the Gaussian distribution) can be assigned to the approximations such that the minimization is reduced to moment matching. The described procedure is done for all factors $j = 1, \ldots, n$ which is called one EP iteration.

More information about moment matching (derivations of moments) is given by Rasmussen and Williams (2006) and Bishop (2006) and are not included as they are quite lengthy and outside the scope of this study. Furthermore, pseudocodes are given for updating the hyperparameters and predictions. We can then implement these algorithms in Python and compare them with the exact solution (i.e. update hyperparameters using gradient ascent and inverting the covariance matrix), practised on a toy problem. Note that in these algorithms, the computational complexity is still dominated by an inversion of a matrix and is $\mathcal{O}(n^3)$.

Let the training data $\mathcal{D} = (X, \mathbf{y}) = \{x_i, y_i\}_{i=1}^{n}$ where $x_i = \frac{(i-1)}{2}$ and $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, k(X, X) + 0.3I_n)$ and $n = 201$. For the covariance function k, a squared-exponential kernel is used with process variance equal to $\sigma_f^2 = 1$ and lengthscale $l = 8$ (for more information about these hyperparameters, see Section 5.4.1). In Figure 6.2 the observations can be seen as well as the posterior mean and twice the standard deviation at each input value x for both the exact and EP method. For this particular problem, the EP algorithm seems to be able to approximate the exact solution quite well. For larger n, the approximations get worse and sometimes no convergence is achieved due to numerical instabilities. Most importantly, the computation time for this toy problem for the exact method is approximately 0.34 seconds for optimizing the hyperparameters and 0.06 seconds for computing the posterior mean and variance. The computation time for the EP method is 3.72 seconds for updating $q_i(\theta)$ for each i until convergence and 0.03 seconds for computing the predictions for EP.

So, updating the hyperparameters using gradient descent and inverting the matrix exactly, is roughly 10 times faster than using the EP approximations. Part of the difference can be

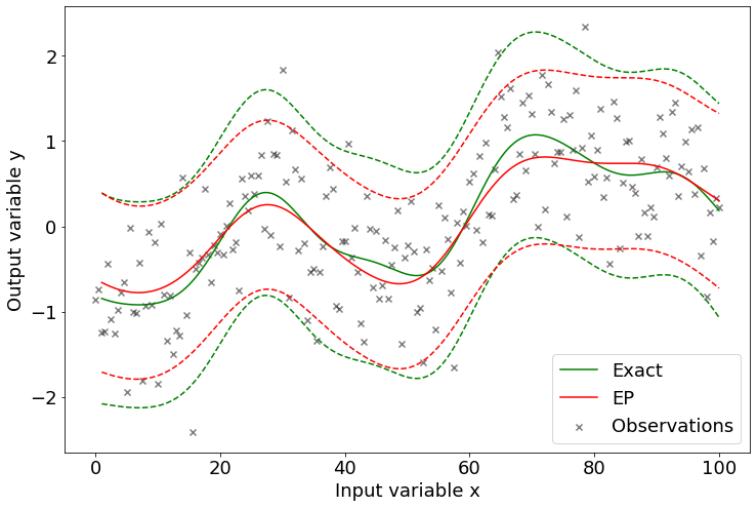


Figure 6.2: Posterior mean and twice the standard deviation at each input value x for the exact method (green) and EP method (red). Using 201 observations (black) the EP method seems to be able to approximate the exact solution, however, the time needed for the EP method is roughly 10 times more than for the exact method.

explained by the fact that the Python code for EP approximations is manually written, whereas for the exact method the GPy package has been used (GPy, since 2012). Initially, the EP method was introduced for GPC to approximate analytically intractable integrals. With GPR, no approximation is required as the posterior can be written as a normal distribution, though it is still desired to approximate the matrix inversion. For these reasons, the EP method will not be able to speed up the GPR.

6.3. Sparse Gaussian Processes

Another option is to use a so-called sparse Gaussian Process. The idea is to use only m observations instead of all n observations ($m < n$) such that the covariance matrix reduces in size. To do this with minimum loss of performance, i.e. the log marginal likelihood is close to the original log marginal likelihood, the observations used need to be optimized. Simply said, the m observations need to be a good representation of all n observations.

There are a few options in choosing the m observations. The first simple method is to randomly select m observations of the total n observations. Secondly, the selection can also be done by some greedy algorithm. For example, an observation is added to the collection of m observations when it minimizes or maximizes some metric, such as the mean squared error or the log marginal likelihood (Quinonero-Candela et al., 2007; Rasmussen and Williams, 2006). Another option is a variational method that optimizes the m inducing input variables, so they are not a subset from the n observations (Titsias, 2009). For all these methods, the n observations are summarized by m observations and the resulting matrix that need to be inverted is now of size $m \times m$ which reduces the time complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(m^3)$.

6.3.1. Variational Learning

Suppose we have m inducing variables $\mathbf{f}_m$ at inputs X_m, these variables should ultimately summarize the data $\mathbf{f}$ at input X. This can be explained more precisely with an one-dimensional input and output problem. The m inducing variables should represent the n observations, so if the relation between the input and output variable is almost perfectly linear, only a two inducing variables are needed as this can explain the complete relationship. For more complex relationships, more inducing variables are needed. From the experiments below, e.g. Figure

6.4, the inducing points are often located around the peaks and valleys of the curve. One can imagine that more inducing points are needed to represent the data when multiple input variables are used.

To establish the relationship between the two variables $\mathbf{f}_m$ and $\mathbf{f}$, a multivariate normal distribution is defined to be the prior:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_m \end{bmatrix} \sim \text{Gau}\left(\begin{bmatrix} m(X) \\ m(X_m) \end{bmatrix}, \begin{bmatrix} \Sigma_{ff} & \Sigma_{ff_m} \\ \Sigma_{f_m f} & \Sigma_{f_m f_m} \end{bmatrix} \right). \tag{6.8}$$

For convenience we set $m(X)$ and $m(X_m)$ equal to zero, and the matrices are defined in a similar way as in Equation 4.2. So $\Sigma_{ff_m} \in \mathbb{R}^{n \times m}$ is the covariance matrix where the element on row i and column j is equal to $k(\mathbf{x}_i, \mathbf{x}_j)$, with k a covariance function and $\mathbf{x}_i$ the i-th input from X and $\mathbf{x}_j$ the j-th input from X_m. Using this prior, the posterior $p(\mathbf{f}|\mathbf{f}_m)$ can be computed identically as before.

$$\mathbf{f}|\mathbf{f}_m \sim \mathcal{N}(\Sigma_{ff_m}\Sigma_{f_m f_m}^{-1}\mathbf{f}_m, \Sigma_{ff} - \Sigma_{ff_m}\Sigma_{f_m f_m}^{-1}\Sigma_{f_m f}) \tag{6.9}$$

Now, in variational learning, the so-called augmented posterior $p(\mathbf{f}, \mathbf{f}_m|\mathbf{y})$ will be approximated by the augmented variational posterior $q(\mathbf{f}, \mathbf{f}_m)$ (Titsias, 2009). Then it is proposed to factorize the augmented variational posterior into: $q(\mathbf{f}, \mathbf{f}_m) = p(\mathbf{f}|\mathbf{f}_m)q(\mathbf{f}_m)$ with the advantage that $p(\mathbf{f}|\mathbf{f}_m)$ is known. The distribution $q(\mathbf{f}_m)$ is defined to be multivariate normal with mean vector $\boldsymbol{\mu}$ and covariance matrix A. The disadvantage of this approach is that there are $m + \frac{1}{2}m(m+1)$ extra parameters (A is symmetric) that need to be optimized.

The optimization is done by using the log marginal likelihood, where the conditional on X will be dropped in notation as everything is conditioned on X, so this can be rewritten into:

$$\begin{aligned}
\log p(\mathbf{y}) &= \log \int\int p(\mathbf{y}, \mathbf{f}, \mathbf{f}_m) d\mathbf{f} d\mathbf{f}_m && (6.10)\\
&= \log \int\int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}, \mathbf{f}_m) d\mathbf{f} d\mathbf{f}_m && (6.11)\\
&= \log \int\int p(\mathbf{y}|\mathbf{f}) p(\mathbf{f}, \mathbf{f}_m) \frac{q(\mathbf{f}, \mathbf{f}_m)}{q(\mathbf{f}, \mathbf{f}_m)} d\mathbf{f} d\mathbf{f}_m && (6.12)\\
&= \log \mathbb{E}_{q(\mathbf{f}, \mathbf{f}_m)} \left[p(\mathbf{y}|\mathbf{f}) \frac{p(\mathbf{f}, \mathbf{f}_m)}{q(\mathbf{f}, \mathbf{f}_m)} \right] && (6.13)\\
&\geq \mathbb{E}_{q(\mathbf{f}, \mathbf{f}_m)} \log \left[p(\mathbf{y}|\mathbf{f}) \frac{p(\mathbf{f}, \mathbf{f}_m)}{q(\mathbf{f}, \mathbf{f}_m)} \right] && (6.14)\\
&= \int\int \log(p(\mathbf{y}|\mathbf{f})) q(\mathbf{f}, \mathbf{f}_m) d\mathbf{f} d\mathbf{f}_m - \int\int \log \left(\frac{q(\mathbf{f}, \mathbf{f}_m)}{p(\mathbf{f}, \mathbf{f}_m)} \right) q(\mathbf{f}, \mathbf{f}_m) d\mathbf{f} d\mathbf{f}_m && (6.15)\\
&= \int \log(p(\mathbf{y}|\mathbf{f})) q(\mathbf{f}) d\mathbf{f} - D_{KL}(q(\mathbf{f}, \mathbf{f}_m)||p(\mathbf{f}, \mathbf{f}_m)) && (6.16)
\end{aligned}$$

Where we used Jensen's inequality for the concave function $\phi(x) = \log(x)$ in Equation 6.14 and the definition for the Kullback-Leibner divergence in the last step. The Kullback-Leibner divergence can be rewritten into (Hensman et al., 2015):

$$\begin{aligned}
D_{KL}(q(\mathbf{f}, \mathbf{f}_m)||p(\mathbf{f}, \mathbf{f}_m)) &= D_{KL}(q(\mathbf{f}_m)||p(\mathbf{f}_m)) && (6.17)\\
&= \frac{1}{2} \left[\log \left(\frac{\det \Sigma_{f_m f_m}}{\det A} \right) - m + tr(\Sigma_{f_m f_m}^{-1} A) + \mu^T \Sigma_{f_m f_m}^{-1} \mu \right] && (6.18)
\end{aligned}$$

The first term: $\int \log(p(\mathbf{y}|\mathbf{f}))q(\mathbf{f})d\mathbf{f}$ can be approximated by an analytical expression by e.g. Gauss-Hermite quadrature (Hensman et al., 2015). The computational cost of computing the

upper bound of the log marginal likelihood is dominated by the inversion of a $m \times m$ sized matrix ($\mathcal{O}(m^3)$). Hensman et al. (2015) suggests to use the augmented variational posterior to make predictions at the test points X_* which has $\mathcal{O}(m^2)$ complexity:

$$p(\mathbf{f}_*|\mathbf{y}) \approx \int p(\mathbf{f}_*|\mathbf{f}_m)q(\mathbf{f}_m)d\mathbf{f}_m. \tag{6.19}$$

This approach has been implemented into the GPy package (GPy, since 2012) and is demonstrated using a simple example. For a one-dimensional problem (one input variable and one output variable) we use $n = 1001$ equidistant input values ranging from 0 to 100, where the output $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, k(X, X)+0.3I_n)$. For the covariance function, the squared exponential kernel is used with lengthscale $l = 10$ and process variance $\sigma_f^2 = 1$. This is solved exactly using the $n \times n$ sized matrix and it is solved for different values of the inducing inputs m. In Figure 6.3 one can see the log (marginal) likelihood and computation time for different values of m and for the the complete set of inputs $n = 1001$. The computation time starts relatively high for $m = 10$ and then drops to around half a second for $m = 30, \ldots, 90$. The reason for this is that for only 10 inducing points, the gradient ascent algorithm needs to do more iterations to find the optimal μ and A. For $m = 30, \ldots, 90$ the initial μ and A are already sufficient to predict well (the log marginal likelihood is high), so only a few iterations need to be done for convergence. For $m = 20, \ldots, 90$ the log marginal likelihood is close to the maximum log marginal likelihood which is achieved using all input values. So in this example, the predictions can be done by using only 30 inducing points, achieving a similar log-likelihood and is approximately 25 times faster.

