Doctorado en Estadística y Optimización
Facultad de Matemáticas

Approximate Gaussian Processes and Derivative Information for Spatio-Temporal Regression and Classification

---

Gabriel Riutort Mayol Enero 2020

Directores:
Michael Riis Andersen
Virgilio Gómez Rubio
José Luis Lerma García

VNIVERSITAT
DE VALÈNCIA


UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA


GOBIERNO
DE ESPAÑA
MINISTERIO
DE CIENCIA, INNOVACIÓN
Y UNIVERSIDADES

Preface

This doctoral thesis is the result of my research work that has been carried out at the Dept. of Cartographic Engineering, Geodesy and Photogrammetry (DICGF) of the Universitat Politècnica de València (UPV) under the doctoral program of Statistics and Optimization of the Faculty of Mathematics of the Universitat de València. I gratefully acknowledge the financial support provided by the Agencia Estatal de Investigación from the Ministerio de Ciencia, Innovación y Universidades, with grants BES-2015-073476 and HAR2014-59873-R leaded by Prof. José Luis Lerma García.

First, I would like to thank my supervisors Prof. Michael Riis Andersen from the Dept. of Applied Mathematics and Computer Science of the Technical University of Denmark, Prof. Virgilio Gómez Rubio from the Dept. of Mathematics of Universidad Castilla La Mancha and Prof. José Luis Lerma García from the DICGF of the UPV for their dedication and support and for providing insightful comments on my research during all these years, as well as Prof. Antonio López Quílez from the Universitat de València.

I am specially grateful to Prof. Aki Vehtari from the Dept. of Computer Science at Aalto University for the possibility to visit the department as a visiting researcher, work with his team at the Probalistic Machine Learning (PML) group, and for giving me the opportunity to get such useful insights from him and his group, which have been essential for my research work and knowledge. Here, I also want to warmly thank my colleagues from the PML group, Eero Siivola, Topi Paananen, Juho Piironen, Federico Pavone, Olli-Pekka Koistinen, Kunal Ghosh, Akash Dhaka, Will Wilkinson, Tuomas Sivula, Måns Magnusson, Alejandro Catalina, Marko Järvenpää. And I would also like to specially thank Prof. Arno Solin from Aalto University since his superb work has been the basis of a part of the present work, and also Paul

Bürkner from the PML for his valuable comments on part of the manuscript and making it usable implementing it in the R-package *brms*.

I wish to thank Prof. Jorge Padin Devesa, director of the DICGF, for his support, my colleague and friend Prof. Ángel Marqués Mateu from the DICGF for our extensive and constructive work discussions but especially for his friendship, and also my friend Prof. Fernando Polo Garrido from the Dept. of Economy and Social Sciences of the UPV.

I am also thankful to my colleagues and friends Eric Gielen, Sergio Palencia, Yaiza Perez, Asenet Sosa y María José Aguilar from the Urban Planning Dept. of the UPV. And last but not less, I would like to thank to my group colleagues at the DICGF, Inés Barbero, Silvia Blanco, Berta Carrión and Adolfo Molada.

Finally, I want to express my gratitude to my family, María José and Biel, parents and brother for supporting me during these years and for always being there for me.

to Mam and Biel

Contents

Chapter 1

Introduction

The Bayesian paradigm is a framework to perform statistical data modeling and inference based on probabilistic theory and Bayes' theorem. This thesis lies between methodological aspects and applications in real world problems of the Bayesian approach. Regarding the applications, we consider two real world and novel applications by using mainly Bayesian hierarchical models and Gaussian processes (GPs), and ultimately gain insights into these applied fields: an *application to rock art paintings* of spatio-temporal modeling and prediction of microfading spectrometry (MFS) measurements, and an *application to image sensor noise* of decomposing and estimating noise sources in image sensors. Furthermore, it is worth noticing that, although it has been more briefly developed in the present work and mainly focused on the statistical modeling, we tackle a classical task of great interest in the field of remote sensing and environmental geo sciences for *spatio-temporal land use classification*.

Regarding the methodological aspects, we make a contribution on the issue of the high computational cost of performing inference on exact GPs when the number of observations is large. We conduct an study, analysis and implementation of a novel method, originally and theoretically developed by Solin and Särkkä [2018], to approximate GP models and making them faster to compute using sampling methods, especially in low dimensional input spaces. This approximate GP model is of application on the spatio-temporal land use classification task performed in this work, where a large number of available observations makes difficult the applicability of exact GPs, and thus make a contribution on the applied remote sensing field.

The other methodological aspect tackled in this work is the analysis of the issue of obtaining overly smoothed posterior distributions when using virtual derivative observations to induce monotonicity in functions, especially with GPs. This problem

arises in the application to rock art paintings where monotonicity constraints have to be implemented in order to ensure that the predicted functions are monotonically increasing.

Statistical data modeling is the appropriate framework to solve many of the problems that involve measurements, especially those which are complex and with large datasets. Among others, the problems typically include discovering relationships and/or patterns in the data, predicting new observations, classifying relevant features, decomposing variability sources or reconstructing missing values. In order to properly perform those problems, the underlying process must be modeled accurately by making reliable assumptions on the model, parameters of that model and structures of dependency among parameters. Furthermore, the models must be scalable and the computational methods efficient, since often the datasets are complex and large.

The Bayesian approach uses and formulates probabilistic models following the probability theory rules to perform modeling and make inferences consistently from data. Bayesian inference is the process of fitting a probabilistic model to a set of data, learning unknown parameters of that model and making inferences for unknown quantities and predictions for new observations. Probability distributions are specified either for observed quantities and unknown quantities. Inferences are performed through probability distributions which completely characterize the location and uncertainty of those quantities.

The main limitation in the Bayesian modeling approach is that inference has analytic solutions only for trivial models, such as Gaussian models with conjugate priors, and it is analytically intractable for the most relevant models. Different approximate computational methods, with different approximation accuracy, are available to perform Bayesian inference. Sampling methods, such as Markov chain Monte Carlo (MCMC) or Hamiltonian Monte Carlo (HMC), are the most natural and accurate methods and probably the most commonly used. However, sampling methods have the drawback of being computationally highly demanding, especially when the number of observations is large or when the number of unknown parameters is large with strong posterior correlations which cause slow or even inconsistent convergence of the sampling chains. In order to overcome this, there is a need for re-parameterizing the models or formulating approximate models with better scaling properties and with desirable similar performances.

On the other hand, real world processes can be complex, for instance, their dynamics in time and/or space may change or discontinuities may appear. Constructing flexible models and including additional information to the models in order to make them more flexible and realistic is essential to model complex features of real-world datasets.

Although Bayesian inference is well and broadly established in many scientific fields such as epidemiology, computer science, robotics and astronomy, there are still many other areas where this fully probabilistic framework for data analysis and inference is not well-known, tested or used. Some of those areas are the ones we enumerated at the beginning of this introduction. First, in application to prehistoric archeological rock art paintings, we perform a MFS analysis of specimens in the field of cultural heritage, where the use of advanced and flexible statistical models are unusual and the Bayesian point of view has not been used yet. We propose two modeling approaches, a spatio-temporal GP model and a spatially correlated splines-based time-series model, with the inclusion of monotonicity and gradient constraints. The goal is to predict MFS measurements, that represent potential color degradation over time, for the whole of unobserved locations on the surface of rock art paintings. Second, in application to image sensor noise, we work on the image sensor calibration field in which the Bayesian methodology is basically unknown in terms of the current applicability. We propose a novel application of Bayesian hierarchical modeling to decompose and characterize the noise components involved in the image sensing process. And finally, in the remote sensing field, most of the models used for classification are based on classical statistics or neural networks which hardly model spatio-temporal structures present in the data. Furthermore, datasets are usually very large, which prevents from computational sampling methods that allow for formulating flexible models and performing accurate inferences. So, in this work, we tackle the land-use classification task by formulating a spatio-temporal GP model for classification using the approximate GP model introduced in Chapter 3, which allows for dealing with much larger datasets than regular GPs.

One of the major advantages of the Bayesian approach is the fully propagation of uncertainty throughout the probabilistic model. The Bayesian approach propagates uncertainty to other quantities that are unknown in the model, which allows us to model certain values of some parameters or attributes of the model, such as degrees of freedom of the model, smoothness coefficients or other quantities. Although guesses on their exact values are not required, assumptions have to be made for their distributions. The Bayesian framework uses the property of defining conditional dependencies among quantities to perform hierarchical modeling which allows for specifying powerful models with complex structures. Bayesian hierarchical modeling is an excellent example of propagating uncertainties among different quantities.

The models considered in this thesis belong to the class of hierarchical models since they have some levels of dependencies among parameters of interest. These models are:

- Gaussian processes for regression and classification, which are used in application to rock art paintings and in the spatio-temporal land use classification task, respectively.
- An additive multilevel random effects linear model, which is used in application to image sensor noise.
- A spatially correlated splines-based time-series model, which is used in application to rock art paintings.

Functional data analysis assumes that there is a functional description for the process under study and the observations are noisy realizations of this underlying function. GPs are flexible non-parametric prior distributions for multivariate functions. GP priors can be used to specify prior assumptions on the underlying function that describe the underlying relationships between inputs and response variables. GP is a fully non-parametric model, so the functional form is determined by the data instead of being fixed with parametric or semi-parametric forms. Furthermore, implicit assumptions on the GPs can be given by specifying its mean and covariance functions. The key element of a GP is the covariance function that encodes the prior assumptions about the correlation structures of function values, determining, for example, the smoothness/wiggliness and variability of the function. Due to their generality and flexibility, GPs are of broad interest in machine learning and statistics, with a wide range of applications, such as regression and classification, density estimation, dimension reduction, and spatio-temporal statistics.

The main limitation of the implementation of GPs in practical applications is their computational demands as they scale, in a direct implementation, as $O(n^3)$, with n being the number of observations. This problem becomes more severe when performing full Bayesian inference through sampling methods, where in each sampling step we need to invert a Gram matrix of the covariance function which is an $O(n^3)$ operation. Several methods have been proposed to alleviate this, such as sparse approximation to GP, compactly supported covariance functions, variational inference for GPs and basis functions approximation to GP. In this thesis, we make a contribution to the class of basis function approximation to GP by performing a study, analysis and implementation of the recently developed Hilbert space method to approximate GP by Solin and Särkkä [2018]. We analyze the performance of the method in relation to the key factors of the method, and implement the methodology in a fully probabilistic programming framework such as Stan.

On the other hand, in modeling problems of learning stochastic functions from data, there is often a priori knowledge and/or additional information available concerning the function to be learned, which can be used to improve the performance of the modeling. This information can be sometimes expressed in terms of the derivatives of the functions, so the dynamics of the functions can be controlled or

constrained, e.g. increase, decrease and stabilization of the function. In particular, since differentiation is a linear operator, the derivative of a GP is still a GP, as well as the derivative of a linear parametric or semiparametric model is also linear. This makes it possible to include derivative information in the modeling by jointly modeling the regular process and its derivative process using GPs and semiparametric models. However, some inference issues can arise with this approach of using derivative observations to induce monotonicity on functions, causing overly smoothed posterior functions when many derivative observations for monotonicity are used. In this thesis, we make a contribution by analyzing and revealing this issue.

This work consists of eight chapters that describe the goals, techniques and discoveries of the research.

In Chapter 2, a short but complete overview of basics of Bayesian data analysis and inference is given. We give an overview of the foundations of probability theory for Bayesian inference, the rule of prior information and model assumptions. We give brief reviews of the main and most commonly used models: linear generalized models, non-parametric models and non-linear additive models, with special emphasis on the flexibility and usefulness of the Bayesian hierarchical models. We make a map of the different computational methods for performing inference in Bayesian models. We put emphasis on the sampling methods based on MCMC and HMC, which are those used through this work to numerically approximate the required integrals in the Bayesian approach. Finally, a brief overview of the methodology to perform model assessment, validation and selection is given.

In Chapter 3 a Hilbert space method to approximate GPs, originally and recently developed by Solin and Särkkä [2018], is methodologically introduced and analyzed, and implemented in a probabilistic programming framework such as Stan. The method has an attractive computational cost as this basically turns the regular GP model into a linear model, which is also an appealing property in modular probabilistic programming models, e.g. Stan, WinBUGS and others. First of all, the exact GP model is described in detail and its main elements of the covariance and spectral density functions are derived. After that, we perform the study of the Hilbert space method to approximate GPs, analyzing the performance of the method in relation to the key factors of the method, seeking to develop useful procedures to make a diagnosis of the approximation accuracy, and ultimately make recommendations for the values of the key factors to improve performance. Several illustrative case studies, simulate and real datasets, where we demonstrate the performance, the applicability and the implementation of the methodology, are carried out in this chapter. Among these case studies, one is dedicated to the spatio-temporal land-use classification task of classifying the land use of parcels dedicated to growing citrus fruits. This consists of a large dataset with multivariate predictors

derived from satellite images. Furthermore, in Appendix A, additional case studies assessing and illustrating the performance of the method are also presented. The model codes for software Stan of case studies are provided for both exact GP and approximate GP models through links to the author's GitHub repository.