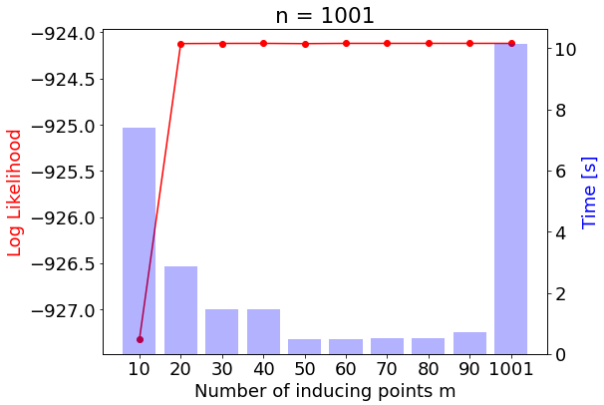


Figure 6.3: The log-likelihood and computation time for the number of inducing points m. Also for the exact method, using all input variables, the computation time and log-likelihood is computed $m = 1001$. The computation time is low for m ranging from 20 to 90 in comparison with the exact method. The log-likelihood for these values of m is similar to the exact method so this suggests that predictions can be made with less computation time.

Plotting the posterior mean and twice the posterior standard deviation, the difference between the predictions can be observed (see Figure 6.4). The prediction using 20 inducing points and the exact prediction has been plotted and no difference can be observed visually. The maximum difference in posterior mean and variance is 0.0039 and 0.0017 respectively.

This approach of introducing inducing parameters is then applied to the chlorophyll-a problem where 2866 in-situ observations are used as training data. Now, there are 7 input variables so probably more inducing parameters are needed than for one input variable. After a few test runs, it seems that no convergence is achieved after 1000 iterations for any choice of m. To speed up this process without a significant loss of log marginal likelihood, a maximum of 100

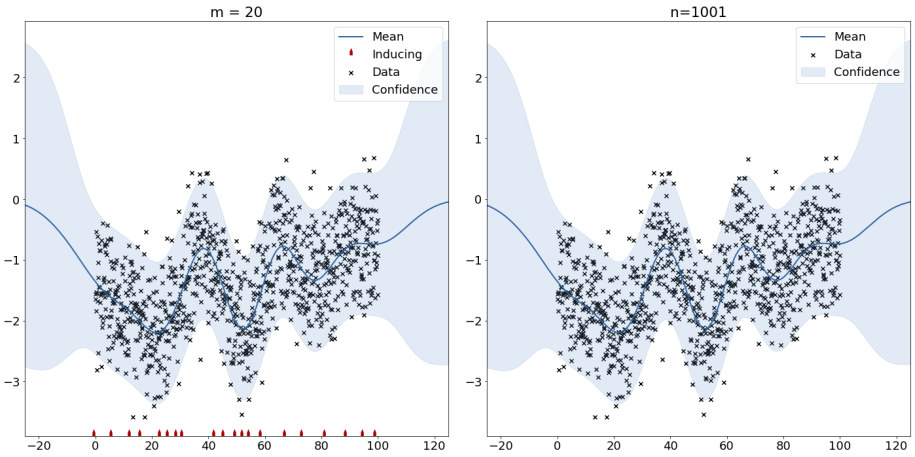


Figure 6.4: Prediction using 20 inducing points (left) and the exact posterior (right).

iterations are done. Similarly, for the exact method, only 20 iterations are done. The resulting computation time and log (marginal) likelihood can be seen in Figure 6.5. It can be seen that the computation time and log-likelihood increase as m increases. This is in contrast with Figure 6.3, where for small m the computation time was relatively high. This difference is caused by fixing the number of iterations. Another change in results is the visual difference in log-likelihood for the exact model and the sparse GPR models (SGPR). Finally, the ratio between the computation time of the exact model and the SGPR models is smaller than in the simple problem before.

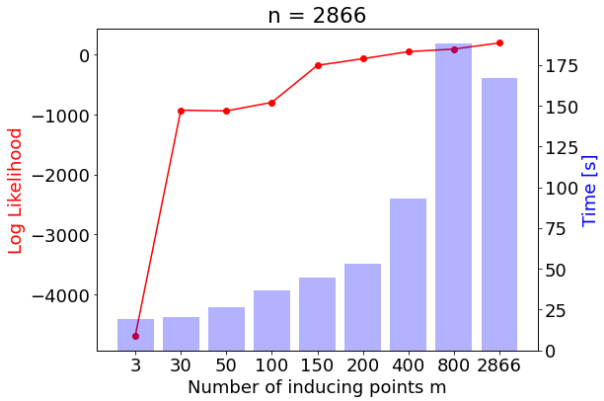


Figure 6.5: The log-likelihood and computation time for the number of inducing points m. Also for the exact method, using all input variables, the computation time and log-likelihood is computed $m = 2866$. As m increases, the computation time and log-likelihood increase. This difference in comparison with Figure 6.3 is caused by fixing the number of iterations.

For $m = 200$ the posterior mean (left) and standard deviation (right) are shown in Figure 6.6 (top) as well as for the exact method (bottom). The scale for the chlorophyll-a concentration is the same for the posterior means and the standard deviation, so the differences can be seen visually. The major distinction between the plots is in the bottom right of the map where high concentrations of chlorophyll-a are estimated. Using the exact method, the concentrations are estimated higher than using 200 inducing points and from the posterior standard deviation, it can be seen that the SGPR model is more certain about this estimation.

To conclude, it is shown that variational learning is a good approximation technique for a Gaussian process regression with one input variable. For a similar log-likelihood, only a

twenty-fifth of the computation time is needed. For more complex Gaussian process regression models with seven input variables, variational learning is less powerful as more inducing points are needed to get a good representation of your n observations. In our problem, there is a significant difference in log-likelihood (and the predictions) and the computation time is reduced by a third.

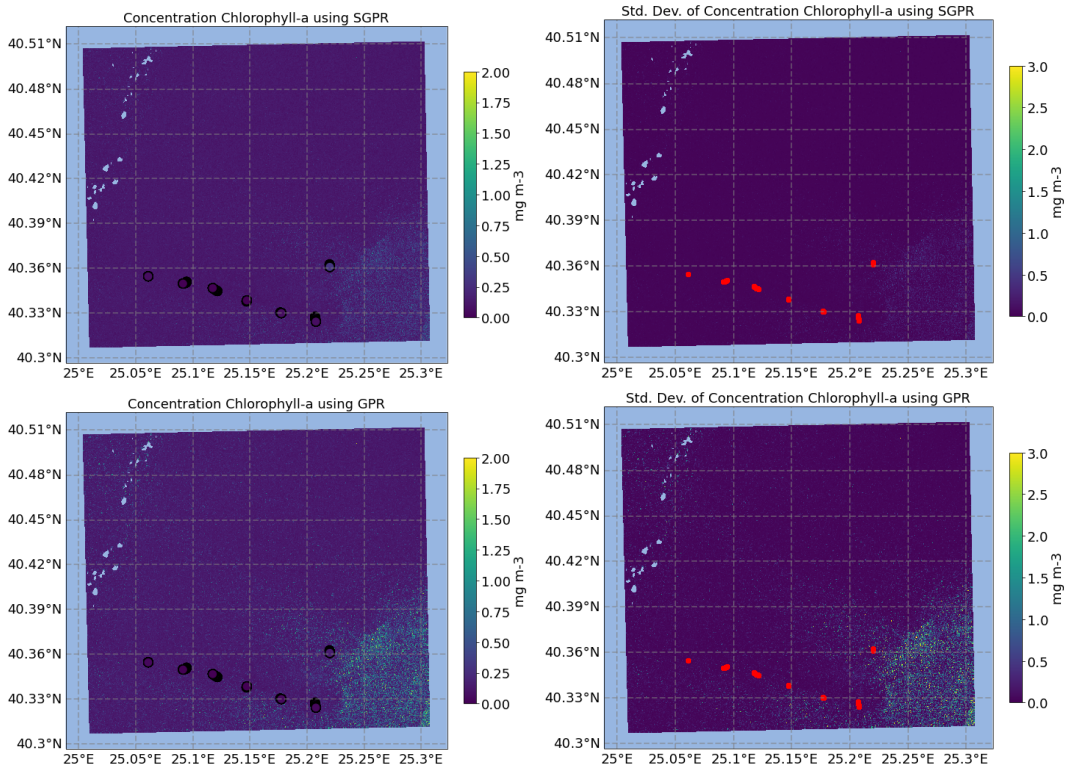


Figure 6.6: The posterior mean (left) and standard deviation (right) for $m = 200$ inducing points (top) and the exact model (bottom). The computation time and log-likelihood are 53 seconds and -69 (SGPR) and 167 seconds and 194 (GPR).

6.4. Singular Value Decomposition

For image compression, a technique called singular value decomposition (SVD) is frequently used to reduce the number of data that needs to be stored while the important information is still saved (Rufai et al., 2014). An image can be represented by a matrix, where each element contains the intensity value of the corresponding pixel.

Let the matrix $A \in \mathbb{R}^{m \times n}$ be of rank r. Then there exists an orthogonal matrix $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ and a diagonal matrix $D \in \mathbb{R}^{m \times n}$ which contains the singular values sorted in descending order such that:

$$A = UDV^T.$$

The singular values $\sigma_i > 0, i = 1, \ldots, r$ of matrix A are defined by $A\mathbf{v}_i = \sigma_i \mathbf{u}_i$ and $A^T\mathbf{u}_i = \sigma_i \mathbf{v}_i$ where $\mathbf{v}_i$ and $\mathbf{u}_i$ are the columns of V and U (they are called the right and left singular vectors, respectively). The relation between the eigenvalues and singular values can be found by computing AA^T and A^TA:

$$AA^T = UDV^TVD^TU^T = UDD^TU^T,$$
$$A^TA = VD^TU^TUDV^T = VD^TDV^T.$$

So, the singular values of A are the square root of the eigenvalues of AA^T and A^TA and in addition, the eigenvectors of AA^T and A^TA are the left and right singular vectors of A. To introduce the Eckart-Young theorem, first the Frobenius norm is defined by:

$$||A||_F := \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}^2|} = \sqrt{tr(A^TA)} = \sqrt{\sum_{i=1}^{r}\sigma_i^2}.$$

Now, the Eckart-Young theorem tells us that for a matrix $A \in \mathbb{R}^{m\times n}$ of rank r, the matrix $B \in \mathbb{R}^{m\times n}$ of rank $k < r$ that minimizes $||A - B||_F$ is $B = U\hat{D}V^T$ where the diagonal of $\hat{D}$ contains the largest k singular values of A.