In Chapter 4, we illustrate the usage of derivative information as additional (virtual) observations in two modeling approaches, a GP model and a penalized splines model. The consideration of derivative information in the modeling can be used to control the dynamics of the functions. We illustrate the main issue of this approach to induce monotonicity on functions, that can produce overly smoothed posterior functions especially when many derivative observations for monotonicity are used.

In Chapter 5, the application to rock art paintings of modeling and predicting MFS color fading time-series for new unobserved spatial locations on the surface of rock art paintings is carried out. Apart from constructing a model that exploits to the full the correlation structure of the data in a scenario of a short set of sampling observations, the main motivation of this study is the consideration of monotonicity and gradient constraints in the modeling aiming to overcome the large fluctuations in the data and fit the desired properties of monotonicity and stabilization in the long term for the predicted color-fading time-series.

In Chapter 6, the application to the image sensor is carried out. A novel approach based on Bayesian hierarchical modeling to decompose the signal recorded by an image sensor into its different noise sources is proposed. We argue that a Bayesian multilevel random effects model is a flexible and highly interpretable model, which also allows for naturally and accurately propagating uncertainty among noise parameters in comparison with the existing standards for image noise measurements.

In the novel and real world application cases related to rock art paintings, to image sensor noise and to spatio-temporal land-use classification, the performance of the models is assessed in order to describe the quality of the model. The predictive performance of the model for future observations is assessed by estimating the mean square error and the likelihood expected utility such as the expected log predictive density. We use cross-validation to approximate the mean square error and the expected utilities.

Finally, in Chapter 7 we give an overall conclusion of the work, goals, techniques, findings and future research lines.

1.1 Summary of publications/submissions

We have worked on the writing of different scientific publications reporting the main research findings of this work.

The research contents developed in Chapter 3 titled *Hilbert space methods to approximate Gaussian processes*, have been included in the following research article that is intended to be sent for publication in the *Journal of Statistical Software*:

> Riutort-Mayol, G., Andersen, M. R., Bürkner, P., and Vehtari, A. (2020). Hilbert space methods to approximate gaussian process using Stan. *Journal of Statistical Software*. Submitted.

The applied study carried out in Chapter 5 titled *Application to rock art paintings: Models with derivative information for modeling microfading spectrometry measurements*, have been reported in the following two publications which are already available in scientific repository *arXiv*:

> Riutort-Mayol, G., Andersen, M. R., Vehtari, A., and Lerma, J. L. (2019). Gaussian process with derivative information for the analysis of the sunlight adverse effects on color of rock art paintings. *arXiv preprint arXiv*:1911.03454.
>
> Riutort-Mayol, G., Gómez-Rubio, V., Lerma, J. L., and del Hoyo-Meléndez, J. M. (2019). Correlated functional models with derivative information for modeling MFS data on rock art paintings. *arXiv preprint arXiv*:1910.12575 .

In addition, related to the work done in Chapter 5, two other publications have been developed, one of them is already published and the other one is already submitted for publication:

> del Hoyo-Meléndez, J. M., Carrión-Ruiz, B., Riutort-Mayol, G., and Lerma, J. L. (2019). Lightfastness assessment of levantine rock art by means of microfading spectrometry. *Color Research & Application* 44, 547–555.
>
> Carrión-Ruiz, B., Riutort-Mayol, G., Molada-Tebar, A., Lerma, J. L., and Villaverde, V. (2019). Color degradation mapping of rock art paintings using microfading spectrometry. *Archaeological and Anthropological Sciences*. Under review.

The applied study carried out in Chapter 6 titled *Application to image sensor noise: Hierarchical modeling for estimating noise in image sensors*, have been included in a research article which is already submitted for publication:

> Riutort-Mayol, G., Gómez-Rubio, V., Marqués-Mateu, A., Lerma, J. L., and López-Quílez, A. (2019). A Bayesian multilevel random-effects model for estimating noise in image sensors. *IET Image Processing*. Under review.

Chapter 2

Bayesian inference

This chapter gives a brief overview of the standard workflow of Bayesian data analysis and inference, as a base framework for all the models, algorithms and analysis made in this work. The contents of this overview covers the theoretical basis of Bayesian inference, the most commonly used Bayesian models and computational methods, and the analysis of model validation and assessment. The Bayesian models used in this work go from generalized linear models, passing through non-parametric modeling based on Gaussian processes and splines models, additive Gaussian processes and hierarchical models. A brief overview of the most commonly used computational strategies in Bayesian inference, making special emphasis on sampling methods based on Markov chain Monte Carlo, which are the ones used throughout all this work, is presented. Finally, the methods for model checking, validation and assessment used in this work are briefly described.

2.1 Fundamentals of Bayesian probability

Bayesian probability [Bernardo and Smith, 2009, Gelman et al., 2013] is a theoretical framework for inference of unknown quantities when uncertainty is on the premises. The Bayesian inference process of updating beliefs of certain quantities when new information is observed relies on Probability theory [Cox, 1946, Jaynes, 2003, Jeffreys, 1961, Pearl, 1988]. The representation of belief of random variables is through probability distributions. Given a model describing mutual dependencies of random variables, Bayesian probability theory can be used to infer all the unknown quantities. All uncertainties, either in observations and model parameters, are modeled as probability distributions.

Bayesian and frequentist paradigms

In statistics there are two major paradigms for inference: *frequentist* and *Bayesian* paradigms.

In *frequentist* statistics, probability has to be seen as frequency of occurrence of events, considering these events as in a process having intrinsic randomness[1]. Probabilities are only assignable to events or outcomes of an intrinsic random process, so parameters governing a phenomena are considered as fixed values. Frequentist approach only refers to aleatory uncertainty of events which is the intrinsic aleatory uncertainty of a random process, and there is no way to reduce this uncertainty with new observations since it is intrinsic to the process.

In *Bayesian* statistics, probability provides a quantification of uncertainty of events described as probability distribution [Cox, 1946, De Finetti, 2017, Jaynes, 1985]. Probabilities are assignable either to events of an intrinsic random process or parameters governing a phenomena. Thus, the Bayesian approach contemplates aleatory uncertainty and epistemic uncertainty [O'Hagan and Forster, 2004, O'Hagan, 2004]. Epistemic uncertainty refers to uncertainty due to lack of knowledge of something that is not intrinsically random, so it can be knowable through new observations, such as the parameters governing a phenomena. If we use probability distributions to define epistemic uncertainty, then we become Bayesian.

The Bayesian approach describes prior knowledge about the parameters governing a phenomena through probability distributions. New knowledge about the parameters governing a phenomena is provided by new observed data described by the likelihood function, which is the probability distribution of the observed data conditioned on the parameters governing the phenomena. Through Bayes' theorem, prior probability distribution of the parameters governing the phenomena is updated with the observed data likelihood function, obtaining a posterior probability distribution for the parameters governing the phenomena.

Probability density

A probability density function is used to characterize continuous random variables. Thus, if x is a random variable with a probability density function $p(x)$, the probability of the event that x is in the interval (a, b) can be computed as:

$$p(x \in (a, b)) = \int_a^b p(x)\mathrm{d}x.$$

[1]The frequentist probability of an event is the limit of its relative frequency of occurrence when the experiment is repeated in a very large number of times [Bickel and Lehmann, 2012].

In addition, the probability density function $p(x)$ must satisfy the two conditions:

$$p(x) \geq 0,$$
$$\int_{-\infty}^{\infty} p(x)\mathrm{d}x = 1.$$

Given another continuous variable, for example y, $p(x, y)$ denotes the joint probability density function of the two variables, and $p(y|x)$ denotes the conditional probability distribution function of the variable y given the variable x.

In this work, the terms *density* and *distribution* are used interchangeably, and we use the letter p to denote them. The same notation is used for probability density of continuous variables and probability mass for discrete variables. We sometimes can use the notation of $z \sim \mathcal{D}(b)$ as a short hand for $p(z) = \mathcal{D}(z|b)$, where b denotes the model parameters of distribution $\mathcal{D}$, and z denotes a random variable.

The sum and product rule

The sum and product are the two fundamental rules [Cox, 1946, Jaynes, 2003] in probability theory, which, for probability densities, take the form:

$$\text{Sum rule} \qquad p(x) = \int p(x, y)\mathrm{d}y,$$
$$\text{Product rule} \qquad p(x, y) = p(y|x)p(x) = p(x|y)p(y).$$

The probability of x, $p(x)$, is sometimes called the marginal probability of the variable x, because it is obtained marginalizing, or integrating out, the variable y. The product rule specifies that the joint probability distribution of two variables can be expressed as the product of a conditional distribution $p(x|y)$ and a marginal distribution $p(x)$, or vice-versa. A formal justification of the sum and product rules for continuous variables can be found in Feller [2008].

The Bayes' theorem

From the product rule, and with the symmetry property $p(x|y)p(y) = p(y|x)p(x)$, we immediately derive the Bayes' rule [Bayes, 1763]:

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} = \frac{p(x|y)p(y)}{\int p(x|y)p(y))\mathrm{d}y},$$

which is the key element in Bayesian inference [Bernardo and Smith, 2009], since it defines the posterior density of y, $p(y|x)$, after including new information of

x through the conditional probability model $p(x|y)$. The marginal probability of x, $p(x)$, makes of a normalization constant for the numerator, becoming a proper probability density function.

The marginalization principle

The marginalization principle comes from the sum rule in probability theory. The marginalized principle formalizes the generalization or predicting the capacity of a learning system.

If we can specify the product rule for two related quantities (z, v),

$$p(z, v) = p(z|v)p(v),$$

that one can be explained by the other through the likelihood function $p(z|v)$, a generalization or prediction of the unknown z can be obtained by integration out over all the different explanations v:

$$p(z) = \int p(z|v)p(v)\mathrm{d}(v).$$

The likelihood function $p(z|v)$ gives the probability of the unknowns for a particular explanation, and $p(v)$ gives the weights for every possible explanation.

2.2 Bayesian modeling and inference

Joint probability distribution

Bayesian modeling consists in describing in a mathematical form all observable (data), y, and unobservable (parameter), θ, quantities in a problem, through defining the joint probability distribution of data and parameters.

We define probability models for the observed quantities, $p(y|\theta)$, and unobserved quantities about we wish to learn, $p(\theta)$, and combine them through the product rule in a joint probability distribution:

$$p(y, \theta) = p(y|\theta)p(\theta).$$

The observational model $p(y|\theta)$ is a probabilistic model for the observed data that relates the observed data y with the unknown quantities (parameters) θ we want to learn. This model represents the evidence provided by the data, summarizes the information from the data. It is the main source of information and is called likelihood function. This is the same as in the frequentist approach. Actually, it is

the unique probabilistic model formulated in a frequentist approach to describe and solve a problem.

The distribution $p(\theta)$ denotes a prior probability distribution for the parameters, that encodes our prior knowledge about the parameters. This probability distribution can be an informative or non-informative prior distribution, depending on the reliable information (knowledge) available for the parameters. This is one of the key features that differentiate from the frequentist approach, i.e. probability distributions are defined for the unknown quantities (parameters) and combined with the likelihood function.

Parameter inference

Obtaining the posterior distribution of the unknowns (parameters) is the key element of the Bayesian approach. Through Bayes' rule, the likelihood function (probability model for the data) and prior distributions for the parameters are combined, and the uncertainty in the parameters once the data have been observed is updated, obtaining the posterior distribution of the parameters:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)\mathrm{d}\theta}.$$

The denominator of Bayes' rule, $p(y) = \int p(y|\theta)p(\theta)\mathrm{d}\theta$, is the marginal likelihood, as it integrates the likelihood over the prior information of parameters, also known as the evidence of the model. The marginal likelihood normalizes the posterior into a proper probability distribution.

The final inference will be a compromise between the evidence provided by the data and the prior information. With non-informative priors, the inference would be based mainly on the data.

Predictive inference

The posterior distribution of the parameters $p(\theta|y)$ can be used to model the uncertainty of predictions $\tilde{y}$ for new observations. In a Bayesian approach, the posterior predictive distribution of $\tilde{y}$ is obtained by marginalizing or integrating out the joint posterior of predictions $\tilde{y}$ and model parameters θ over the model parameters:

$$p(\tilde{y}|y) = \int p(\tilde{y}, \theta|y)\mathrm{d}\theta = \int p(\tilde{y}|\theta, y)p(\theta|y)\mathrm{d}\theta.$$

The predictive distribution can also be seen as averaging the predictions of the model $p(\tilde{y}|\theta, y)$ over the posterior distribution of the model $p(\theta|y)$.