In our problem, we have a symmetric matrix $\Sigma_n = \Sigma_{ff} + \sigma_{noise}^2 I_n$ which we would like to inverse. Using SVD with a symmetric matrix the inverse can easily be computed by:

$$A^{-1} = (UDV^T)^{-1} = (V^T)^{-1}D^{-1}U^{-1} = VD^{-1}U^T.$$

Where in the last step the property $U^{-1} = U^T$ for orthogonal matrices is used. So only a diagonal matrix D needs to be inverted. Furthermore, since Σ_n is symmetric, the singular values are the absolute values of the eigenvalues of Σ_n and $U = V$. As an example, the covariance matrix created with the squared exponential kernel is used as matrix A (rank $n = 1000$) and for ranks $k = 1, 2, \dots 8$ the approximations B are visualized in Figure 6.7, here $\sigma_n = 1$ is used. As the rank increases, the approximations begin to look like the original matrix in the bottom-right corner.

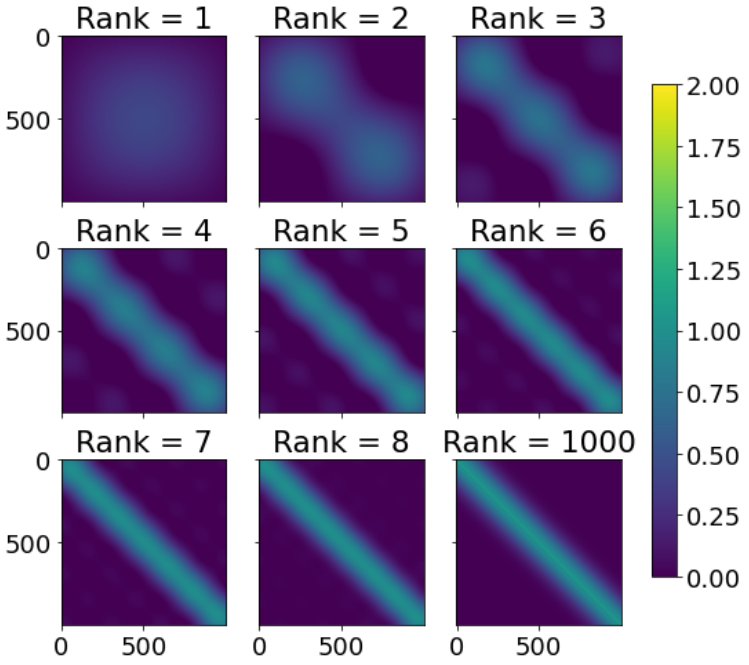


Figure 6.7: Approximations for matrix Σ_n (bottom-right) for different ranks. As the rank increases, more singular values are incorporated and the approximation gets better.

As a measure of error, the Frobenius norm is used again: $\frac{||A-B||_F}{||A||_F}$. The advantage of the approximations is that less memory is needed and also less computation time. Though, for each iteration in updating the hyperparameters, U and D need to be determined which is done by the Lanczos method. The computational cost of the Lanczos method is $\mathcal{O}(dn^2)$ flops, where d is the average number of nonzero elements in the rows of the original matrix

A. When rank k is known, U does not need to be calculated completely as the matrix D will contain multiple rows and columns of zeroes so only the first k columns of U are needed. So $n \cdot k$ numbers are stored for the matrix U and k numbers are stored for D which brings the total storage to $(n+1)k$ numbers. In Table 6.1 the error, rank, numbers to store, compression factor and the computation time can be obtained. The computation time is the average time needed to compute the inverse 1000 times. The error decreases as k increases (this was observed visually in Figure 6.7) and so is the compression factor: $\frac{n^2}{(n+1)k}$. The computation time is roughly the same for the approximations and almost six times lower than for computing the exact inverse. The standard errors are approximately $1.1 \cdot 10^{-4}$ and $5.7 \times \cdot 10^{-4}$ for the approximations and exact method, respectively. The computation time for the approximations is close to each other because the number of singular values is close to each other.

Rank	Error	Storage	Factor	Time [$\times 10^{-2}$ s]
1	0.81	1,001	999	1.60
4	0.31	4,004	250	1.63
8	0.08	8,008	125	1.62
1000	0	1,000,000	1	9.40

Table 6.1: The error, numbers to store, compression factor and computation time for inverting A using different ranks and the full matrix (rank is 1000).

The choice for rank k is yet to be established and there is no single rule to do this. We want k to be as small as possible as this requires a few numbers to store and low computation time. However, the approximation needs to be 'good enough' so a large k is perhaps desired. One desires to take the k singular values (and singular vectors) that contain as much information of the structure of A. This is done by taking the largest k singular values, thus a clear distinction between the high singular values and low singular values is desired. The 1000 singular values of Σ_n are shown in Figure 6.8. Note that the scale on the y-axis is logarithmic. Due to the choice for the variance of the noise ($\sigma_n^2 = 1$), many singular values are close to 1. Furthermore, the important observation is the clear distinction in high and low singular values. The 11 highest singular values are higher than 10 while the others are below this threshold. This suggests that k can be 11 or maybe even lower depending on the specific requirements.

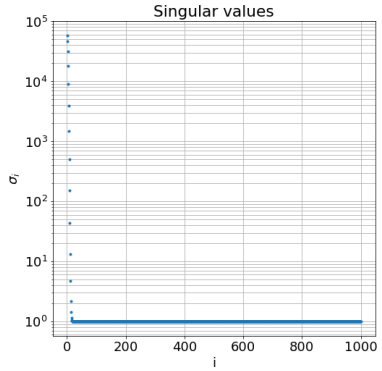


Figure 6.8: Singular values σ_i sorted in descending order, for $i = 1, 2, \ldots, 1000$ of matrix $\Sigma_{n=1000}$. Note that the scale on the $y - axis$ is logarithmic.

So far, we have only looked at the error made by approximating A by B and not the inverse A^{-1} approximated by B^{-1}. With the same set-up as before (i.e. $n = 1000$ and $\sigma_n = 1$) the inverse is computed for different ranks. In Figure 6.9 the inverse of the approximations can be seen for different ranks. Note that the chosen ranks are different from before as the

approximations for the inverse are slowly converging to the original inverse. From the inverse approximation for rank 50, the structure of the original inverse is visible, whereas this is not visible for ranks 10 and lower. Again, the measure for error can be computed using the Frobenius norm: $\frac{||A^{-1}-B^{-1}||_F}{||A^{-1}||_F}$ which is 0.95 for using rank 100. So, for approximating the inverse a high rank is needed for a low error. However, the computation time for computing the rank 100 approximation and inverse is higher than for directly computing the inverse of the original matrix.

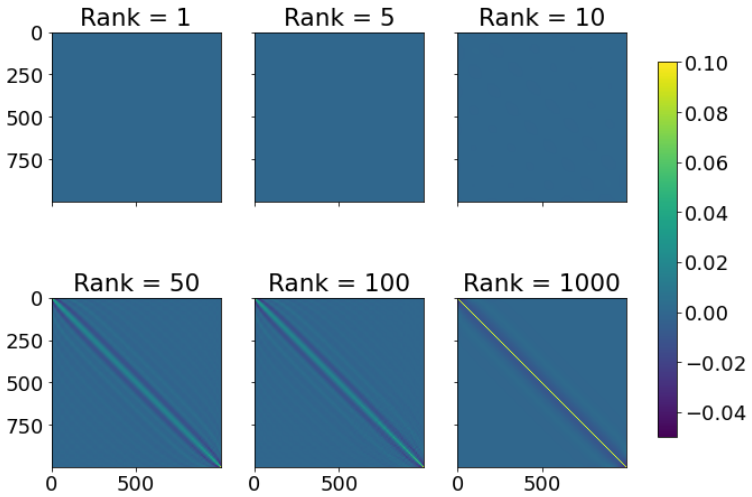


Figure 6.9: The inverse of approximations for matrix Σ_n (bottom-right) for different ranks. As the rank increases, more singular values are incorporated and the approximation gets better.

Despite the large errors, there may be enough structure in these approximations for the inverse to perform GPR. Unfortunately, no package in Python has been found that optimizes the hyperparameters using the approximations by SVD. So this optimization is programmed manually, though this can lead to inefficient computations. First, the optimization is done by the exact inverse and only the prediction is done by the approximated inverse, see Figure 6.10. Here, the exact predictions are visualized in green and the approximations in red for different ranks. One can see that for rank 1, it is simplifying the data and more information is included for higher ranks. For rank 10 the approximation is already quite well with some differences in the tails (where no observations are present).

The next step is to update the hyperparameters by using the approximation of the inverse in Formula 4.10. The determinant needs also to be computed, luckily this is done easily using the singular value decomposition of Σ_n:

$$\det(\Sigma_n) = \det(UDU^T) = \det(U)\det(D)\det(U^T) = \det(D) = \Pi_{i=1}^{n} d_{ii} = \Pi_{i=1}^{n} \sigma_i.$$

This is the result of the fact that the determinant of an orthogonal matrix is -1 or 1 and that $\det(U^T) = \det(U)$. For the exact method and the SVD approximations the L-BFGS-B algorithm is used to find the maximum log marginal likelihood where bound constraints for the hyperparameters can be used.

Unfortunately, no promising results have been found. Updating the parameters resulted in very general predictions or very specific predictions, but neither are close to the exact solution computed with Σ_n^{-1}. In Figure 6.11, some of the results of the simulations can be seen where the hyperparameters are updated with SVD.

The prediction with SVD has been simulated for the chlorophyll-a problem as well. For a subset of the satellite dataset containing 26,085 observations and 2866 in-situ observations,

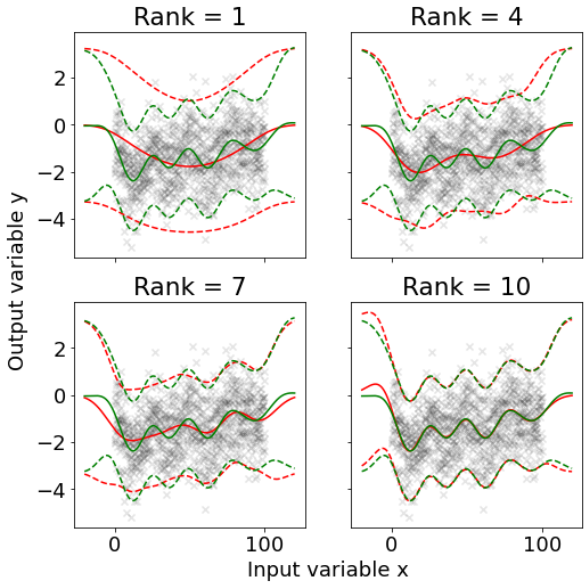


Figure 6.10: Posterior mean and twice the standard deviation at each input value x for the exact method (green) and with SVD for different ranks (red). The 1000 observations are visualized by the almost transparent black crosses. The predictions with SVD are approximating the exact predictions better when the rank increases. Note that the hyperparameters are fixed (optimized with the exact inverse).

the manual derivations (without using the package GPy) of the posterior mean and standard deviation are computed in approximately 10.3 seconds. Here, a Cholesky decomposition is used for computing the inverse. For several values of rank k, the posterior mean and standard deviation is computed and the computation time, mean absolute error of the mean and standard deviation has been noted. As the rank increases the absolute errors generally decreases and the computation time slowly increases. For the first few ranks ($k = 1, \ldots, 10$) the error decreases quite fast. When more singular values are taken into account, the mean absolute errors stay relatively constant. For rank $k = 10$ the mean absolute error for the posterior mean is 0.0483 with a standard error of $3 \cdot 10^{-4}$, see Table 6.2. Note that the values of the concentration chlorophyll-a range from 0.09 to 1.17. The absolute error of the standard error is 0.0281 with a standard error of $3 \cdot 10^{-4}$. Here, the values range from 0.02 to 1.10. The computation time for $k = 10$ is 8.5 seconds. So, the computation for the posterior mean and standard deviation can be done faster with an SVD approximation for the inverse. Unfortunately, the time is not substantially reduced with multiple factors of 10, though in percentage terms roughly 80% of the computation time is needed for $k = 10$. When the number of in-situ observations increases, the decrease of computation time will be more present.