Brief comments about priors in the unknowns

One of the main distinguishing things in the Bayesian approach is the consideration of prior knowledge about the model parameters.

The Bayesian approach allows for performing consistent inferences even when the prior information is lacking, by marginalizing or integrating out over this prior information. This property also allows us not to make guesses on certain unknown quantities, in contrast to classical methods. Furthermore, the Bayesian approach, by its property of defining conditional dependencies among parameters and model assumptions in hierarchical modeling, allows for defining lack of prior information in an appropriate way.

However, the use of uncertainty assumptions makes the Bayesian approach to be more sensible to prior assumptions than classical methods.

2.3 Models and priors

2.3.1 Generalized linear models

The term *generalized* refers to specifying different observational models from the exponential family, denoted by F, for the observations y. The models are linked to a predictor function η by a specific link function $g(\cdot)$,

$$\begin{aligned} p(y|\eta, \phi) &= F(y|\mu, \phi), \\ g(\mu) &= \eta, \end{aligned}$$

where μ represents the mean of the model, $\mathbb{E}[y|\phi] = \mu$, and ϕ represent the other parameters of the model, such as, for example, the variance parameter in normal models or the shape parameter in gamma models. The predictor function η is usually linked to the mean parameter μ of the model (although it might also be another parameter of the model) by a strictly monotonic link function g with inverse mapping $\mu = g^{-1}(\eta)$. Often, g is a differentiable function in order to be able to obtain the maximum likelihood estimate (see Section 2.4) conveniently.

The term *linear model* usually refers to using a parametric form for the functional relationship (predictor function η) between observed data and predictors (input variables), such that:

$$\eta = \boldsymbol{\beta}\boldsymbol{x},$$

where $\boldsymbol{\beta}$ is the row-vector of coefficients in a parametric model and $\boldsymbol{x}$ is the column vector of input values (in this work, the terms *input variable*, *covariate* and *predictor* are used interchangeably). In parametric models the form of predictor functions are

defined by a finite, often small, set of parameters $\boldsymbol{\beta}$. Parametric models fit possible nonlinear effects through, for example, binning or polynomials.

In the Bayesian framework, prior distributions have to be defined for the model parameters μ and ϕ. In the case of the mean parameter μ, if we have a predictor function, we will define the priors in the parameters β of the predictor function instead.

Normal model

The observational model is a Normal distribution, denoted by $\mathcal{N}$, of parameters mean μ and noise variance σ^2:

$$\begin{aligned} p(y|\mu,\sigma) &= \mathcal{N}(y|\mu,\sigma^2), \\ \mu &= \eta. \end{aligned}$$

In this case, the canonical link function is the identity function.

Poisson model

The observations are expected to follow a Poisson distribution, denoted by $\mathcal{P}$, with mean parameter μ:

$$\begin{aligned} p(y|\mu) &= \mathcal{P}(y|\mu), \\ \log(\mu) &= \eta. \end{aligned}$$

In this case, the link function is, for example, the log function, in order to transform the values of the predictor function η, usually in the continuous real space, to the strictly positive or equal to zero range of values of the mean of the Poisson model.

Binomial model

In this case, the observations are binary-valued (0,1) observations. These binary observations are expected to follow a Binomial distribution, denoted by $\mathcal{B}$, with probability parameter p of being 1:

$$\begin{aligned} p(y|p) &= \mathcal{B}(p), \\ \text{logit}(p) &= \eta. \end{aligned}$$

In this case, the probability p is linked to the predictor function η through the 'logistic' transformation, logit$(\cdot)$, which transforms the values of the predictor function, usually in the continuous real space, to the [0,1] range of probabilities.

In this model, the 'probit' link function can also be used [Aldrich et al., 1984]. A Binomial observational model is used in the case studies in Sections 3.10 of Chapter 3 and A.1 of Appendix A.

Multinomial model

In multi-class classification problems, the observations are multi-class-valued $(1, \ldots, J)$. In this case, a multinomial observational model, denoted by $\mathcal{M}$, may be used:

$$p(y) = \mathcal{M}(\boldsymbol{p}),$$

where $\boldsymbol{p} = (p_1, \ldots, p_j, \ldots, p_J)$ is a vector of probabilities of each possible class. In this model, in order the vector of probabilities $\boldsymbol{p}$ of an observation to sum to 1, $\sum_{j=1}^{J} p_j = 1$, the following constraint has to be considered:

$$p_J = 1 - \sum((p_1, \ldots, p_{J-1})).$$

The probability of belonging to a class j can be computed by the 'softmax' transformation [Bishop, 2006]:

$$p_j = \frac{\exp(\eta_j)}{\sum_{k=1}^{J} \exp(\eta_k)}.$$

where η_j denotes the predictor function for modeling the probability of belonging to class j. A multinomial observational model is used in the case study in Section 3.11 in Chapter 3.

2.3.2 Conjugate priors

A prior distribution $p(\theta)$ is said to be conjugate to a likelihood function $p(y|\theta)$ if the posterior distribution $p(\theta|y)$ has the same functional form as the prior distribution [Bernardo and Smith, 2009, Gelman et al., 2013]. The use of conjugate priors for the likelihood produces posterior distributions in a closed form, then makes it possible to solve the posterior analytically.

Conjugate priors of exponential family For any likelihood function of the exponential family, a conjugate prior distribution always exists. That is the reason of the importance of the exponential family distributions, that in simple models and with the use of conjugate priors, the posterior has analytic solution, which is known as the *conjugate-exponential* model (Ghahramani and Beal, 2001).

2.3.3 Non-parametric models

The term *non-parametric* means that the shape of the predictor functions are fully determined by the data as opposed to parametric functions that are defined by a typically fixed set of parameters. In non-parametric models the number of parameters grows with the number of data.

Non-parametric functions are extremely flexible since the shape adapts to the underlying patterns in the data, either linear or nonlinear, smooth or wiggly, without knowing what these patterns look like. This property may be useful to find unknown patterns in the data, in contrast to a parametric model.

However, in parametric models, selecting the best model involves constructing a multitude of models with different forms, parameters and covariables, in the predictor, followed by a search algorithm to select the best option, which can be a potentially greedy step.

Non-parametric models are commonly based on kernel functions [Rasmussen and Williams, 2006, Shawe-Taylor et al., 2004], such as the case of Gaussian processes, which are extensively used along this work. Basically, a *kernel* function $k(\boldsymbol{x}, \boldsymbol{x}')$ is a function that maps a pair of inputs $\boldsymbol{x}$ and $\boldsymbol{x}' \in \mathcal{X}$ (with input domain $\mathcal{X} \subset \mathbb{R}^D$) into $\mathbb{R}$ characterizing the similarity of the pair of inputs [Shawe-Taylor et al., 2004]. Semi-parametric models based on series expansion of basis functions are usually referred to as non-parametric models. In this work, we use splines models in Chapters 4 and 5.

2.3.3.1 Gaussian processes

Chapter 3.4 of this thesis is specifically dedicated to Gaussian processes (GPs). In the present section, we just briefly introduce them.

GP is a non-parametric model, i.e. an infinite parametric kernel model. GP can be used as a prior probability distribution for multivariate and non-linear functions $f(\boldsymbol{x}) : \mathbb{R}^D \to \mathbb{R}$, and has the defining property that any finite collection of random variables are multivariate Gaussian distributed [Rasmussen and Williams, 2006]. A GP model is completely characterized by its mean and covariance function. The key element of a GP is the covariance function as it defines the correlation structure of function values. A covariance function can be expressed as a positive semidefinite kernel function, such that the Gram matrix corresponding to the covariance function is positive semidefinite [Rasmussen and Williams, 2006]. Several covariance functions can be used, from stationary (e.g. exponentiated quadratic, Matern) to non-stationary (e.g. dot product, neural network) functions, which can also be combined for further increased flexibility [Duvenaud et al., 2013, 2011, Rasmussen and Williams, 2006]. For a review of the different covariance functions in Gaussian

processes, see Rasmussen and Williams [2006]. Due to their generality and flexibility, GPs are of abroad interest across machine learning and statistics [Neal, 1999, Rasmussen and Williams, 2006].

2.3.3.2 Basis function models

Finite parametric non-linear models are basically based on the class of additive models of non-linear functions [Ruppert et al., 2003, Wood, 2017]. The common basis function models rest on the series expansion of basis functions. In the one dimensional case, the predictor of a basis function model approach can be illustrated as follows:

$$\mu(x) = \sum_k Z_k(x) b_k \tag{2.1}$$

where $b = (b_1, \ldots, b_K)$ is the vector of coefficients and $Z = \{Z_k(x)\}_{k=1}^K$ is the set of basis functions. A simple example is the Taylor series expansion in which the basis functions are polynomials of increasing degree. This scheme allows the mean to vary nonlinearly as a function of the predictors. The weighted sum of the basis functions can model non-linear and smooth functions.

Useful options for the form of the basis function, which allows for more flexible and accurate relationships, are using natural cubic-splines, B-splines, radial splines, etc. The flexibility of the model depends on the number K of basis functions, so the more basis functions the more flexible functions are obtained. In order to avoid overfitting, different approaches can be used, such as the use of a prior on the number and locations of knots in a kernel or spline model, the use of shrinkage prior on the splines coefficients, similarly to variable selection procedures [Piironen et al., 2018], or the use of a penalized version of these models, which are probably the most commonly used [Crainiceanu et al., 2005, Wood, 2003]. In this thesis, we illustrate in Chapters 4 and 5 the formulation of one-dimensional penalized Thin-plate cubic-splines [Ruppert et al., 2003] represented in the form of linear mixed model as presented in Crainiceanu et al. [2005].

Conditionally on the chosen basis functions, the model is linear in the parameters, then inference can be faced as in linear regression models.

2.3.4 Hierarchical models

The Bayesian framework uses the property of defining conditional dependencies among quantities to perform hierarchical modeling, which allows for specifying powerful models with complex structures [Gelman et al., 2013, Gelman and Hill, 2006, Ntzoufras, 2011].

The basic idea of hierarchical modeling, also known as multilevel models, is to organize the model using a set of statements of probability conditional dependencies among quantities and model assumptions. The joint probability should reflect this dependencies. Generally, a hierarchical structure can be written if the joint distribution of the parameters can be decomposed to a series of conditional distributions. The term hierarchical models refers to a general set of modeling principles than to a specific family of models.

Priors of model parameters are the first level of hierarchy. Priors of model parameters can depend on other parameters (prior parameters), and new priors (hyperpriors) are defined over these prior parameters. In this case, the hyperpriors would be the second level of hierarchy. Following this scheme [Ntzoufras, 2011], the structure can be extended to more levels of hierarchy. In principle, there is not a limited level of hierarchy.

Following, we illustrate in equation (2.2) the posterior distribution of a model parameter in a conditional structure with two-levels of hierarchy. The prior distribution of model parameter θ, $p(\theta|a)$, depends on the prior parameter a, which hyperprior distribution $p(a|b)$ depends on a fixed value b (hyper-parameter). Notice that, in equation (2.2), the posterior distribution is presented as proportional to the likelihood and priors, avoiding the normalizing constant of the marginal likelihood. In Figure 2.1, the direct acyclic graph of these hierarchical structure is depicted.

$$p(\theta|y) \propto p(y|\theta)p(\theta|a)p(a|b) \tag{2.2}$$

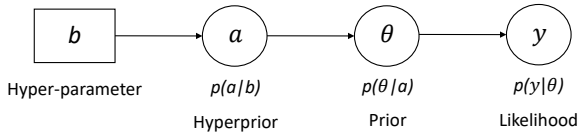


Figure 2.1: Graphical representation of a two-level hierarchical prior dependency.

This hierarchical structure usually arises in problems such as multiparameter models, where parameters can be regarded or connected in some way, or models in complex phenomena with different levels of hierarchy in the data, or models with covariance structure, where the covariance is governed by some other parameters, etc. Bayesian hierarchical modeling is an excellent example of propagating uncertainties among different quantities in complex models with conditional dependencies. Furthermore, Bayesian hierarchical modeling allows us not to make guesses on certain unknown quantities, in contrast to classical statistics.