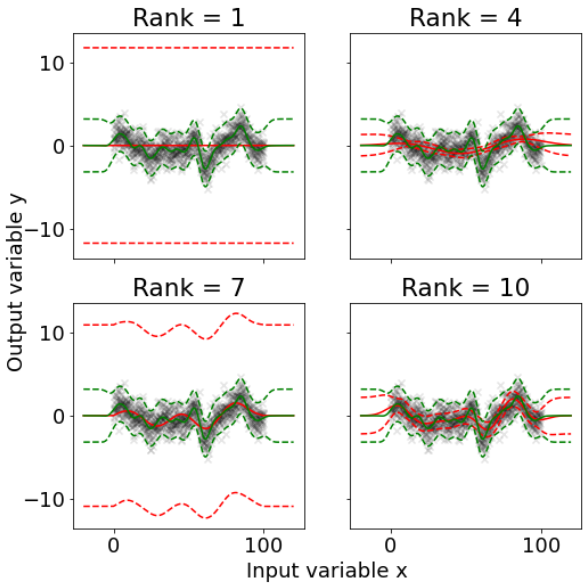


Figure 6.11: Posterior mean and twice the standard deviation at each input value x for the exact method (green) and with SVD for different ranks (red). The 1000 observations are visualized by the almost transparent black crosses. Note that the hyperparameters are updated with the corresponding SVD.

Rank	MAE mean (s.e.)	MAE std. (s.e.)	Time [s]
4	$0.1666\ (4 \cdot 10^{-4})$	$0.1515\ (6 \cdot 10^{-4})$	7.02
5	$0.0976\ (4 \cdot 10^{-4})$	$0.1081\ (5 \cdot 10^{-4})$	7.25
6	$0.0804\ (3 \cdot 10^{-4})$	$0.0866\ (4 \cdot 10^{-4})$	7.76
7	$0.0916\ (5 \cdot 10^{-4})$	$0.0957\ (5 \cdot 10^{-4})$	8.06
8	$0.0637\ (3 \cdot 10^{-4})$	$0.0687\ (4 \cdot 10^{-4})$	9.08
9	$0.0776\ (5 \cdot 10^{-4})$	$0.0772\ (4 \cdot 10^{-4})$	8.76
10	$0.0483\ (3 \cdot 10^{-4})$	$0.0557\ (3 \cdot 10^{-4})$	8.50

Table 6.2: Mean absolute error (MAE) of the posterior mean and standard deviation including the computation time for different ranks. The predictions are computed using the SVD approximation of the matrix inversion. The computation time is the time of computing the posterior mean and standard deviation.

7

Results

In the previous chapter, different approximation techniques were applied to see whether updating the hyperparameters and the prediction could be done faster, i.e. with fewer floating-point operations. As no single method stood out to speed up the inversion, the Cholesky decomposition is used here. In this chapter, we will focus on the main result of the GPR model which is the prediction of chlorophyll-a concentration on the locations where satellite data is present. All in-situ data from 9 August 2019 ($n = 2866$) and a subset of the satellite image is used. The subset contains 423 by 371 pixels which is in total 156,933 data points. With IDW interpolation, the reflectances at the in-situ data are computed. This is done in 40 seconds and depends on the number of satellite observations used as the distance is measured from every in-situ observation to every satellite observation. It is possible to speed-up this process by considering only the k closest data points instead of all satellite observations. We take the logarithm of the chlorophyll-a concentration to make sure that all predictions are positive values. This is done by using the log-normal distribution where the mean and variance for the final concentrations can be computed from the mean and variance of the logarithmic concentrations (see Appendix A.2).

The hyperparameters are computed by maximizing the log marginal likelihood using the L-BFGS-B optimization algorithm which does 93 iterations in 5 minutes and 34 seconds for convergence. The log marginal likelihood is maximized at a value of 14.74 (similar to the optimization in Table 5.4). The hyperparameters that set this log marginal likelihood are the same as when the hyperparameters were analyzed and are restated here for completeness: $\sigma_n = 0.67, \sigma_f = 0.22$ and lengthscale vector

$$l = [99.96, 100.00, 75.51, 0.009, 0.40, 0.0011, 0.009].$$

Then the posterior mean and standard deviation can be computed for the satellite observations, which is done in 2 minutes and 23 seconds. The results can be seen in Figure 7.1. Note that these observations are for a depth equal to one meter. The in-situ observations are visualized by the circles in color (left) and red dots (right). As the measurements are done close to each other, some overlap is present. The average concentration is about $0.11\ mg \cdot m^{-3}$ and the average standard deviation is around $0.07\ mg \cdot m^{-3}$. In the both plots, a line is visible from the lower-left corner $(25°E, 40.3°N)$ to $(25.05°E, 40.5°N)$. This is the remains of the atmospheric correction (AC). The satellite image is taken at 9 a.m., so the solar zenith angle is large (this means that the sun is close to the horizon). A low-resolution map is used to compensate for the difference of sunlight intensities on the area, which causes these unnatural straight lines visible in the estimations. The individual maps of the reflectances are created to confirm that

the line is not 'created' by our model but appears due to the input variables. Another observation is that the standard deviation values are relatively low for the low estimations in the bottom right corner and high for the higher concentrations. This seems to be a general rule for all the estimations: a low concentration is estimated with a higher certainty than a high concentration of chlorophyll-a.

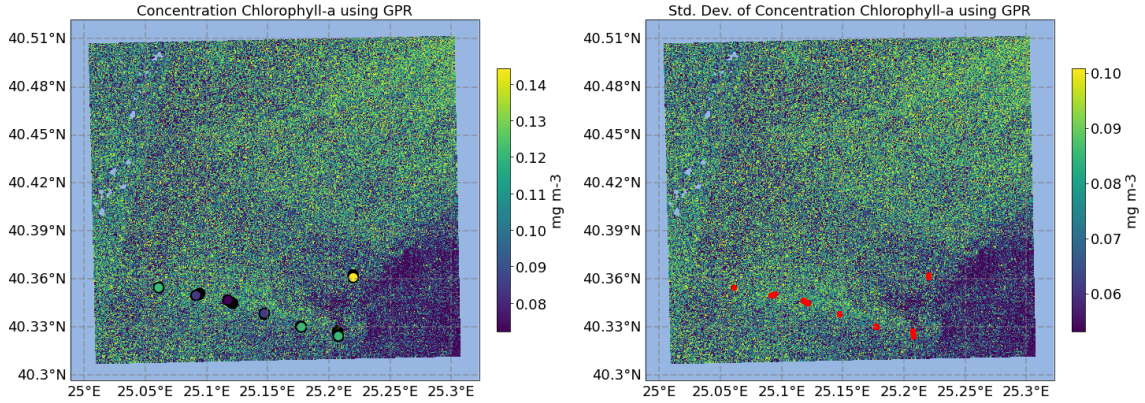


Figure 7.1: The posterior mean (left) and corresponding standard deviation (right) for the area around the in-situ observations. The color represents the value for each pixel and the circles are the in-situ locations/values (exaggerated in size for the visualization).

As the GPR model is mainly based on the reflectances, it is possible to do predictions at locations further away from the in-situ locations. First, the complete satellite image was tried to do estimations on, however, a memory error occurs when this is tried. The package GPy tries to allocate a matrix of size $2866 \times 3,348,900$ which contains almost 10 billion elements. From the error statement, it is possible to derive that this is the matrix Σ_{ff_*} that is used to compute the posterior mean and variance. This problem can be solved by splitting the remote sensing dataset in smaller pieces and iteratively computing the posterior mean and standard deviation. The inverse Σ_n^{-1} needs to be computed only once as this does only depend on the in-situ data.

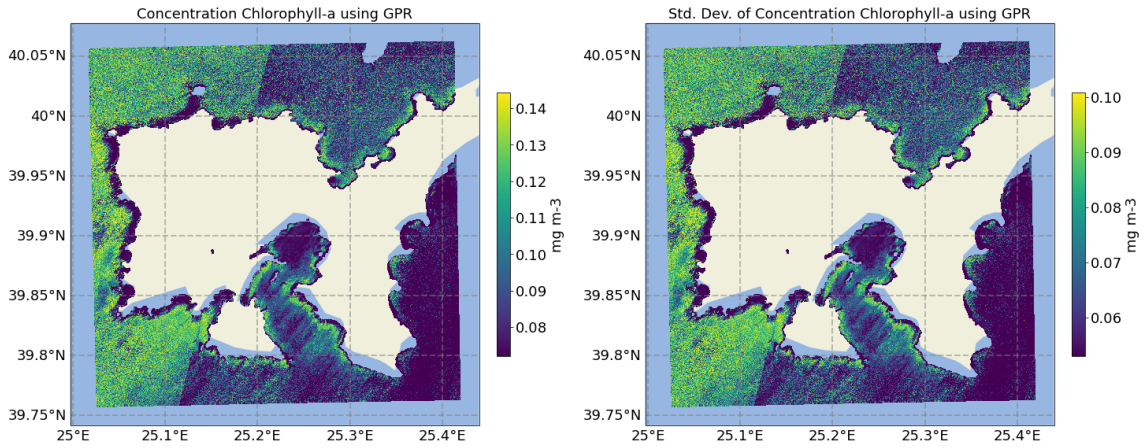


Figure 7.2: The posterior mean (left) and corresponding standard deviation (right) for the area around Limnos island.

A smaller subset is thus chosen around the island Limnos, see Figure 7.2. Some areas close to the coast have low concentrations (close to $0.08\ mg \cdot m^{-3}$), for example in the western part. Along the southern part of the coastline, a band of high concentrations of chlorophyll-a are

estimated. Likewise in the plot on the right, the standard deviation in these areas is relatively high which suggests that there is more uncertainty. Again a line is well visible in both plots due to the position of the sun. The verification of these figures can be done by adding chlorophyll-a maps of other sources. From CMEMS, monthly averages as well as daily averages from multiple satellite observations (SeaWIFS, MERIS, MODIS-Aqua, NPP- VIIRS, NOAA20-VIIRS). The spatial resolution is 1 kilometer and the corresponding chlorophyll-a map of 9 August 2019 around Limnos Island can be seen in Figure 7.3. From the left plot (same zoom level as before) it can be seen that estimations close to the coast are not available, therefore a zoomed out plot on the right is created. Noteworthy is the reduced resolution, the pixels can easily be identified and it is impossible to observe the structure along the coast as seen in Figure 7.2. Moreover, no uncertainty map is available for these plots. The average concentration chlorophyll is roughly the same in both estimations (around $0.1\ mg\ m^{-3}$), however the increase in concentration on the east coast is not visible in our estimations.