The most common and simple view of a hierarchical structure arises when

model parameters come from a common population distribution. For example, a model with a common population distribution, $\mathcal{N}(0, \sigma_\mu^2)$, for the mean parameters, μ_i, of a Normal observation model, can be expressed as follows:

$$\begin{aligned} \text{Likelihood} \quad & p(y_i|\mu_i, \sigma_i) = \mathcal{N}(y_i|\mu_i, \sigma) \\ \text{Prior} \quad & p(\mu_i|\sigma_\mu) = \mathcal{N}(\mu_i|0, \sigma_\mu) \\ \text{Prior} \quad & p(\sigma|d) = \mathcal{J}_1(\sigma|d) \\ \text{Hyperprior} \quad & p(\sigma_\mu|b) = \mathcal{J}_2(\sigma_\mu|b) \end{aligned}$$

where $\mathcal{J}_1$ and $\mathcal{J}_2$ denote some prior distributions for the scale parameters σ and σ_μ, respectively, and b and d are specific-valued parameters for these prior distributions. The posterior distribution of that model takes the form:

$$\begin{aligned} p(\mu, \sigma, \sigma_\mu|y) &\propto p(y|\mu, \sigma)\, p(\mu|\sigma_\mu)\, p(\sigma_\mu)\, p(\sigma) \\ &= \mathcal{N}(y|\mu, \sigma)\, \mathcal{N}(\mu|0, \sigma_\mu)\, \mathcal{J}_2(\sigma_\mu|b)\, \mathcal{J}_1(\sigma|d). \end{aligned}$$

In exact Bayes, if we use a diffuse (improper) prior, we should check that the posterior is proper. In most problems, one should have enough substantive knowledge about the hyperparameters, at least to constrain them into a finite space. We can control the variation on the posterior of the hyperparameters with the prior.

In this work, we use sampling methods in all the applications, so we do not need to study the analytic solution of hierarchical models using conjugate priors. For a conjugate analysis of hierarchical model see e.g. Gelman et al. [2006].

In hierarchical models, it makes sense the generalization (predictive distribution) for future values of the parameters. And then, generalization for a future observation (outcome) corresponding to a future parameter value. For the example above, the predictive distribution of a future observation $\tilde{y}$ for a future predicted parameter value $\tilde{\mu}$, given the set of actual observations y is:

$$\begin{aligned} p(\tilde{y}|y) &= \int p(\tilde{y}|\tilde{\mu}, \mu, \sigma, y) p(\tilde{\mu}|\mu, \sigma_\mu, y) p(\mu|\sigma_\mu) p(\sigma_\mu) p(\sigma) \mathrm{d}\sigma_\mu \mathrm{d}\sigma \\ &= \int \mathcal{N}(\tilde{y}|\tilde{\mu}, \sigma) \mathcal{N}(\tilde{\mu}|\mu, \sigma_\mu) \mathcal{N}(\mu|0, \sigma_\mu) G(\sigma_\mu|b) F(\sigma|d) \mathrm{d}\sigma_\mu \mathrm{d}\sigma. \end{aligned}$$

Such a hierarchical thinking helps in understanding multiparameter models and is also useful for developing computational strategies. Hierarchical models can have enough parameters to fit data well in the case of big datasets. The use of prior probability distribution to structure some dependency into the parameters can avoid for overfitting.

Hierarchical models can be considered as a large set of stochastic formulations

that include many popular models such as the random effects, the variance components, the multilevel, the generalized linear mixed, spatial and temporal, and Gaussian processes. Models with random effects follow a hierarchical structure, since different parameters share a common distribution with a common variance, for example generalized random-effects model. Hierarchical models are widely used in meta-analyses [Woodworth, 2004]. Spatio, temporal and spatio-temporal models also are models that follow a hierarchical structure since common random effects are shared among neighbor variables, for example, first order random walk model, Kriging model, and Conditional Auto-Regressive Models for Areal Data [Banerjee et al., 2014, Gelman and Hill, 2006]. Models with Gaussian process priors for functions follow a hierarchical structure since function values are dependent on covariance function parameters [Gelman et al., 2013].

2.3.5 Generalized additive models

Widely known drawbacks of flexible non-parametric and multi-dimensional models are that they are often extremely difficult to interpret. If these flexible models are constrained to be additive over the input dimensions, the fitted models are much easier to interpret. In Generalized additive models (GAM) [Hastie, 2017, Larsen, 2015], the functional relationship between predictors and response variable is decomposed into a sum of low-dimensional non-parametric functions (and an additional fixed effects part model part if desired):

$$\mu(x_1, x_2, \ldots, x_J) = \boldsymbol{\beta x} + s_1(x_1) + s_2(x_2) + \cdots + s_J(x_J),$$

where the term $\boldsymbol{\beta x}$ denotes a linear term with the covariates $\boldsymbol{x} = (x_1, x_2, \ldots, x_J)$, and the terms $s_j(x_j)$ denote one-dimensional smooth non-parametric functions.

The problem with additive models is that they can be inaccurate if the phenomenon being modeled is not additive. A tradeoff between accuracy and interpretability can be achieved progressively constraining a fully flexible multidimensional model towards to be more and more additive with lower-dimensional functional additive components.

Low-dimensional functional components helps the interpretation of the model as marginal effects of a single component does not depend on the values of the other components, e.g. in the case of unidimensional functions, single input effects can be interpreted.

Nonparametric models can be extremely flexible as described in section 2.3.3. Some nonparametric models, for example GPs, control smoothness of the predictor functions to prevent overfitting, and tackle the bias/variance tradeoff. Following, we briefly discuss the use of GPs as individual components in an additive scheme.

2.3.5.1 Additive Gaussian processes

An additive GP model is a GAM model composed of GP functions . An additive GP model results in a GP model with a covariance function that is decomposed into the sum of lower dimensional kernels [Durrande et al., 2012, Duvenaud et al., 2011] (in this work, the terms *covariance function* and *kernel* are used interchangeably, although we know that not all kernels are covariance functions and that a kernel is a more general function [Rasmussen and Williams, 2006]). The new additive kernel allows additive interactions of all orders, ranging from first order interaction (one-dimensional kernels) all the way to multi-order interaction (multi-dimensional kernel). The additive kernels of 1st, 2nd, 3rd and dth order of interaction in a D-dimensional input space, $\boldsymbol{x} \in \mathbb{R}^D$, are computed as follows:

$$
\begin{aligned}
k_1(\boldsymbol{x}, \boldsymbol{x}') &= \sigma_1 \sum_{i=1}^{D} k_i(x_i, x_i') \\
k_2(\boldsymbol{x}, \boldsymbol{x}') &= \sigma_2 \sum_{i_1=1}^{D-1} \sum_{i_2=i_1+1}^{D} k_{i_1}(x_{i_1}, x_{i_1}') k_{i_2}(x_{i_2}, x_{i_2}') \\
k_3(\boldsymbol{x}, \boldsymbol{x}') &= \sigma_3 \sum_{i_1=1}^{D-2} \sum_{i_2=i_1+1}^{D-1} \sum_{i_3=i_2+1}^{D} k_{i_1}(x_{i_1}, x_{i_1}') k_{i_2}(x_{i_2}, x_{i_2}') k_{i_3}(x_{i_3}, x_{i_3}') \\
&\vdots \\
k_d(\boldsymbol{x}, \boldsymbol{x}') &= \\
&\sigma_d \sum_{i_1=1}^{I_1=D-d+1} \sum_{i_2=i_1+1}^{I_2=I_1+1} \cdots \sum_{i_d=i_{d-1}+1}^{I_d=I_{d-1}+1} k_{i_1}(x_{i_1}, x_{i_1}') k_{i_2}(x_{i_2}, x_{i_2}') \cdots k_{i_d}(x_{i_d}, x_{i_d}')
\end{aligned}
$$

where $k_i(x_i, x_i')$ denotes the kernel with one-dimensional inputs x_i and x_i', and $k_d(\boldsymbol{x}, \boldsymbol{x}')$ denotes the additive kernel of order d with D-dimensional inputs $\boldsymbol{x}$ and $\boldsymbol{x}'$. The resulting kernel of an additive GP model up to the dth order is the sum of the additive kernel from the 1th to dth order:

$$k_{1,\ldots,d}(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}') + \cdots + k_d(\boldsymbol{x}, \boldsymbol{x}').$$

The full additive kernel is a sum of the additive kernel of all orders:

$$k_{1,\ldots,D}(\boldsymbol{x}, \boldsymbol{x}') = k_1(\boldsymbol{x}, \boldsymbol{x}') + \cdots + k_D(\boldsymbol{x}, \boldsymbol{x}').$$

A GP with this kernel (an additive kernel) help us to determine which orders of interaction might be important. This model can improve computational efficacy

and interpretability. An additive kernel of all the orders of interaction sums over an exponential number of terms, which makes it intractable. Duvenaud et al. [2011] shows how to compute this kernel effectively.

In this work, an additive GP is the model used in the case study of land use classification in Section 3.11, where we deal with 53 input variables and we consider a 1st order additive GP model which implies 53 terms. An additive GP model composed of 53 one-dimensional GP components can be computationally extensive to be fitted using sampling methods. However, using approximate GPs, as the one proposed in Chapter 3 of this work, instead of using exact GPs, this additive GP model with 52 terms (one-dimensional terms) can be fitted using sampling methods in a few hours of computation.

2.4 Bayesian inference computation methods

Unfortunately, the posterior probability distribution cannot be handled analytically except in the simplest cases and using conjugate priors [Gelman et al., 2013, Minka, 2000]. Furthermore, it can be analytically intractable with high dimensionality. As a consequence, the predictive distribution cannot be computed in a closed form. Therefore, approximations are needed in these cases where exact Bayes can not be conducted.

In exact Bayes, the use of conjugate priors leads, in most of the cases, to closed-form and analytically tractable posterior and predictive distributions. The use of conjugate priors can reduce modeling flexibility and, furthermore, in many cases it is not possible the use of conjugate priors. Furthermore, even using conjugate priors, analytic solutions for high dimensional models or very complex hierarchical models can be difficult.

Approximate methods for solving posterior and marginal distributions in Bayesian inference can be roughly categorized in point estimates, distributional approximations, and sampling methods.

Point estimates approach the posterior of the parameter with a single best point estimate [O'Hagan and Forster, 2004]. A point estimate does not provide information about the shape of the posterior, so uncertainty is not considered. Point estimates have the drawbacks of getting possible local optima in non-linear functions and, also, being based on high probability density instead of high probability mass. The problems of overfitting are mostly related to point estimates [Bishop, 2006, Raiko et al., 2006].

- *Maximum a posteriori* (MAP) optimizes the posterior distribution over the parameter space. In MAP, the inference is based on conditioning the parameters on the data, which corresponds to a Bayesian approach.

- *Maximum likelihood* (ML) methods optimize the likelihood function over the parameter space. In ML, the prior distribution of the parameters are flat or uniform, and inference is based on conditioning the data on the parameters, which corresponds to a frequentist point of view. ML is the MAP solution with flat prior. ML estimates are attracted to high but sometimes narrow peaks and, unfortunately, this effect becomes stronger when the dimensionality increases [Bishop, 2006, Raiko et al., 2006].

Distributional approximation approaches try to approach the posterior by means of simple and analytically tractable distributions. *Laplace* method [Cseke and Heskes, 2011, Geweke, 1989] is a distributional approach that, centered on single best estimate, tries to approach the posterior with a simple distribution. Variational methods try to approach the full posterior with simple distributions by means of variational methods, such as *variational Bayes* [Beal et al., 2003, Gershman et al., 2012], *expectation propagation* [Cseke and Heskes, 2011, Minka, 2001] or *expectation maximization* [Little and Rubin, 2002, Liu et al., 1998].

Empirical Bayes consists in conducting Bayesian inference using point estimates for the prior parameters, so uncertainty on the prior parameters is ignored.

In a fully Bayesian approach, marginalization over the hyperparameter to obtain the posterior of model parameters or over the parameters to obtain the predictions, is needed. To approach these integrals, sampling methods based on Markov chain Monte Carlo (MCMC) are used.

2.4.1 Sampling methods

As said before, the posterior distribution

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{\int p(y|\theta)p(\theta)\mathrm{d}\theta}$$

and, consequently, the predictive distribution

$$p(\tilde{y}|y) = \int p(\tilde{y}|\theta, y)p(\theta|y)\mathrm{d}\theta$$

are, in most of cases, unable to be evaluated in a closed form. It is due to the fact that computing $\int p(y|\theta)p(\theta)\mathrm{d}\theta$ is usually difficult. However, the numerator $p(y|\theta)p(\theta)$ is usually easily evaluable for any θ.

Monte Carlo methods are based on sampling from the posterior using the unnormalized posterior (the numerator) $h(\theta|y) = p(y|\theta)p(\theta)$, and these draws are treated as a sample of parameter observations. As long as this sample can be considered as a realization of a *stationary ergodic process* [Roberts and Rosenthal,

2007], it can be used to compute means, deviations, quantiles, to draw histograms and to marginalize, etc. For example, the expectation of an arbitrary function f, which depends on a random variable θ with probability distribution $p(\theta)$, can be approximated by Monte Carlo integration:

$$\mathbb{E}[f(\theta)] = \int f(\theta)p(\theta)\mathrm{d}\theta \approx \frac{1}{N}\sum_{i=1}^{N} f(\theta^*),$$

where θ^* denotes a sample draw and N the number of draws of the sample. Similarly, as a main interest in Bayesian inference, the predictive distribution can be approximated as:

$$p(\tilde{y}|y) \approx \frac{1}{N}\sum_{i=1}^{N} p(\tilde{y}|\theta^*, y).$$

Thus, when using sampling methods, generally only the unnormalized posterior is taken into account, as the posterior is proportional to the likelihood and priors:

$$p(\theta|y) \propto p(y|\theta)p(\theta).$$

Unfortunately, generating a representative sample from the posterior efficiently is not trivial. The Monte Carlo methods used in Bayesian inference for sampling from the posterior are: *rejection sampling*, *importance sampling* and *Markov chain Monte Carlo*.