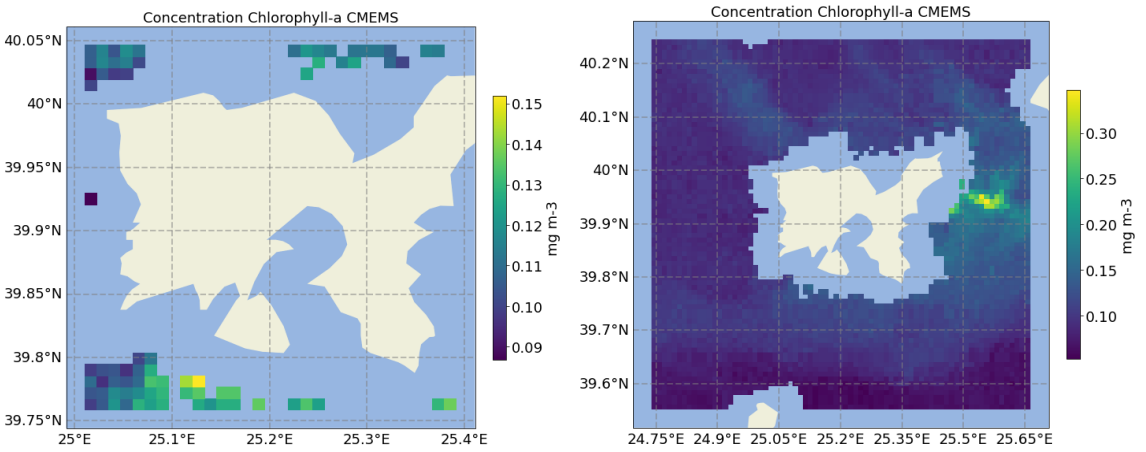


Figure 7.3: Estimation of chlorophyll-a concentration around Limnos island. Data retrieved from CMEMS.

Instead of a different area, the concentration of chlorophyll-a can also be estimated at a certain depth. From Figures 5.11 and 5.17 it is expected to obtain high values of concentrations for a depth around 90 meters as well as a high standard deviation. In Figure 7.4 the posterior mean and standard deviation are plotted for the same area as before, but now at a depth of 90 meters. Indeed, the estimations and uncertainties are significantly higher than before.

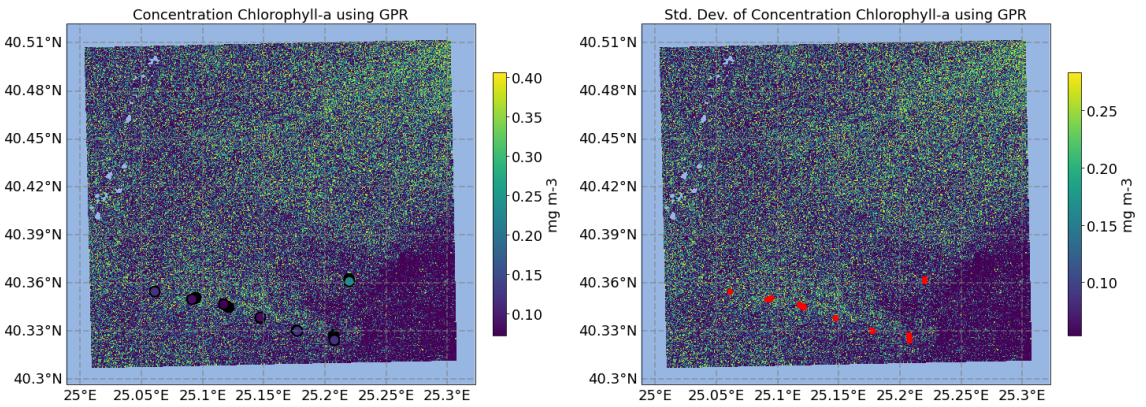


Figure 7.4: The posterior mean (left) and corresponding standard deviation (right) for the area around the in-situ observations. The color represents the value for each pixel and the circles are the in-situ locations/values (exaggerated in size for the visualization). The top estimations are done at a depth of 90 meters.

For completeness, another estimation is done at a depth of 200 meters, which can be seen in Figure 7.5. Clearly, the concentrations are very low in comparison with the other figures as well as the uncertainties. It can also be seen that low concentrations at a depth of 1 meter tend to stay relatively low at a depth of 200 meters. Similarly, high concentrations at 1 meter tend to be the high concentrations at 200 meters as well.

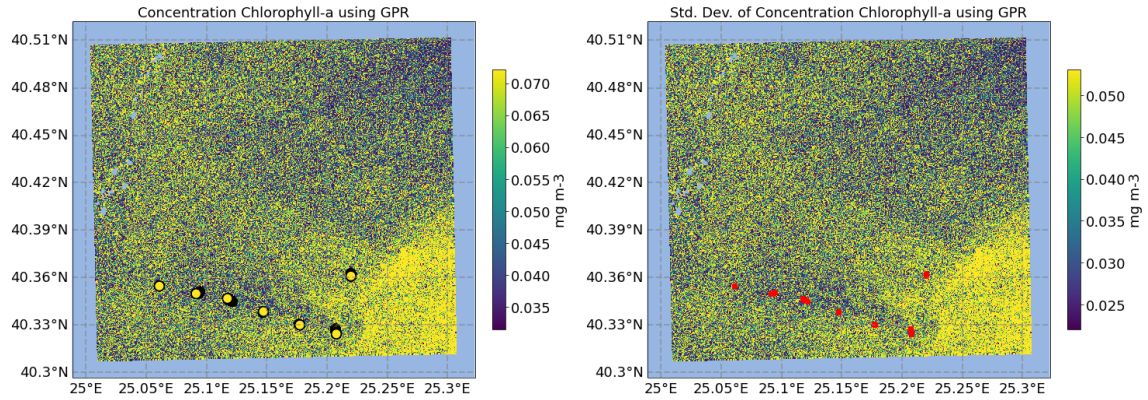


Figure 7.5: The posterior mean (left) and corresponding standard deviation (right) for the area around the in-situ observations. The color represents the value for each pixel and the circles are the in-situ locations/values (exaggerated in size for the visualization). The estimations are done at a depth of 200 meters.

From the parameter analysis in section 5.5 it can be concluded that the variables longitude, latitude, as well as the green reflectance, can be removed from the model. When doing so, for both Figure 7.1 and 7.2 the estimations are identical. Removing any other variable resulted in significant changes in the estimations as the log marginal likelihood is reduced. This reduces the computation time by a little because the matrix that needs to be inverted remains to be the same size.

Furthermore, for the state-of-the-art model, the estimations for the chlorophyll-a concentration are computed. An estimation is computed for the subset around the in-situ data and around Limnos island, see Figure 7.6. The computation time of the IOP variable was approximately 45 minutes, while training and predicting is done in less than 2 seconds. Note that the data product imported into the SNAP tool is 1C (TOA) and that the C2RCC algorithm does the AC itself. Comparing these estimations with the left figures of Figure 7.1 and 7.2, it is noticable that the AC is done better here. Furthermore, there is no classification dataset available so the clouds in the left plot are included and the island Limnos is manually removed from the dataset. The estimated concentrations for the C2RCC algorithm are generally a bit lower. The GPR model indicates that there is an area in the bottom-right corner with lower concentrations (see Figure 7.1). This is not seen in the figure below. For the area around Limnos island, a narrow band with relatively high chlorophyll-a concentrations can be seen especially on the south and east coast of the island. The uncertainty quantification for both plots is not included as no literature is found that explains how this can be done exactly. Note that the concentrations are computed at the surface of the sea. The depth is not incorporated in this method and no estimations can be done for different depths.

The computation for the parameters is done using all in-situ observations, including observations that are taken far below the surface of the sea. When considering only the observations close to the surface (in this example, observations with a depth smaller than 5 meters), the estimations are computed and shown in Figure 7.7. The estimated concentrations are a bit higher than before (average increased by 0.013 $mg\ m^{-3}$) and the maximum concentration is now $\approx$ 0.18 $mg\ m^{-3}$. In contrast with Figure 7.6, low concentrations can be found near the coastline instead of high concentrations before.

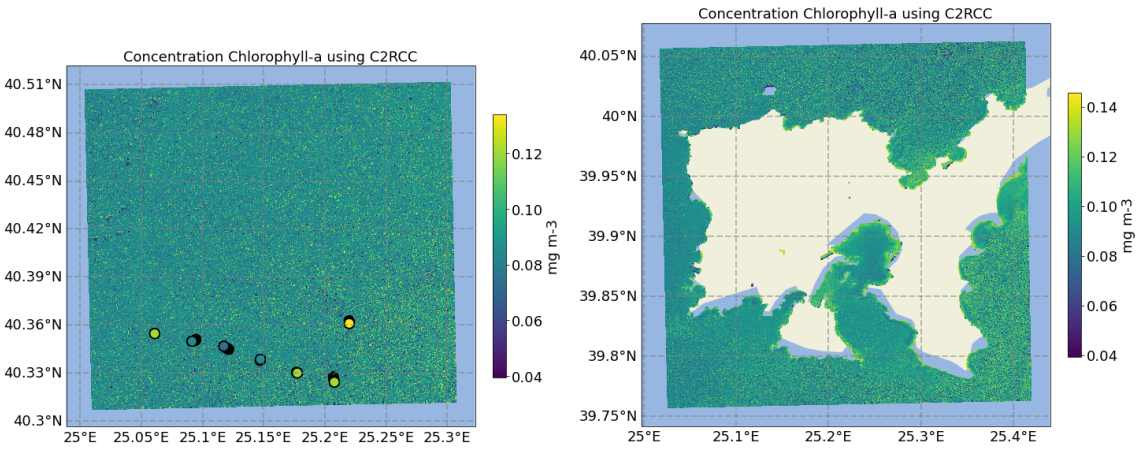


Figure 7.6: Chlorophyll-a concentration estimation using the C2RCC algorithm. The subset around the in-situ observations is shown on the left. On the right the area around Limnos island is shown.

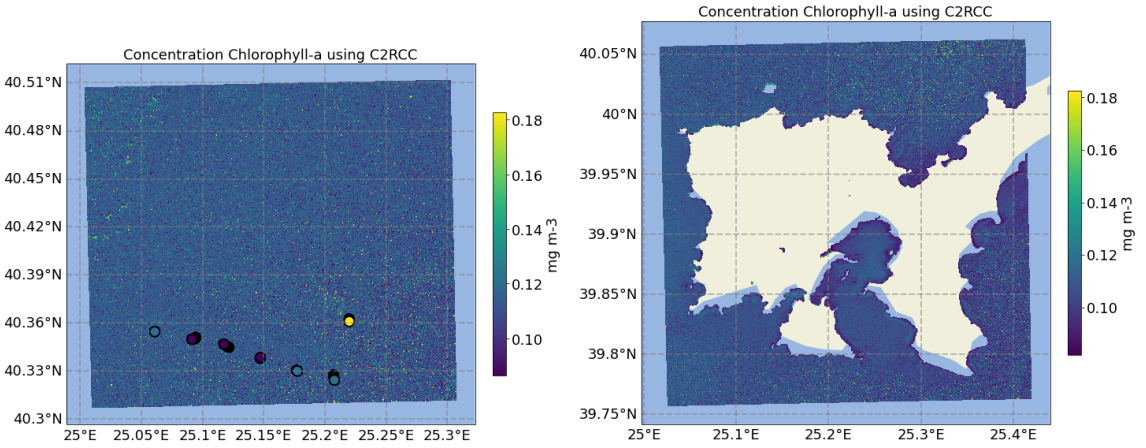


Figure 7.7: Chlorophyll-a concentration estimation using the C2RCC algorithm when considering the shallow observations only (in this example, depth is smaller than 5 meters). The subset around the in-situ observations is shown on the left. On the right the area around Limnos island is shown.

Finally, the drawback of the GPR model is that a matrix of size n needs to be inverted, where n is the number of in-situ observations. This needs to be done when updating the hyperparameters (once for every iteration) and when computing the posterior mean and standard deviation. For a single in-situ dataset, the models need to be 'trained' once to set the hyperparameters, whereafter the model can be used to do predictions. In this last step the inverse remains exactly the same, as the matrix only depends on the in-situ data, so computing the estimations can be done relatively fast. This last step is dominated by matrix-matrix multiplications. Multiple approximation techniques have been investigated to speed up the matrix inversion. With singular value decomposition it seems to be possible to compute accurate estimations with the approximated matrix. However, for computing the hyperparameters, there is not enough structure in the approximation to get accurate results.

8

Conclusion and Discussion

In this chapter, the research and analysis will be summarized and an answer will be given to the main research question which was stated in the introduction: *What spatio-temporal model, that provides uncertainty quantification, can be used to estimate the chlorophyll-a concentration using high-resolution optical remote sensing data?* Afterwards, the limitations and advantages of the GPR model are discussed. Finally, some recommendation for future research will be given.

8.1. Conclusion

In this thesis, we tried to estimate the concentration chlorophyll-a using the reflectances obtained by high-resolution satellite sensors. The state-of-the-art models are based on machine learning techniques that require large amount of training data, take a long time to train the model and give an uncertainty quantification by giving the input variables a small deviation. Furthermore, the C2RCC algorithm uses neural networks for which the reason of the choice for the number of hidden layers and neurons is ambiguous. The advantage is, once trained, that the model should be able to estimate the concentrations quite fast.

A Gaussian process regression method is proposed to estimate the chlorophyll-a concentration using the reflectances. It can be calibrated with limited data access and has an uncertainty quantification caused by the Bayesian structure. However, the main limitation is that a matrix inversion is needed for finding the hyperparameters and to do predictions.

In-situ data from the ODYSSEA project and the remote sensing data from the Sentinel-2 satellite were obtained. For the in-situ data, the requirements were that spatial as well as temporal data was desired and that the acquired observations are measured in the last couple of years. For the remote-sensing data, freely available data from a high-resolution sensor was desired. This was not available, so as alternative, the medium-resolution Sentinel-2 satellite data is utilized.

First the reflectances were interpolated on the in-situ locations, whereafter the GPR model was analyzed extensively, with a focus on the covariance function and the hyperparameters. The squared-exponential kernel is used most often, though there is no reason why, other than convenience and simplicity. Therefore, other Matérn kernels, a linear kernel and a multilayer perceptron kernel were analyzed. Using cross-validation the Matérn kernel with $\nu = \frac{3}{2}$ was suggested based on the mean-squared-error as well as the log marginal likelihood. The hyperparameters were the noise variance σ_n^2, lengthscale vector l and the process variance σ_f^2. A graph was made of the log marginal likelihood versus each of the hyperparameters. As a lengthscale for each variable was used, a variable selection procedure was possible by investigating these graphs. It became clear that the longitude, latitude and green wavelength did

not improve the log marginal likelihood and could be removed from the model. More analysis was done by making contour plots of the log marginal likelihood for different values of two hyperparameters. Finally, continuous plots were created. So the concentration chlorophyll-a can be estimated along, for example, the depth at a certain location.

The GPR model was compared with the C2RCC algorithm and the polynomial regression techniques. For both these models it became clear that having observations close to the surface is needed to train the models, because the depth is not included. Indeed, the MSE decreased, however this is mainly caused by having less variation, this was supported by the value of R^2. For both the shallow observations and all observations, the GPR model performed best in terms of MSE and R^2.

Finally, the drawback of GPR is investigated by testing multiple approximation techniques for computing the inverse of a matrix. Maximizing the log marginal likelihood requires multiple matrix inversions and computing the posterior mean and standard deviation requires one more matrix inversion. This only needs to be done once, after which estimating the chlorophyll-a concentration can be done relatively fast (dominated by matrix-matrix multiplications). With a singular value decomposition it seems to be possible to approximate an inverse of a matrix to produce accurate estimations. However, maximizing the log marginal likelihood gave inaccurate results in comparison to using the Cholesky decomposition.

8.2. Assumptions

In this section, we state the assumptions of the GPR model and elaborate on them.

The most important assumption of this model is to assume that the real process follows a Gaussian process. The chlorophyll-a concentration can only take positive values, so can never follow a Gaussian distribution. Despite this fact, previous research has shown that a GPR can be used to model the chlorophyll-a concentration and this is established in this research once again.

Another assumption made along the process is that the interpolated reflectances are the 'true' reflectances. As explained in more detail in the analysis, this can be justified by the use of medium/high-resolution satellite data. So, the error made by interpolation will be very small.

The change of spatial support is a relevant issue in the types of variables used in this research (Gelfand et al., 2001). Satellites observe an average emission along the reflective path in the water column and this may be changing for each wavelength. The blue light will penetrate the water more easily in comparison to red light (that is why deep-sea animals have a red color). So, the problem in spatial support lies in the scale of the measurements, as the glider measures the chlorophyll-a concentration using a small subset of the water, while satellite measures in a 10x10m times the penetrated depth. There is almost no variability in the 10x10 meter grid, however, there is a lot of variability available in the depth as we saw before. So the satellite is observing an average across the depth, while the chlorophyll-a concentration is a point observation. Using an isotropic kernel ensures a low covariance between two observations when the value for the depth is far apart. Furthermore, the reflectance data that is obtained from Sentinel-2 is preprocessed such that it should represent the radiances reflected from the water surface.

Finally, it is assumed, doing our estimations, that the in-situ data is measured at the same time as the satellite data. This is not true, as the satellite obtains the data once at a certain time whilst the glider measures concentrations twice every minute. This is a necessary assumption, because it is desired to have the reflectances at the in-situ locations for which the only source is the satellite data. When the 'true' reflectances are used, problems will arise for measurements taken at night.

8.3. Recommendations

In this research, Gaussian process regression is analyzed quite extensively. However, more analysis can be done and the model can be expanded. In this section, a list of recommendations will be given for future work.

High-resolution satellite data can be used to analyze the performance of the GPR. As the in-situ data remains the same, the hyperparameters will be unaltered. However, for computational requirements, it is interesting to see how the model performs and if anything can be done to improve the computation time for estimating the concentrations.

The GPR model can be extended by including the time variable. First, it is desired to find recent in-situ data in other seasons (for cloudless days), then the hyperparameters need to be calibrated and the inclusion of other covariance functions can be analyzed for modelling the chlorophyll-a concentration, e.g. a periodic kernel. Covariance functions can be multiplied and added to each other, so a Matérn kernel for the reflectances and depth, multiplied with a periodic kernel for the time variable can be tested.

Instead of the chlorophyll-a concentration, other water quality indicators, such as colored dissolved organic matter, can be used. It would be interesting to see how accurate the predictions are and also how the model performs in other areas than the Thracian sea. The GPR model can easily be adapted for other variables and areas.

Furthermore, the computational complexity of updating the hyperparameters can be analyzed more. There are a number of suggestions to look into, such as stochastic gradient descent, which is a technique that uses samples of observations to maximize the marginal log likelihood. Instead of variational learning, other sparse GPR techniques, such as the Nyström method or using a subset of datapoints instead of the complete dataset, can be investigated. Another method is to create a sparse covariance matrix by setting elements to zero when the value is below a certain threshold.

Finally, an interesting extension of a Gaussian process regression model is shown by Dunson et al. (2020), where a specific kernel is designed such that the geometry of the domain is respected. Here, an approximation for the covariance is used based on finitely-many eigenpairs of the Graph Laplacian (GL). With this algorithm, it is possible to incorporate the physical effects of having a narrow lake or an island between two relatively close waters. In a 'regular' Gaussian process model with Euclidean distance, the chlorophyll-a concentration (or any other variable) is able to be (inappropriately) smoothed across the land, which is not desirable. This type of inaccuracies can occur in areas where narrow islands or landmasses split waters such as the waters around Long Island (New York) and the Gulf of California.

Bibliography

Abramowitz, M., & Stegun, I. A. (1964). *Handbook of mathematical functions with formulas, graphs, and mathematical tables* (Vol. 55). US Government printing office.

Akgul, Y. (2021). *Topshot turkey sea environment*. AFP. www.gettyimages.nl/license/1233338742. License: Creative Commons BY-NC-SA

Almeida, L. P., Almar, R., Bergsma, E. W., Berthier, E., Baptista, P., Garel, E., Dada, O. A., & Alves, B. (2019). Deriving high spatial-resolution coastal topography from sub-meter satellite stereo imagery. *Remote Sensing*, *11*(5), 590.

Amato, U., Antoniadis, A., Cuomo, V., Cutillo, L., Franzese, M., Murino, L., & Serio, C. (2008). Statistical cloud detection from seviri multispectral images. *Remote Sensing of Environment*, *112*(3), 750–766.

Anderson, D. M., Glibert, P. M., & Burkholder, J. M. (2002). Harmful algal blooms and eutrophication: Nutrient sources, composition, and consequences. *Estuaries*, *25*(4), 704–726.

Bazi, Y., Alajlan, N., Melgani, F., AlHichri, H., & Yager, R. R. (2014). Robust estimation of water chlorophyll concentrations with gaussian process regression and iowa aggregation operators. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(7), 3019–3028.

Bhatia, R., & Jain, T. (2009). Higher order derivatives and perturbation bounds for determinants. *Linear Algebra and its Applications*, *431*(11), 2102–2108.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Boain, R. J. (2004). Ab-cs of sun-synchronous orbit mission design.

Brockmann, C., Doerffer, R., Peters, M., Kerstin, S., Embacher, S., & Ruescas, A. (2016). Evolution of the c2rcc neural network for sentinel 2 and 3 for the retrieval of ocean colour products in normal and extreme optically complex waters. *Living Planet Symposium*, *740*, 54.

Calewaert, J.-B., Weaver, P., Gunn, V., Gorringe, P., & Novellino, A. (2016). The european marine data and observation network (emodnet): Your gateway to european marine and coastal data. *Quantitative monitoring of the underwater environment* (pp. 31–46). Springer.

Campbell, J. B., & Wynne, R. H. (2011). *Introduction to remote sensing*. Guilford Press.

Cannizzaro, J. P., & Carder, K. L. (2006). Estimating chlorophyll a concentrations from remote-sensing reflectance in optically shallow waters. *Remote Sensing of Environment*, *101*(1), 13–24.

Cullen, J. J. (1982). The deep chlorophyll maximum: Comparing vertical profiles of chlorophyll a. *Canadian Journal of Fisheries and Aquatic Sciences*, *39*(5), 791–803.

Cutillo, L., Amato, U., Antoniadis, A., Cuomo, V., & Serio, C. (2004). Cloud detection from multispectral satellite images. *Proceedings from IEEE Gold Conference, University Parthenope, Naples, Italy*.

Dall'Olmo, G., & Gitelson, A. A. (2005). Effect of bio-optical parameter variability on the remote estimation of chlorophyll-a concentration in turbid productive waters: Experimental results. *Applied optics*, *44*(3), 412–422.

DigitalGlobe. (2010). *Radiometric use of worldview-2 imagery: Technical note* (tech. rep.). Colorado, USA.

Doerffer, R. (2015). Algorithm theoretical bases document (atbd) for l2 processing of meris data of case 2 waters, 4 th reprocessing. *Rapport technique*, *2*.

Doerffer, R., & Schiller, H. (2007). The meris case 2 water algorithm. *International Journal of Remote Sensing*, *28*(3-4), 517–535.