Markov chain Monte Carlo (MCMC) methods [Brooks et al., 2011, Gilks et al., 1995, MacKay, 2003] aim to estimate the posterior distribution by means of the generation of a sequence of random samples from it. MCMC samples are based on constructing a Markov chain that has the true posterior distribution as its equilibrium distribution. In a Markov Chain, $(\theta^1, \dots, \theta^t, \dots)$, each state θ^t only depends on the previous state, and consecutive states are related by a transition distribution $q(\theta^{t+1}|\theta^t)$. The difficulty is to generate a Markov chain that converges to its equilibrium distribution rapidly and with not too high autocorrelation in its sequence of values. A great amount of research has been conducted on MCMC methods, and numerous algorithms have been proposed. Excellent references on MCMC and advanced methods can be found in [Brooks et al., 2011, Gilks et al., 1995, Robert and Casella, 2013]. Following we make a brief review of the more relevant MCMC algorithms which have been used in this thesis.

Metropolis-Hasting sampling

Metropolis-Hasting algorithm [Hastings, 1970] is one of the most used due to its simplicity. It is always applicable when the unnormalized posterior can be evaluated pointwise. The algorithm produces a parameter sequence that converges to the target distribution $p(\theta|y)$. The draw θ^{t+1} is sampled from a proposal distribution $q(\theta^{t+1}|\theta^t)$ with an accepting probability given by

$$\min\Big(1, \frac{p(\theta^{t+1}|y)q(\theta^t|\theta^{t+1})}{p(\theta^t|y)q(\theta^{t+1}|\theta^t)}\Big).$$

If the proposed draw θ^{t+1} is not accepted, another different draw for the same state $t+1$ is proposed. The chosen proposal distribution is essential for efficient sampling and convergence of the algorithm. In practice, finding good proposal can be difficult in order not to get to many rejections or, contrarily, too often acceptance that parameter space is too little explored by the chain. This problem may lead to unrepresentative posterior sample or very slow convergence, especially in high dimensions.

Gibbs sampling

The Gibbs sampler [Geman and Geman, 1993] can be used for simulating from multivariate distributions when one is able to simulate from conditional distributions. The model parameters $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_D\}$ are updated cyclically from the full conditional distribution of the dth parameter given all the others, $\boldsymbol{\theta}_{-d}$ and the data y,

$$p(\theta_d|\boldsymbol{\theta}_{-d}, y).$$

If one can easily sample from full conditional distributions, with Gibbs sampling there is no need for tuning parameters and searching for transition probabilities. This is applicable in conditionally conjugate Bayesian models [Gelman et al., 2006] and in many hierarchical models with conjugate priors. The concept of *conditionally conjugate*, as mentioned in Gelman et al. [2013], can be useful when using Gibbs sampling methods. The conditionally conjugate refers to the posterior of a parameter conditioned to all the others $p(a|b, c)$ is of the same family of the prior used for that parameter $p(a)$.

Hamiltonian Monte Carlo

Hamiltonian Monte Carlo algorithm [Duane et al., 1987, Neal, 1993, Neal et al., 2011] introduces gradient information in improving efficiency on the proposals and

reduce random walk behavior of the sampling. The gradients help the algorithm to find high probability states. HMC can potentially improve sampling efficiently, but the gradients of the distribution need to be tractable. Additional parameters need to be tuned, which makes its implementation more difficult than MC. However, methods have been recently developed for automatic adaptation of the parameters, such as no-U-turn sampler (NUTS) [Betancourt, 2016, Hoffman and Gelman, 2014]. HMC algorithm with sampler NUTS in the probabilistic programming software Stan [Carpenter et al., 2017, Team, 2017] has been extensively used in Chapters 3, 4 and 5 of this work.

Convergence diagnosis

Reaching convergence on the sample is essential, otherwise, the samples do not come from the posterior distribution and subsequent analysis are meaningless.

The initial part of the chain may have not reached convergence yet. Furthermore, it may also include phase for adapting algorithm parameters. So the initial part of the chain may be non-representative. This initial part is commonly called *burn-in* or *warm-up* and must be thrown away.

Running several chains from different initial values can help diagnosis. Visual inspection of the chains is straightforward. Furthermore, several methods for evaluating convergence have been proposed. Perhaps, the most commonly used is the *potential scale reduction factor* [Brooks et al., 2011], which is basically based on comparing the mean and variance of a single chain to the mean and variance of all chains in order to assess whether convergence of chains has been achieved.

The effective number of samples, which describes how many independent samples have been generated, is descriptive of the effectiveness of the sampling and of the autocorrelation of the chain.

Finally, there are problematic distributions for performing sampling methods, such as distributions with nonlinear dependencies or 'funnels' distributions, where optimal proposal depends on location. Furthermore, it can be difficult to move from one mode to another in multimodal distributions, and the central limit theorem for expectations does not hold in long-tailed distributions with non-finite variance and mean.

2.5 Sensitivity analysis and model validation and comparison

A well-known premise in statistics, which was attributed to the statistician George Box, is that "*all models are wrong, but some are useful*". A model is just a simple

representation of a true phenomena. Checking model performance and inference is essential since models can be sensitive to additional information not used in the modeling or models can be sensitive to the underlying assumptions, such as to prior and model assumptions. Which may make the posterior distributions either to underestimate or overestimate the 'true' posterior density.

In order to conduct a sensitivity analysis, some strategies can be used. For example, a comparison to simpler methods can be made and, if our method gives poorer results, our assumptions are questionable. Different models can be applied and/or different choices for priors can be checked. Sensitivity on essential inference quantities can be compared, taking into account that, for example, extreme quantiles are more sensitive than means and medians or extrapolation is more sensitive than interpolation.

Furthermore, we need to do posterior checking against the observed data. If an additional representative set of sample data is not available, cross validation methods for model checking and comparison by checking predictive accuracy can be used. For model checking, the *posterior predictive checks*, which are also known as the *leave-one-out probability integral transformation* (LOO-PIT) can be used in order to guarantee good model performance and ensure that the model is compatible with the observed data. They are based on computing the probability of a new observation $\tilde{y}_i$ to be lower or equal to its corresponding actual observation y_i [Gelfand et al., 1992, Gelman et al., 2013]:

$$\text{LOO-PIT}_i = P(\tilde{y}_i \leq y_i).$$

The similarity or provenance of these probabilities from standard uniform distributions endorses these probabilities with the desirable property of having the same interpretation across models, which implies good fit to the data and good prediction [Bayarri and Berger, 2000].

For model comparison, the *mean square predictive error* (MSE) or the *expected log posterior predictive density* (ELPD) can be used. The MSE evaluates, by averaging over all checking observations, how far new data is from the model by using the distance (error) between the actual observation y_i and the predictive mean $\tilde{y}_i$

$$\text{MSE} = \frac{1}{N}\sum_{i}^{N}(y_i - \tilde{y}_i)^2.$$

The ELPD evaluates, by averaging over all checking observations, how far new data is from the model while taking the posterior uncertainties into account. It is based on the log-density of new data given the model [Andersen et al., 2019, Vehtari et al.,

2012]:

$$\text{ELPD} = \frac{1}{N}\sum_{i}^{N} \ln(p(y_i|\boldsymbol{y}_{-i})), \tag{2.3}$$

where $\boldsymbol{y}_{-i}$ denotes the dataset without the observation i.

Chapter 3

Hilbert space methods to approximate Gaussian processes

In this chapter, we analyze the performance and practical implementation of a recently theoretically developed novel approach for low-rank approximate Gaussian processes. Low-rank approximate Gaussian processes are of main interest in machine learning and statistics due to the high computational demands of exact Gaussian process models. With this study we make the contribution of analyzing in detail the performance of the method, providing the recommendations for its practical implementation. We show the simplicity of the method, with an attractive computational complexity, due to its linear structure, which makes it easier to be used as building blocks in more complicated models and using statistical programming frameworks. Several illustrative examples of the performance, applicability and implementation of the method in the Stan programming software are presented, and their Stan model codes are also provided.

Furthermore, and before going through the main contribution of the chapter, a detailed introduction to exact Gaussian processes is presented. The prior, posterior and predictive probability distributions of Gaussian processes are derived. And the main elements of Gaussian processes, the covariance and the spectral density functions, are also briefly described. Gaussian processes are the main modeling framework on which we base and derive most of the modeling contributions and applications made in this work.

3.1 Introduction

Gaussian processes (GPs) are flexible statistical models for specifying probability distributions over multi-dimensional non-linear functions [Neal, 1997, Rasmussen and Williams, 2006]. Their name stems from the fact that any finite set of function values is jointly distributed as a multivariate Gaussian. GPs are defined by a mean and a covariance function. The covariance function encodes our prior assumptions about the functional relationship, such as continuity, smoothness, periodicity and scale properties. GPs not only allow for non-linear effects but can also implicitly handle interactions between input variables (covariates). Different types of covariance functions can be combined for further increased flexibility. Due to their generality and flexibility, GPs are of broad interest across machine learning and statistics [Neal, 1997, Rasmussen and Williams, 2006]. Among others, they find application in the fields of spatial epidemiology [Banerjee et al., 2014, Diggle, 2013], robotics and control [Deisenroth et al., 2015], signal processing [Särkkä et al., 2013], as well as Bayesian optimization and probabilistic numerics [Briol et al., 2015, Hennig et al., 2015, Roberts, 2010].

The key element of a GP is the covariance function that defines the dependence structure between function values at different inputs, and allows for non-linear effects. However, the covariance function is also a computational issue because of the need of inverting its Gram matrix to optimize the hyperparameters. That is, given n observations in the data, the computational complexity in covariance matrix inversion and memory requirements of exact GP implementation in general scale as $O(n^3)$ and $O(n^2)$, respectively. This limits their application to rather small data sets of a few tens of thousands observations at most. The problem becomes more severe when performing full Bayesian inference via sampling methods, where in each sampling step we need $O(n^3)$ computations when inverting the Gram matrix of the covariance function, usually through Cholesky factorization. To alleviate these computational demands, several approximate methods have been proposed. In Section 3.2, we make a brief review of the most common present methods for low-rank approximations of GPs. As a summary, these can be roughly classified into two general approaches. Sparse GPs are based on low-rank approximations of the covariance matrix with a set of $m \ll n$ *inducing points* that summarizes the actual data (n). An alternative class of low-rank approximations is based on forming a basis function approximation with $m \ll n$ basis functions. The basis functions are usually presented explicitly, but can also be used to form a low rank covariance matrix approximation.

In this study, we propose an approximate framework for fast and accurate inference for Gaussian processes. We focus on the basis function approximation via Laplace eigenfunctions for stationary covariance functions proposed by Solin and

Särkkä [2018]. Basis function approaches behave computationally like linear models, which is an attractive property in modular probabilistic programming models where there is a big benefit if approximation specific computation is simple. Then, it is easier to use Gaussian processes as building blocks in more complicated models and can be used as latent functions in non-Gaussian observational models allowing modeling flexibility. The Laplace eigenfunctions can be computed analytically and they are independent of the particular choice of the covariance kernel including the hyperparameters. While the pre-computation cost of the basis functions is $O(mn)$, the computational cost in learning the covariance function hyperparameters scales as $O(nm + m)$ in every step of the optimizer. This is a big advantage in terms of speed for iterative algorithms such as Markov chain Monte Carlo (MCMC). Another advantage is the reduced memory requirements of automatic differentiation methods used in modern probabilistic programming frameworks, such as Stan [Carpenter et al., 2017], WinBUGS [Lunn et al., 2000] and others. This is because the memory requirements of automatic differentiation rather scale with the computational complexity instead of with the usual memory requirements for the posterior density computation. The basis function approach also provides an easy way to apply the non-centered parameterization of GPs, which reduces the posterior dependency between parameters representing the estimated function and the hyperparameters of the covariance function, which further improves MCMC efficiency. Furthermore, it can be made arbitrarily accurate and the trade-off between computational complexity and approximation accuracy can easily be controlled.

While Solin and Särkkä [2018] have fully developed the mathematical theory behind this specific approximation of GPs, further work is needed for its practical implementation in probabilistic programming frameworks. They do not put much effort in describing and analyzing the relation among the key factors of the box size (this can also be referred to as desired prediction space or boundary condition), the number of basis functions, and the properties of the true functional relationship between covariates and response variable (smoothness or roughness of the function to be learned). The performance and accuracy of the method are directly related with the number of basis functions and the box size. At the same time, successful values for these two factors depend on the smoothness or roughness of the function to be learned (the non-linearity of the function to be learned). The time of computation is mainly dependent on the number of basis functions. In this study, we analyze in detail the performance and accuracy of the method in relation to these key factors: the number of basis functions, desired prediction space, and smoothness or roughness of the function. We provide intuitive visualizations and practical recommendations for the choice of these factors, which will help users to improve computational performance while maintaining close approximation to exact GPs.

Although there are several GP specific software packages available to date

(GPML [Rasmussen and Nickisch, 2010], GPstuff [Vanhatalo et al., 2013], GPy [GPy, 2012], GPflow [Matthews et al., 2017]), each provide efficient implementations only for a restricted range of GP based models. In this study, we do not focus on the fastest possible inference for some specific GP models, but instead are interested in how GPs can be easily used as modular components in probabilistic programming frameworks.

The remainder of the chapter is structured as follows. In Section 3.2, a brief overview of related work is given. In Section 3.3, we describe the main contributions of this chapter. In Section 3.4, we introduce the exact GP model. In Section 3.5, we introduce the reduced rank approximations to GPs proposed by Solin and Särkkä [2018]. In Section 3.6, we analyze the accuracy of these approximations under several conditions using analytical and numerical methods. In Chapter 3.7, we introduce the low-rank approximation for the particular case of a GP model with a periodic covariance function following Solin and Särkkä [2014]. Finally, several case studies in which we fit exact and approximate GPs to real and simulated data using Stan, a probabilistic programming software, are provided in Sections 3.8, 3.9, 3.10 and 3.11. We end with a brief conclusion in Section 3.12.

More case studies are presented in Appendix A. The Stan model codes for all case studies are provided through links to the author's GitHub repository.

3.2 State of the art

The GP prior entails an $O(n^3)$ complexity that is computationally intractable for many practical problems. To overcome this scaling problem several schemes have been proposed. One approach is to partition the data set into separate groups and performing local inference in each partition [Snelson and Ghahramani, 2007, Urtasun and Darrell, 2008].

Other global approach is to build a low-rank approximation to the covariance matrix of the complete data based around 'inducing variables', also known as sparse GPs [Bui et al., 2017, Quiñonero-Candela and Rasmussen, 2005]. This approach, based on inducing points, employs a small set of pseudo data points (m) to summarise the actual data (n). The storage requirements are reduced to $O(nm)$ and complexity to $O(nm^2)$, where $m \ll n$. Some of these methods have been reviewed in Rasmussen and Williams [2006], and Quiñonero-Candela and Rasmussen [2005] provide a unifying view of these methods based on approximate generative methods. Burt et al. [2019] show that for regression with normally distributed covariates in D dimensions and using the squared exponential covariance function, $m = O(\log^D n)$ is sufficient for accurate approximation. Several of these methods (e.g., SoR, DTC, VAR, FIC) are basically based on choosing a set of m inducing inputs x_u aiming to

match their corresponding covariance matrix $K_{u,u}$ to the covariance matrix of the actual data, by means of approximating the eigensprectrums. These methods can be seen as modifications of the Nyström method (see Arthur [1979]) and were originally introduced to approximate GPs by Williams and Seeger [2001]. In conventional sparse GP approximations based on inducing points, although the rank of the GP is reduced considerably to the number of inducing points, this still needs to do the automatic differentiation and covariance matrix inversion.

The inducing points-based sparse approximation methods works, in practice, reasonable well in relatively smoothed processes. Vanhatalo et al. [2010] propose the use of compactly supported covariance function jointly with sparse approximations to model both short and long range correlations. In general sparse GPs, the number of inducing points or their locations are crucial in order to capture the correlation structure. For a discussion on the effects of the inducing points, see Vanhatalo et al. [2010] and Banerjee et al. [2008].

More recent developments in the context of sparse GPs include a structured kernel interpolation method [Wilson and Nickisch, 2015], which combines the inducing point approach and the structure exploiting for scalability, such as Kronecker [Saatçi, 2012] or Toeplitz [Cunningham et al., 2008] approaches. This framework improves the scalability and accuracy of fast kernel approximations through kernel interpolation. On the other hand, Wang et al. [2019] have recently developed a scalable approach for exact GPs and they demonstrate that an exact GP can be fitted over a million points. They make use of GPU parallelization and methods like linear conjugate gradients, accessing the kernel matrix only through matrix multiplication.

Another global approach of low-rank approximations is based on forming a basis function approximation with $m \ll n$ basis functions. The basis functions are usually presented explicitly, but can also be used to form a low rank covariance matrix approximation. Common basis function approximations rely on the spectral analysis and series expansions of Gaussian processes [Adler, 1981, Cramér and Leadbetter, 2013, Loève, 1977]. A classical result is that the covariance function can be approximated with a finite truncation of Mercer series and the approximation is guaranteed to converge to the exact covariance function when the number of terms is increased. Another related classical connection is to the works in the relationship of spline interpolation and Gaussian process priors [Kimeldorf and Wahba, 1970, Wahba, 1978, 1990]. In particular, it is well-known (see, e.g. Wahba [1990] and Furrer and Nychka [2007]) that spline smoothing is equivalent to Gaussian process regression with certain covariance function. The relationship of the spline regularization with Laplace operators then leads to series expansion representations that are closely related to the approximations considered here. The splines models are also based on a series expansion of basis functions, then the computational demands are similar to the demands of this approach. However, spline models do

not have an explicit parameter controlling the correlation length as many of the common GP covariance functions do, and then the fit is covered by the magnitude parameter. In that sense, splines models do not have the useful interpretation of the lengthscale parameter. Recent Splines models can reproduce the Matern family of covariance functions (see, e.g., Wood [2003]), however our approach can reproduce basically all of the stationary covariance functions.

Sparse spectrum GPs are based on a sparse approximation to the frequency domain representation of a GP [Gal and Turner, 2015, Lázaro Gredilla, 2010, Quiñonero-Candela et al., 2010], where the spectral representation of the covariance function is used. Recently, Hensman et al. [2017] presented a variational Fourier feature approximation for Gaussian processes that was derived for the Matérn class of kernels, where the approximation structure is set up by a low-rank plus diagonal structure. They combine the variational methodology with Fourier based approximations. Another related method for approximating kernels relies on random Fourier features [Rahimi and Recht, 2008, 2009]. The approximate kernel has a finite basis function expansion. While Sparse Spectrum GP is based on a sparse spectrum, the reduced-rank method proposed in this study aims to make the spectrum as 'full' as possible at a given rank.

The literature contains many parametric models that approximate GP behaviours; for example, Bui and Turner [2014] included tree-structures in the approximation for extra scalability, and Moore and Russell [2015] combined local Gaussian processes with Gaussian random fields.

3.3 Contributions of the chapter

Our main contributions to this recently developed methodology for low-rank GP model by Solin and Särkkä [2018] goes around these aspects:

• Firstly, clear summarized formulae of the method for the univariate and multivariate cases is presented. Furthermore, the methodology in the particular case of a GP with a periodic covariance function is also presented.

• The relations going on among the key factors, the number of basis functions, the box size, and the lengthscale of the functions to be learned, are investigated. Let us highlight that the lengthscale is the parameter of the covariance function that ultimately characterizes the non-linearity of the posterior functions.

• Recommendations for the values of the key factors based on the recognized relations among them are given. We provide useful graphs of these relations that will help the users to improve performance and save time of computation.

• A diagnosis of whether the chosen values for the number of basis functions

and the box size are adequate to fit to the actual data is proposed.

• The generalization of the method to the multidimensional case is described.

• The approach is implemented in a fully probabilistic framework and for the Stan programming probabilistic software. This work has served as a basis for the subsequent implementation of the method in the R-package *brms* [Bürkner et al., 2017].

• Several illustrative examples, simulated and real datasets, of the performance and applicability of the model, and accompanied by their Stan model codes, are developed.

3.4 Gaussian process model

A Gaussian process (GP) is a stochastic process that defines the distribution over a collection of random variables indexed by a continuous variable, i.e. $\{f(t) : t \in \mathcal{T}\}$ for some index set $\mathcal{T}$. Gaussian processes have the defining property that the distribution of any finite subset of random variables, $\{f(t_1), f(t_2), \ldots, f(t_K)\}$, is a multivariate Gaussian distribution.

In this work, Gaussian processes will take the role of a prior distribution over function spaces for non-parametric latent functions in a Bayesian setting. Consider a data set $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, where y_n is modelled conditionally as $p(y_n|f(\boldsymbol{x}_n), \phi)$, where p is some parametric distribution with parameters f and ϕ, and f is an unknown function with Gaussian process prior, which depends on an input $\boldsymbol{x}_n \in \mathbb{R}^D$. This generalizes trivially to more complex models depending on several unknown functions, such as $p(y_n|f(\boldsymbol{x}_n), g(\boldsymbol{x}_n))$, or multilevel models. Our goal is to obtain posterior distribution for the value of the function $\tilde{f} = f(\tilde{\boldsymbol{x}})$ evaluated at a new input $\tilde{\boldsymbol{x}}$.

We assume a Gaussian process prior for $f \sim \mathcal{GP}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$, where $\mu : \mathbb{R}^D \to \mathbb{R}$ and $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ are the mean and covariance functions, respectively,

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

The mean and covariance functions completely characterize the Gaussian process prior, and control the a priori behavior of the function f. Let $\boldsymbol{f} = \{f(\boldsymbol{x}_n)\}_{n=1}^N$, then the resulting prior distribution for $\boldsymbol{f}$ is a multivariate Gaussian distribution $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$, where $\boldsymbol{\mu} = \{\mu(\boldsymbol{x}_n)\}_{n=1}^N$ is the mean and $\boldsymbol{K}$ the covariance matrix, where $K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ may depend on a set of

hyperparameters, $\boldsymbol{\theta}$, but we will not write this dependency explicitly to ease the notation. The joint distribution of $\boldsymbol{f}$ and a new $\tilde{f}$ is also a multivariate Gaussian as

$$p(\boldsymbol{f}, \tilde{f}) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \tilde{\mu} \end{bmatrix} \middle| \mathbf{0}, \begin{bmatrix} \boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} & \boldsymbol{k}_{\boldsymbol{f},\tilde{f}} \\ \boldsymbol{k}_{\tilde{f},\boldsymbol{f}} & k_{\tilde{f},\tilde{f}} \end{bmatrix} \right),$$

where $\boldsymbol{k}_{\boldsymbol{f},\tilde{f}}$ is the covariance between $\boldsymbol{f}$ and $\tilde{f}$, and $k_{\tilde{f},\tilde{f}}$ is the prior variance of $\tilde{f}$. By using the conditional properties of multivariate Gaussian distributions, we can derive the predictive distribution for $\tilde{f}$ given $\boldsymbol{f}$ analytically,

$$p(\tilde{f}|\boldsymbol{f}) = \mathcal{N}(\tilde{f}|\boldsymbol{k}_{\tilde{f},\boldsymbol{f}}\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}}^{-1}\boldsymbol{f},\ k_{\tilde{f},\tilde{f}} - \boldsymbol{k}_{\tilde{f},\boldsymbol{f}}\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}}^{-1}\boldsymbol{k}_{\boldsymbol{f},\tilde{f}}).$$

The joint distribution of the observations $\boldsymbol{y} = \{y_n\}_{n=1}^N$ and function values $\boldsymbol{f}$ and $\tilde{f}$, $p(\boldsymbol{y}, \boldsymbol{f}, \tilde{f})$, is the product of the conditional distribution for $\boldsymbol{y}$ given $\boldsymbol{f}$ and the joint distribution for $\boldsymbol{f}$ and $\tilde{f}$:

$$p(\boldsymbol{y}, \boldsymbol{f}, \tilde{f}) = p(\boldsymbol{y}|\boldsymbol{f})\, p(\boldsymbol{f}, \tilde{f}).$$

By marginalizing over $\boldsymbol{f}$ and conditioning on the observations $\boldsymbol{y}$, we obtain the posterior distribution of interest

$$p(\tilde{f}|\boldsymbol{y}) = \frac{\int p(\boldsymbol{y}|\boldsymbol{f})\, p(\boldsymbol{f}, \tilde{f})\, \mathrm{d}\boldsymbol{f}}{p(\boldsymbol{y})}, \tag{3.1}$$

where $p(\boldsymbol{y})$ is the marginal likelihood and is given by

$$p(\boldsymbol{y}) = \int p(\boldsymbol{y}|\boldsymbol{f})\, p(\boldsymbol{f}, \tilde{f})\, \mathrm{d}\boldsymbol{f}\, \mathrm{d}\tilde{f}. \tag{3.2}$$

If the observational model $p(\boldsymbol{y}|\boldsymbol{f})$ is Gaussian, both integrals in equation (3.1) and equation (3.2) can be solved analytically conditioned on the hyperparameters. For example, a Gaussian likelihood $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{f}, \sigma^2)$, with noise variance σ^2, yields the following closed-form solution:

$$\begin{aligned} p(\tilde{f}|\boldsymbol{y}) =&\ \mathcal{N}(\tilde{f}|\mu_{\tilde{f}}, \sigma_{\tilde{f}}^2), \\ \mu_{\tilde{f}} =&\ \boldsymbol{k}_{\tilde{f},\boldsymbol{f}}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{f}, \\ \sigma_{\tilde{f}}^2 =&\ k_{\tilde{f},\tilde{f}} - \boldsymbol{k}_{\tilde{f},\boldsymbol{f}}(\boldsymbol{K}_{\boldsymbol{f},\boldsymbol{f}} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{k}_{\boldsymbol{f},\tilde{f}}. \end{aligned}$$

If $p(y_n|f(\boldsymbol{x}_n), \phi) = N(y_n|f(\boldsymbol{x}_n), \sigma)$ then $\boldsymbol{f}$ can be integrated out analytically (with a computational cost of $O(n^3)$ for exact GP and $O(nm^2)$ for sparse GP).