Drusch, M., Del Bello, U., Carlier, S., Colin, O., Fernandez, V., Gascon, F., Hoersch, B., Isola, C., Laberinti, P., Martimort, P., et al. (2012). Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote sensing of Environment*, *120*, 25–36.

Dunson, D. B., Wu, H., & Wu, N. (2020). Diffusion based gaussian processes on restricted domains.

Enomoto, T. (2008). Bicubic interpolation with spectral derivatives. *SOLA*, *4*, 5–8.

European Space Agency. (2019). Working towards AI and Earth observation. https://www.esa.int/Applications/Observing_the_Earth/Working_towards_AI_and_Earth_observation

European Space Agency. (2020). Space debris by numbers. Retrieved December 8, 2020, from https://www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_numbers

Fang, Q., Hong, H., Zhao, L., Kukolich, S., Yin, K., & Wang, C. (2018). Visible and near-infrared reflectance spectroscopy for investigating soil mineralogy: A review. *Journal of Spectroscopy*.

Fletcher, K. (2012). *Sentinel-2 : Esa's optical high-resolution mission for gmes operational services*. ESA Communications.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *international conference on machine learning*, 1050–1059.

Gelfand, A. E., Zhu, L., & Carlin, B. P. (2001). On the change of support problem for spatio-temporal data. *Biostatistics*, *2*(1), 31–45.

Gons, H. J., Auer, M. T., & Effler, S. W. (2008). Meris satellite chlorophyll mapping of oligotrophic and eutrophic waters in the laurentian great lakes. *Remote Sensing of Environment*, *112*(11), 4098–4106.

Gons, H. J., Rijkeboer, M., & Ruddick, K. G. (2002). A chlorophyll-retrieval algorithm for satellite imagery (medium resolution imaging spectrometer) of inland and coastal waters. *Journal of Plankton Research*, *24*(9), 947–951.

GPy. (since 2012). GPy: A gaussian process framework in python.

Grobe, H., Diepenbroek, M., Dittert, N., Reinke, M., & Sieger, R. (2006). Archiving and distributing earth-science data with the pangaea information system. *Antarctica* (pp. 403–406). Springer.

Han, L. (1997). Spectral reflectance with varying suspended sediment concentrations in clear and algae-laden waters. *Photogrammetric engineering and remote sensing*, *63*(6), 701–705.

Harpham, C., & Dawson, C. W. (2006). The effect of different basis functions on a radial basis function network for time series prediction: A comparative study. *Neurocomputing*, *69*(16-18), 2161–2170.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hensman, J., Matthews, A., & Ghahramani, Z. (2015). Scalable variational gaussian process classification. *Artificial Intelligence and Statistics*, 351–360.

Ho, J. C., Michalak, A. M., & Pahlevan, N. (2019). Widespread global increase in intense lake phytoplankton blooms since the 1980s. *Nature*, *574*(7780), 667–670.

Hooker, S. B., Firestone, E. R., OReilly, J. E., Maritorena, S., OBrien, M. C., Siegel, D. A., Toole, D., Mueller, J. L., Mitchell, B. G., Kahru, M., et al. (2000). Seawifs postlaunch technical report series. volume 11; seawifs postlaunch calibration and validation analyses.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, *2*(5), 359–366.

Huisman, J., Thi, N. N. P., Karl, D. M., & Sommeijer, B. (2006). Reduced mixing generates oscillations and chaos in the oceanic deep chlorophyll maximum. *Nature*, *439*(7074), 322–325.

Hwang, J. W., & Lee, H. S. (2004). Adaptive image interpolation based on local gradient features. *IEEE signal processing letters*, *11*(3), 359–362.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Kabbara, N., Benkhelil, J., Awad, M., & Barale, V. (2008). Monitoring water quality in the coastal area of tripoli (lebanon) using high-resolution satellite data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *63*(5), 488–495.

Kang, F., Xu, B., Li, J., & Zhao, S. (2017). Slope stability evaluation using gaussian processes with various covariance functions. *Applied Soft Computing*, *60*, 387–396.

Kim, S., Kim, H., Lee, J., Yoon, S., Kahou, S. E., Kashinath, K., & Prabhat, M. (2019). Deep-hurricane-tracker: Tracking and forecasting extreme climate events. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1761–1769.

Klose, C., Reuter, R., Byfield, V., & Robertson, C. (n.d.). Marine pollution. Retrieved July 1, 2021, from https://seos-project.eu/marinepollution/marinepollution-c09-p01

Le Traon, P. Y., Reppucci, A., Alvarez Fanjul, E., Aouf, L., Behrens, A., Belmonte, M., Bentamy, A., Bertino, L., Brando, V. E., Kreiner, M. B., et al. (2019). From observation to information and users: The copernicus marine service perspective. *Frontiers in Marine Science*, *6*, 234.

Lebègue, L., Greslou, D., deLussy, F., Fourest, S., Blanchet, G., Latry, C., Lachérade, S., Delvit, J.-M., Kubik, P., Déchoz, C., et al. (2012). Pleiades-hr image quality commissioning. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *39*, B1.

Lee, Z., Zhang, M., Carder, K., & Hall, L. (1998). A neural network approach to deriving optical properties and depths of shallow waters. *Proceedings, Ocean Optics XIV*.

Li, C., Zhu, X., Wei, Y., Cao, S., Guo, X., Yu, X., & Chang, C. (2018). Estimating apple tree canopy chlorophyll content based on sentinel-2a remote sensing imaging. *Scientific reports*, *8*(1), 1–10.

Liew, S., Saengtuksin, B., & Kwoh, L. (2011). Mapping water quality of coastal and inland waters using high resolution worldview-2 satellite imagery. *Proc. 34th. International Symposium on Remote Sensing of Environment*, 10–15. https://www.isprs.org/proceedings/2011/ISRSE-34/211104015Final00763.pdf

Mahajan, S., & Fataniya, B. (2019). Cloud detection methodologies: Variants and development—a review. *Complex & Intelligent Systems*, 1–11.

Minka, T. P. (2001). *A family of algorithms for approximate bayesian inference* (Doctoral dissertation). Massachusetts Institute of Technology.

Mishra, S., & Mishra, D. R. (2012). Normalized difference chlorophyll index: A novel model for remote estimation of chlorophyll-a concentration in turbid productive waters. *Remote Sensing of Environment*, *117*, 394–406.

Mohammed, R. O., & Cawley, G. C. (2017). Over-fitting in model selection with gaussian process regression. *International Conference on Machine Learning and Data Mining in Pattern Recognition*, 192–205.

Mongillo, M. (2011). Choosing basis functions and shape parameters for radial basis function methods. *SIAM undergraduate research online*, *4*(190-209), 2–6.

Moses, W. J., Gitelson, A. A., Berdnikov, S., & Povazhnyy, V. (2009). Satellite estimation of chlorophyll-a concentration using the red and nir bands of meris—the azov sea case study. *IEEE Geoscience and Remote Sensing Letters*, *6*(4), 845–849.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT press.

Neal, R. M. (2012). *Bayesian learning for neural networks* (Vol. 118). Springer Science & Business Media.

O'Reilly, J. E., Maritorena, S., Mitchell, B. G., Siegel, D. A., Carder, K. L., Garver, S. A., Kahru, M., & McClain, C. (1998). Ocean color chlorophyll algorithms for seawifs. *Journal of Geophysical Research: Oceans*, *103*(C11), 24937–24953.

O'Reilly, J. E., & Werdell, P. J. (2019). Chlorophyll algorithms for ocean color sensors-oc4, oc5 & oc6. *Remote sensing of environment*, *229*, 32–47.

Parks, L. (2009). Digging into google earth: An analysis of "crisis in darfur". *Geoforum*, *40*(4), 535–545.

Pasolli, L., Melgani, F., & Blanzieri, E. (2010). Gaussian process regression for estimating chlorophyll concentration in subsurface waters from remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, *7*(3), 464–468.

Petruzzello, M. (2020). Chlorophyll. https://www.britannica.com/science/chlorophyll

Pignatti, S., Palombo, A., Pascucci, S., Romano, F., Santini, F., Simoniello, T., Umberto, A., Vincenzo, C., Acito, N., Diani, M., et al. (2013). The prisma hyperspectral mission: Science activities and opportunities for agriculture and land monitoring. *2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS*, 4558–4561.

Quinonero-Candela, J., Rasmussen, C. E., & Williams, C. K. (2007). Approximation methods for gaussian process regression. *Large-scale kernel machines* (pp. 203–223). MIT Press.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. MIT Press Ltd. https://www.ebook.de/de/product/5192092/carl_edward_university_of_cambridge_rasmussen_christopher_k_i_university_of_edinburgh_williams_gaussian_processes_for_machine_learning.html

Rufai, A. M., Anbarjafari, G., & Demirel, H. (2014). Lossy image compression using singular value decomposition and wavelet difference reduction. *Digital signal processing*, *24*, 117–123.

Santner, T. J., Williams, B. J., Notz, W. I., & Williams, B. J. (2003). *The design and analysis of computer experiments* (Vol. 1). Springer.

Schaap, D. M., & Lowry, R. K. (2010). Seadatanet–pan-european infrastructure for marine and ocean data management: Unified access to distributed data sets. *International Journal of Digital Earth*, *3*(S1), 50–69.

Sellers, J. J., Astore, W. J., Giffen, R. B., & Larson, W. J. (2000). *Understanding space: An introduction to astronautics* (D. H. Kirkpatrick, Ed.; Third). McGraw-Hill Companies.

Sentinel-2 MSI - Technical Guide. (n.d.). Retrieved August 1, 2021, from https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi

Sishodia, R. P., Ray, R. L., & Singh, S. K. (2020). Applications of remote sensing in precision agriculture: A review. *Remote Sensing*, *12*(19), 3136.

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, *14*(3), 199–222.

SNAP. (2020, October 19). *Esa sentinel application platform* (Version 8.0.0). http://step.esa.int

Sofianos, S., Coppini, G., & Alvarez Fanjul, E. (2018). *Mongoos science and strategy plan*.

Soppa, M. A., Peeken, I., & Bracher, A. (2017). Global chlorophyll a concentrations for diatoms, haptophytes and prokaryotes obtained with the Diagnostic Pigment Analysis of HPLC data compiled from several databases and individual cruises. PANGAEA. https://doi.org/10.1594/PANGAEA.875879

Spanoudaki, K., Kokkos, N., Zachopoulos, K., Sylaios, G., Kampanis, N., de Koning, D., Meszaros, L., Wanke, S., & El Serafy, G. (2020). Monitoring and forecasting of marine pollution in the mediterranean sea: The odyssea project approach. *EGU General Assembly Conference Abstracts*, 15191.

Stein, M. L. (2012). *Interpolation of spatial data: Some theory for kriging*. Springer Science & Business Media.

Sun, D., Li, Y., & Wang, Q. (2009). A unified model for remotely estimating chlorophyll a in lake taihu, china, based on svm and in situ hyperspectral data. *IEEE transactions on geoscience and remote sensing*, *47*(8), 2957–2965.

Titsias, M. (2009). Variational learning of inducing variables in sparse gaussian processes. In D. van Dyk & M. Welling (Eds.), *Proceedings of the twelth international conference on artificial intelligence and statistics* (pp. 567–574). PMLR. http://proceedings.mlr.press/v5/titsias09a.html

Tobler, W. R. (1970). A computer movie simulating urban growth in the detroit region. *Economic geography*, *46*(sup1), 234–240.

Tsang, I. W., Kwok, J. T., Cheung, P.-M., & Cristianini, N. (2005). Core vector machines: Fast svm training on very large data sets. *Journal of Machine Learning Research*, *6*(4).