If $p(y_n|f(\boldsymbol{x}_n), \phi) = N(y_n|f(\boldsymbol{x}_n), g(\boldsymbol{x}_n))$ or $p(y_n|f(\boldsymbol{x}_n), \phi)$ is non-Gaussian, the marginalization does not have closed form solution. Furthermore, if a prior distribution is imposed on ϕ and θ to form a joint posterior for ϕ, θ and $\boldsymbol{f}$, approximate inference such as Markov chain Monte Carlo (MCMC) [Brooks et al., 2011] Laplace approximation ([Rasmussen and Williams, 2006, Williams and Barber, 1998], expectation propagation [Minka, 2001], or variational Bayes methods [Csató et al., 2000, Gibbs and MacKay, 2000] need to be used. In this paper we focus on the use of MCMC for integrating over the joint posterior. MCMC is not usually the fastest approach, but allows accurate inference for general models in probabilistic programming settings. We consider the computational costs of GPs specifically from this point of view.

3.4.1 Covariance function and spectral density

The covariance function is the crucial ingredient in a GP as it encodes our prior assumptions about the variation of the function, and defines a correlation structure which characterizes the correlations between function values at different inputs. A covariance function can be expressed as a positive semidefinite kernel function, such that the Gram matrix corresponding to the covariance function is positive semidefinite [Rasmussen and Williams, 2006]. In this work, the terms *covariance function* and *kernel* might be used interchangeably.

A stationary covariance function is a function of $\boldsymbol{\tau} = \boldsymbol{x} - \boldsymbol{x}' \in \mathbb{R}^D$, such that it can be written $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{\tau})$, which means that the covariance is invariant to translations. Isotropic covariance functions are those that are function of the distance between observations, $k(\boldsymbol{x}, \boldsymbol{x}') = k(|\boldsymbol{x} - \boldsymbol{x}'|) = k(r)$, $r \in \mathbb{R}$, which means that the covariance is both translation and rotation invariant. The most commonly used distance between observations is the norm L2 ($|\boldsymbol{x} - \boldsymbol{x}'|_{L2}$), also known as Euclidean distance, although other types of distances can be considered.

The Matérn class of isotropic covariance functions is given by

$$k_\nu(r) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{\ell} \right)^\nu K_\nu \left(\frac{\sqrt{2\nu} r}{\ell} \right),$$

where ν is the order the kernel, K_ν the modified Bessel function of the second kind, and ℓ and $\sqrt{\sigma^2}$ are the length-scale and magnitude, respectively, of the kernel (covariance function). The particular case where $\nu = \infty$ and $\nu = 3/2$ are probably

the most commonly used kernels [Rasmussen and Williams, 2006]:

$$k_\infty(r) = \sigma^2 \exp\left(-\frac{1}{2}\frac{r^2}{\ell^2}\right),$$
$$k_{\frac{3}{2}}(r) = \sigma^2\left(1+\frac{\sqrt{3}r}{\ell}\right)\exp\left(-\frac{\sqrt{3}r}{\ell}\right).$$

The former is commonly known as squared exponential (exponentiated quadratic) covariance function. Assuming the Euclidean distance between observations, $r = |\boldsymbol{x}-\boldsymbol{x}'|_{L2} = \sqrt{\sum_{i=1}^{D}(x_i - x_i')^2}$, the kernels written above take the form:

$$k_\infty(|\boldsymbol{x}-\boldsymbol{x}'|_{L2}) = \sigma^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{D}\frac{(x_i-x_i')^2}{\ell_i^2}\right),$$
$$k_{\frac{3}{2}}(|\boldsymbol{x}-\boldsymbol{x}'|_{L2}) = \sigma^2\left(1+\sqrt{\sum_{i=1}^{D}\frac{3(x_i-x_i')^2}{\ell_i^2}}\right)\exp\left(-\sqrt{\sum_{i=1}^{D}\frac{3(x_i-x_i')^2}{\ell_i^2}}\right).$$

Notice that the previous expressions have been easily generalized to using a multidimensional length-scale $\boldsymbol{\ell} \in \mathbb{R}^D$. The use of a multidimensional length-scale basically turns the isotropic covariance function into non-isotropic.

Stationary covariance functions can be represented in terms of their spectral densities. That is, the covariance function of a stationary process, that is function of $\boldsymbol{\tau} = \boldsymbol{x} - \boldsymbol{x'}$, can be represented as the Fourier transform of a positive finite measure (*Bochner's theorem*, see, e.g. Akhiezer and Glazman [1993]).

(Bochner's theorem) A complex-valued function k on $\mathbb{R}^D$ is the covariance function of a weakly stationary mean square continuous complex valued random process on $\mathbb{R}^D$ if and only if it can be represented as

$$k(\boldsymbol{\tau}) = \int_{\mathbb{R}^D} e^{2\pi i \boldsymbol{s}\cdot\boldsymbol{\tau}} d\mu(\boldsymbol{\tau}),$$

where μ is a positive finite measure.

If the measure μ has a density, it is known as the spectral density $S(\omega)$ of the covariance function, and the covariance function and the spectral density are Fourier duals, known as the *Wiener-Khintchine theorem* [Rasmussen and Williams, 2006].

It gives the following relations:

$$k(\boldsymbol{\tau}) = \int S(\boldsymbol{s}) e^{2\pi i \boldsymbol{s}\cdot\boldsymbol{\tau}} d\boldsymbol{s},$$
$$S(\boldsymbol{s}) = \int k(\boldsymbol{\tau}) e^{-2\pi i \boldsymbol{s}\cdot\boldsymbol{\tau}} d\boldsymbol{\tau}.$$

The spectral density functions associated with the Matérn class of covariance functions are given by

$$S_\nu(\omega) = \sigma^2 \, \frac{2^D \pi^{D/2} \Gamma(\nu + D/2)(2\nu)^\nu}{\Gamma(\nu)\ell^{2\nu}} \left(\frac{2\nu}{\ell^2} + 4\pi^2\omega^2\right)^{(\nu+D/2)},$$

in D dimensions, where variable $\omega \in \mathbb{R}$ is a distance in the frequency domain, and ℓ and σ are the lengthscale and magnitude, respectively, of the kernel (see Rasmussen and Williams [2006]). The particular cases where $\nu = \infty$ and $\nu = 3/2$ take the form

$$S_\infty(\omega) = \sigma^2 \left(\sqrt{2\pi}\right)^D \ell^D \exp\left(-0.5\ell^2\omega^2\right), \tag{3.3}$$
$$S_{\frac{3}{2}}(\omega) = \sigma^2 \, \frac{2^D \pi^{D/2} \Gamma(\frac{D+3}{2}) \left(\sqrt{3}\right)^3}{\frac{1}{2}\sqrt{\pi}\ell^3} \left(\frac{3}{\ell^2} + \omega^2\right)^{-\frac{D+3}{2}}. \tag{3.4}$$

Particularizing to an input dimension $D = 3$ and Euclidean distance $\omega = \sqrt{\sum_{i=1}^3 s_i^2}$, and considering a multidimensional lengthscale $\boldsymbol{\ell} \in \mathbb{R}^3$, the spectral densities written above take the form

$$S_\infty(\omega) = \sigma^2 \left(\sqrt{2\pi}\right)^3 \prod_{i=1}^3 \ell_i \exp\left(-\frac{1}{2}\sum_{i=1}^3 \ell_i^2 s_i^2\right),$$
$$S_{\frac{3}{2}}(\omega) = \sigma^2 \, 32\pi \left(\sqrt{3}\right)^3 \prod_{i=1}^3 \ell_i \left(3 + \sum_{i=1}^3 \ell_i^2 s_i^2\right)^{-3}.$$

3.5 Hilbert space approximate Gaussian process model

The approximate Gaussian process method, developed by Solin and Särkkä [2018] and implemented in this chapter, is based on considering the covariance operator of a homogeneous (stationary) covariance function as a pseudo-differential operator constructed as a series of Laplace operators. Then, the pseudo-differential operator is approximated with Hilbert space methods on a compact subset $\Omega \subset \mathbb{R}^D$ subject to some boundary condition. For brevity, we will refer to these approximate GPs as HSGPs. Below, we will present the main results around HSGPs relevant for practical application. More details and mathematical proofs are provided in Solin and Särkkä [2018]. Our starting point for presenting the method is the main result obtained by Solin and Särkkä [2018] of the definition of the covariance function as a series expansion of eigenvalues and eigenfunctions of the Laplacian operator. The mathematical details of this approximation have been briefly presented in Appendix B.

We begin by focusing on the case of a unidimensional input space (i.e., on GPs with just a single covariate) such that $\Omega \in [-L, L] \subset \mathbb{R}$, where L is some positive real value to which we also refer as boundary condition. As Ω describes the interval in which the approximations are valid, L plays a critical role in the accuracy of HSGPs. We will come back to this issue in Section 3.6.

Within Ω, we can write any stationary covariance function with input values $\{x, x'\} \in \Omega$ as

$$k(x, x') = \sum_{j=1}^{\infty} S_\theta(\sqrt{\lambda_j})\phi_j(x)\phi_j(x'), \tag{3.5}$$

where S_θ is the spectral density of the stationary covariance function k (see Section 3.4.1) and θ the set of hyperparameters of k. Terms $\{\lambda_j\}_{j=1}^{\infty}$ and $\{\phi_j(x)\}_{j=1}^{\infty}$ are the sets of eigenvalues and eigenfunctions, respectively, of the Laplacian operator in the given domain Ω. Namely, they satisfy the following eigenvalue problem in Ω when applying the Dirichlet boundary condition (other boundary conditions could be used as well):

$$\begin{aligned} -\nabla^2\phi_j(x) &= \lambda\phi_j(x), & x &\in \Omega \\ \phi_j(x) &= 0, & x &\notin \Omega. \end{aligned} \tag{3.6}$$

The eigenvalues $\lambda_j > 0$ are real and positive because the Laplacian is a positive definite Hermitian operator, and the eigenfunctions ϕ_j for the eigenvalues problem in Equation (3.6) are sinusoidal functions. Independently of the covariance function,

they can be computed as

$$\lambda_j = \left(\frac{j\pi}{2L}\right)^2, \tag{3.7}$$

$$\phi_j(x) = \sqrt{\frac{1}{L}}\sin\left(\sqrt{\lambda_j}(x+L)\right). \tag{3.8}$$

If we truncate the sum in (3.5) to the first m terms, the approximate covariance function becomes

$$k(x,x') \approx \sum_{j=1}^{m} S_\theta(\sqrt{\lambda_j})\phi_j(x)\phi_j(x') = \boldsymbol{\phi}(x)^\intercal \Delta \boldsymbol{\phi}(x'),$$

where $\boldsymbol{\phi}(x) = \{\phi_j(x)\}_{j=1}^m \in \mathbb{R}^m$ is the column vector of basis functions, and $\Delta \in \mathbb{R}^{m\times m}$ is the diagonal matrix of the spectral densities $S_\theta(\sqrt{\lambda_j})$:

$$\Delta = \begin{bmatrix} S_\theta(\sqrt{\lambda_1}) & & \\ & \ddots & \\ & & S_\theta(\sqrt{\lambda_m}) \end{bmatrix}.$$

Thus, the Gram matrix K of the covariance function k for a set of observations $i = 1,\ldots,n$ and corresponding input values $\{x_i\}_{i=1}^n \in \Omega^n$ can be represented as

$$\mathrm{K} = \Phi\Delta\Phi^\intercal,$$

where $\Phi \in \mathbb{R}^{n\times m}$ is the matrix of eigenfunctions $\phi_j(x_i)$:

$$\Phi = \begin{bmatrix} \phi_1(x_1) & \cdots & \phi_m(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_n) & \cdots & \phi_m(x_n) \end{bmatrix}.$$

As a result, the model for f can be written as

$$\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \Phi\Delta\Phi^\intercal).$$

This equivalently leads to a linear representation of $\boldsymbol{f}$ via

$$\boldsymbol{f} \approx \boldsymbol{\mu} + \Phi\Delta^{1/2}\boldsymbol{\beta},$$

where $\boldsymbol{\beta} = (\beta_1,\ldots,\beta_m) \sim \mathrm{Normal}(0, I)$, with I the identity matrix.