Tzortziou, M., Subramaniam, A., Herman, J. R., Gallegos, C. L., Neale, P. J., & Harding Jr, L. W. (2007). Remote sensing reflectance and inherent optical properties in the mid chesapeake bay. *Estuarine, Coastal and Shelf Science*, *72*(1-2), 16–32.

Uğurtaş, S. (2021). Turkey struck by 'sea snot' because of global heating. *The Guardian*. https://www.theguardian.com/environment/2021/may/25/turkey-struck-by-sea-snot-because-of-global-heating

Van Rossum, G., & Drake Jr, F. L. (1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer, New York.

Verrelst, J., Alonso, L., Caicedo, J. P. R., Moreno, J., & Camps-Valls, G. (2012). Gaussian process retrieval of chlorophyll content from imaging spectroscopy data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *6*(2), 867–874.

Vuik, C., Vermolen, F. J., Gijzen, M. B., & Vuik, M. (2007). *Numerical methods for ordinary differential equations*. VSSD.

Wang, X., Gong, Z., & Pu, R. (2018). Estimation of chlorophyll a content in inland turbidity waters using worldview-2 imagery: A case study of the guanting reservoir, beijing, china. *Environmental monitoring and assessment*, *190*(10), 620.

Wikle, C. K., Zammit-Mangion, A., & Cressie, N. (2019). *Spatio-temporal statistics with r*. Chapman & Hall/CRC.

Williams, C. K. (1998). Computation with infinite neural networks. *Neural Computation*, *10*(5), 1203–1216.

Woźniak, S. B., Darecki, M., & Sagan, S. (2019). Empirical formulas for estimating backscattering and absorption coefficients in complex waters from remote-sensing reflectance spectra and examples of their application. *Sensors*, *19*(18), 4043.

Zhang, Y., Liu, M., Qin, B., Van Der Woerd, H. J., Li, J., & Li, Y. (2009). Modeling remote-sensing reflectance and retrieving chlorophyll-a concentration in extremely turbid case-2 waters (lake taihu, china). *IEEE Transactions on Geoscience and Remote Sensing*, *47*(7), 1937–1948.

A

Probability Density Distributions

The probability density functions for the distributions used in this research are added here for completeness.

A.1. Multivariate Normal Distribution

Let X be a d-dimensional random vector with $X = (X_1, \ldots, X_d)^T$ and multivariate normal distributed. With the mean vector $\boldsymbol{\mu} = \mathbb{E}[X]$ and covariance matrix Σ the following notation is used: $X \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$. The element on row i and column j is equal to: $\Sigma_{i,j} = cov(X_i, X_j)$.

For a symmetric positive definite covariance matrix Σ, the probability density function can be written as:

$$f_X(x_1, \ldots, x_d) = \frac{\exp(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}))}{\sqrt{(2\pi)^d |\Sigma|}}$$

. Where $\mathbf{x} = [x_1, \ldots, x_d]^T$ is a real column vector.

A.2. Log-Normal Distribution

Let X be normally distributed with mean μ and variance $\sigma^2 > 0$ (i.e. $X \sim \mathcal{N}(\mu, \sigma^2)$). Then $Y = e^X$ is log-normal distributed with the following probability density function:

$$f_Y(y) = \begin{cases} \frac{1}{y\sigma\sqrt{2\pi}} \exp(-\frac{1}{2\sigma^2}(\ln(y) - \mu)^2) & \text{if } y > 0, \\ 0 & \text{if } y \leq 0. \end{cases}$$

Moreover, $X = \ln(Y) = \mathcal{N}(\mu, \sigma^2)$ so the mean of Y: $\mathbb{E}[Y] = \exp(\mu + \frac{\sigma^2}{2})$ and the variance: $Var(Y) = \exp(2\mu + \sigma^2)[\exp(\sigma^2) - 1]$. This distribution is used for the chlorophyll-a concentration, as a concentration is always positive.

B

Additional Data Analysis

B.1. Log Marginal Likelihood versus Hyperparameters

In this section, all plots of the log marginal likelihood versus each of the hyperparameters is plotted.

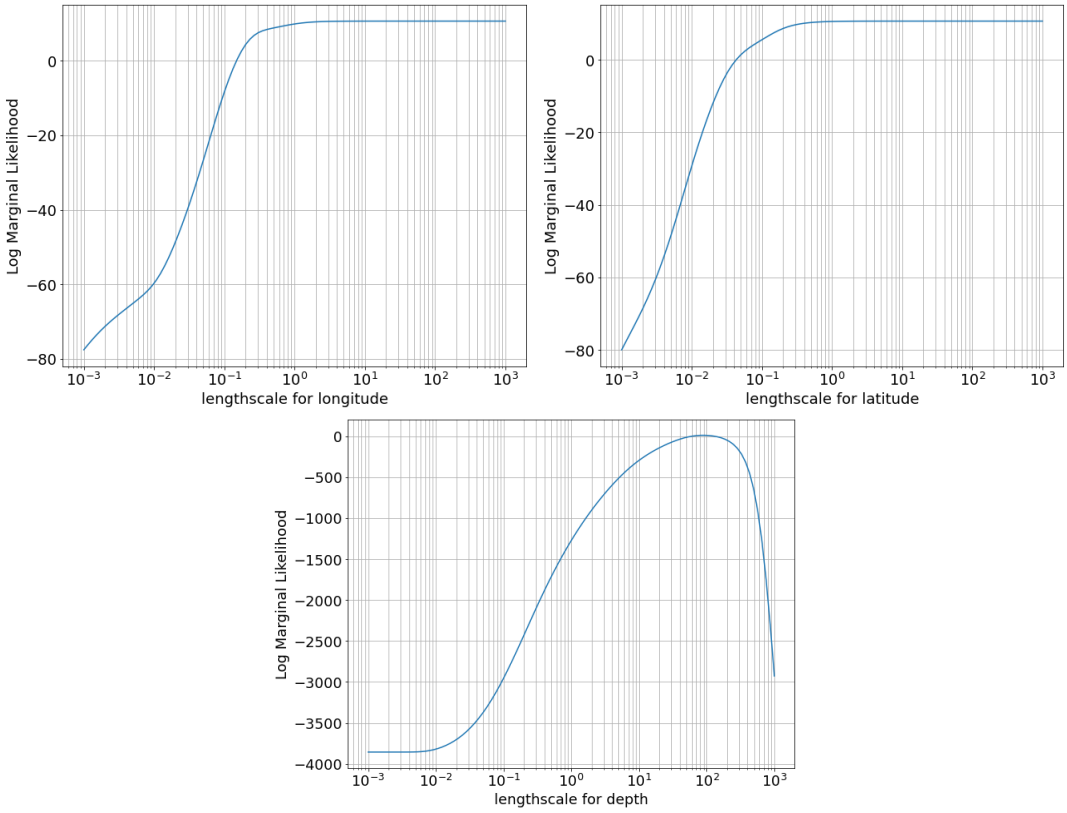


Figure B.1: Log marginal likelihood versus the lengthscale of the variables longitude, latitude and depth (log scaled x-axis).

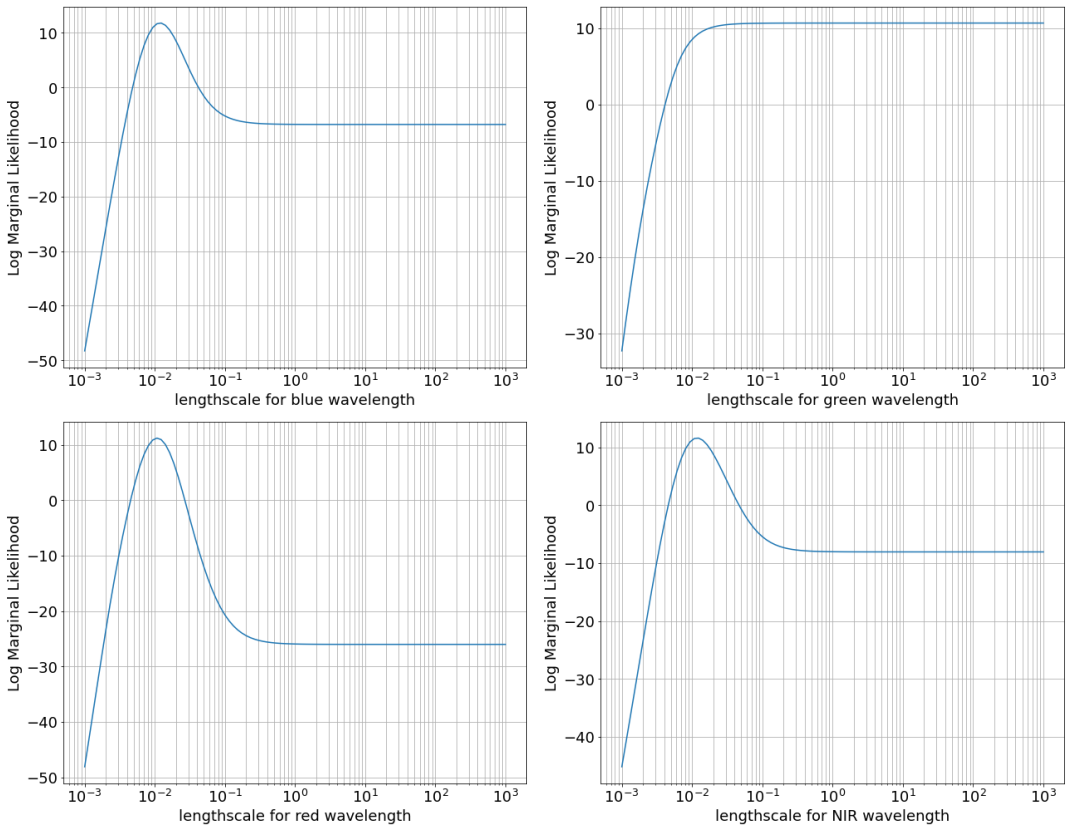


Figure B.2: Log marginal likelihood versus the lengthscale of the four reflectances (log scaled x-axis).

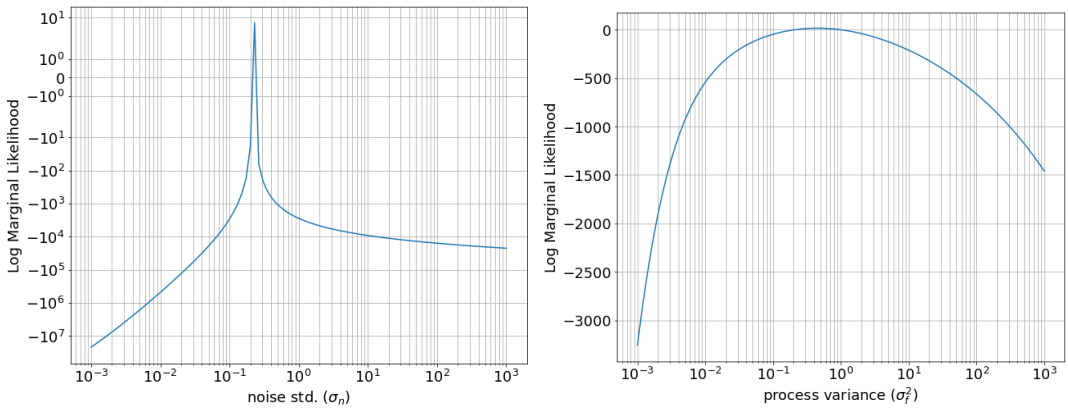


Figure B.3: Log marginal likelihood versus the hyperparameters σ_n and σ_f (log scaled x-axis and symlog scaled y-axis for σ_n).