Let $\boldsymbol{f} = \{f(x_i)\}_{i=1}^n$, then the function f takes the form:

$$f(x) \approx \sum_{j}^{m} \left(S_\theta(\sqrt{\lambda_j})\right)^{1/2} \phi_j(x)\beta_j, \tag{3.9}$$

where $\beta_j \sim \text{Normal}(0, 1)$. Thus, the function f is approximated with a finite basis function expansion (using the eigenfunctions ϕ_j of the Laplace operator), scaled by the square root of spectral density values. A key property of this approximation is that the eigenfunctions ϕ_j do not depend on the covariance hyperparameters θ, and therefore only need to be constructed once, at cost $O(mn)$. Instead, the only dependence on θ is through the spectral density S_θ. The eigenvalues λ_j are monotonically increasing with j and S_θ goes rapidly to zero for bounded covariance functions. Therefore, equation (3.9) can be expected to be a good approximation for a finite number of m terms in the series as long as the inputs values x_i are not too close to the boundaries $-L$ and L of Ω. The computational cost, in learning the covariance function hyperparameters, of univariate HSGPs scales as $O(nm + m)$ in every step of the optimizer, where n is the number of observations and m the number of basis functions.

The parameterization in equation (3.9) is naturally in the non-centered parameterization form with independent prior distribution on β_j, which makes posterior inference easier. Furthermore, all dependencies on the covariance kernel and the hyperparameters is through the prior distribution of the regression weights β_j. The parameter posterior distribution $p(\boldsymbol{\beta}|\boldsymbol{y})$ is m-dimensional, where m is much smaller than the number of observations n. Therefore, the parameter space is greatly reduced and this makes inference faster, especially when sampling methods are used.

3.5.1 Generalization to multidimensional Gaussian processes

The results from the previous section can be generalized to a multidimensional input space with compact regular domain $\Omega = [-L_1, L_1] \times \cdots \times [-L_d, L_d]$ and Dirichlet boundary conditions. In a D-dimensional input space, the total number of eigenfunctions and eigenvalues in the approximation is equal to the number of D-tuples, that is possible combinations of univariate eigenfunctions over all dimensions. The number of D-tuples is given by

$$m^* = \prod_{d=1}^{D} m_d, \tag{3.10}$$

where m_d is the number of basis functions for the dimension d. Let $\mathbb{S} \in \mathbb{N}^{m^* \times D}$ be the matrix of all those D-tuples. For example, suppose we have $D = 3$ dimensions

and use $m_1 = 2$, $m_2 = 2$ and $m_3 = 3$ eigenfunctions and eigenvalues for the first, second and third dimension, respectively. Then, the number of multivariate eigenfunctions and eigenvalues is $m^* = m_1 \cdot m_2 \cdot m_3 = 12$ and the matrix $\mathbb{S} \in \mathbb{N}^{12\times 3}$ is given by

$$\mathbb{S} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 1 & 3 \\ 1 & 2 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 3 \\ 2 & 1 & 1 \\ 2 & 1 & 2 \\ 2 & 1 & 3 \\ 2 & 2 & 1 \\ 2 & 2 & 2 \\ 2 & 2 & 3 \end{bmatrix}$$

Each multivariate eigenfunction ϕ_j^* corresponds to the product of the univariate eigenfunctions whose indices corresponds to the elements of the D-tuple $\mathbb{S}_{j\cdot}$, and each multivariate eigenvalue $\boldsymbol{\lambda}_j^*$ is a D-vector with elements that are the univariate eigenvalues whose indices correspond to the elements of the D-tuple $\mathbb{S}_{j\cdot}$. Thus, for $\boldsymbol{x} = \{x_d\}_{d=1}^D \in \Omega$ and $j = 1, \ldots, m^*$, we have

$$\boldsymbol{\lambda}_j^* = \left\{\lambda_{\mathbb{S}_{jd}}\right\}_{d=1}^D = \left\{\left(\frac{\pi \mathbb{S}_{jd}}{2L_d}\right)^2\right\}_{d=1}^D. \tag{3.11}$$

$$\phi_j^*(\boldsymbol{x}) = \prod_{d=1}^D \phi_{\mathbb{S}_{jd}}(x_d) = \prod_{d=1}^D \sqrt{\frac{1}{L_d}} \sin\left(\sqrt{\lambda_{\mathbb{S}_{jd}}}(x_d + L_d)\right) \tag{3.12}$$

The approximate covariance function is then represented as

$$k(\boldsymbol{x}, \boldsymbol{x}') \approx \sum_{j=1}^{m^*} S_\theta^*\left(\sqrt{\boldsymbol{\lambda}_j^*}\right) \phi_j^*(\boldsymbol{x}) \phi_j^*(\boldsymbol{x}'),$$

where S_θ^* is the spectral density of the D-dimensional covariance function (see Section 3.4.1. We can now write the approximate series expansion of the multivariate function f as,

$$f(\boldsymbol{x}) \approx \sum_{j=1}^{m^*} \left(S_\theta^*\left(\sqrt{\boldsymbol{\lambda}_j^*}\right)\right)^{1/2} \phi_j^*(\boldsymbol{x}) \beta_j, \tag{3.13}$$

where, again, $\beta_j \sim \text{Normal}(0, 1)$. The computational cost, in learning the covari-

ance function hyperparameters, of multivariate HSGPs scales as $O(nm^* + m^*)$, where n is the number of observations and m^* is the number of multivariate basis functions. Although this still implies linear scaling in n, the approximation is more costly than in the univariate case, as m^* is the product of the number of univariate basis functions over the input dimensions and grows exponentially with respect to the number of dimensions.

3.6 The accuracy of the approximation

The accuracy and speed of the HSGP model depends on several interrelated factors, most notably on the number of basis functions and on the boundary condition of the Laplace eigenfunctions. Furthermore, appropriate values for these factors will depend on the non-linearity of the estimated function, which is in turn characterized by the lengthscale of the covariance function. In this section, we analyze the effects of the number of basis functions and the boundary condition on the approximation accuracy. We present recommendations on how they should be chosen and diagnostics to check the accuracy of the obtained approximation.

Ultimately, these recommendations lie on the relationships among the number of basis functions, the boundary factor and the lengthscale of the function, which depend on the particular choice of the kernel function. In this work, we built these relationships for the squared exponential covariance function and Matern ($\nu = 3/2$) covariance function in the present Section, and for the periodic squared exponential covariance function in Section 3.7. For other kernels, the relationships will be slightly different, in function of mainly the smoothness or wiggliness of the kernel effects.

3.6.1 Dependency on the number of basis functions and the boundary condition

As explained in Section 3.5, the approximation of the covariance function is a series expansion of eigenfunctions and eigenvalues of the Laplace operator in a given domain Ω, for instance in a one-dimensional input space $\Omega = [-L, L] \subset \mathbb{R}$:

$$k(\tau) = \sum_{j=1}^{\infty} S_\theta \left(\sqrt{\lambda_j}\right) \phi_j(\tau)\phi_j(0),$$

where L describes the boundary condition, j is the index for the eigenfunctions and eigenvalues, and $\tau = x - x'$ is the difference between two input values x and x' in Ω. The eigenvalues λ_j and eigenfunctions ϕ_j are given in equations (3.7) and (3.8) for the unidimensional case and in equations (3.11) and (3.12) for the

multidimensional case. The number of basis functions can be truncated at some finite positive value m such that the difference between the densities of the exact and approximate covariance functions is less than a predefined threshold $\varepsilon > 0$:

$$\int k(\tau)\, d\tau - \int \sum_{j=1}^{m} S_\theta\left(\sqrt{\lambda_j}\right) \phi_j(\tau)\phi_j(0)\, d\tau < \varepsilon. \tag{3.14}$$

The finite number m of basis functions in the approximation needed to satisfy equation (3.14) depends on the non-linearity of the function to be learned, that is on its lengthscale ℓ, which constitutes a hyperparameter of the GP. The approximation also depends on the boundary L, which will affect its accuracy especially near the boundaries. As we will see later on, L will also influence the number of basis functions required in the approximation. In the present study, we will set L an extension of the desired covariate input domain Ψ. Without loss of generality, we can assume Ψ to be symmetric around zero, that is $\Psi = [-S, S] \subset \mathbb{R}$. We now define L as

$$L = c \cdot S, \tag{3.15}$$

where S (for $S > 0$) represents the half-range of the input space, and c (for $c \geq 1$) is the proportional extension factor. In the following, we will to refer to c as the boundary factor of the approximation. The boundary factor can also be regarded as the boundary L normalized by the half-range S of the input space.

We start with an illustration on how the number of basis functions m and boundary factor c influences the accuracy of the HSGP approximations, separately. For this purpose, a set of noisy observations are drawn from an exact GP model with lengthscale $\ell = 0.3$ and marginal variance $\alpha = 1$, using input values from the zero-mean input domain with half-range $S = 1$. Several HSGP models with varying m and L are fitted to this data. In this example, the lengthscale and marginal variance parameters used in the HSGPs are fixed to the true values of the data-generating model. Figures 3.1 and 3.2 illustrate the individual effects of m and c, respectively, on the posterior predictions of the estimated function and on the covariance function itself. For c fixed to a large enough value, Figure 3.1 shows clearly how m affects the accuracy on the approximation and the non-linearity of the estimated function, in the sense that fewer basis functions inaccurately imply larger lengthscales and consequently more linear functional forms. The higher the wiggliness of the function to be estimated, the more basis functions will be required. If m fixed to a large enough value, Figure 3.2 shows that c mainly affects the approximation near the boundaries as well as covariances at long distances.

Next, we will focus on analyzing the interaction effects between these m and c on the performance of the approximation. The lengthscale and marginal variance

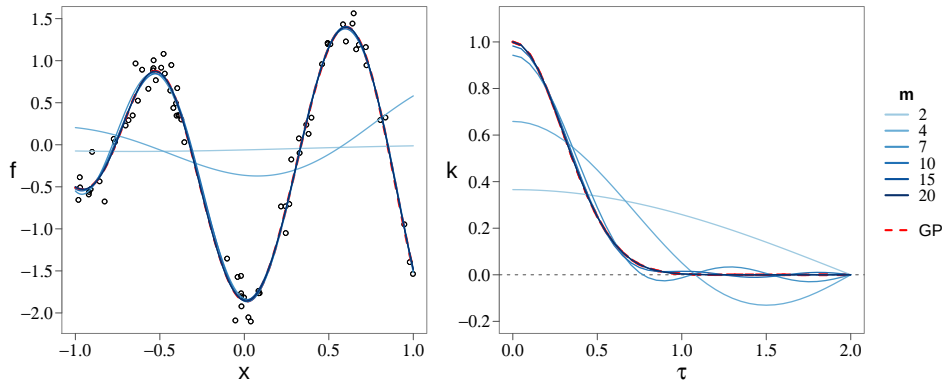


Figure 3.1: Mean posterior predictive functions (left) and covariance functions (right) of both the regular GP model (dashed red line) and the HSGP model for different number of basis functions m, with the boundary factor fixed to a large enough value.

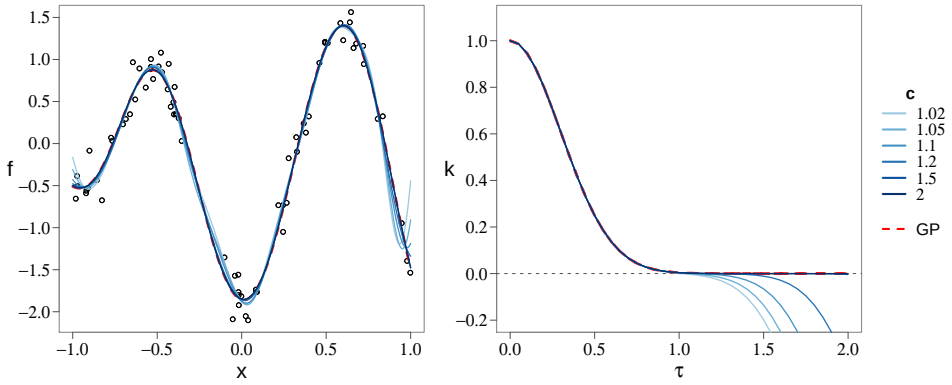


Figure 3.2: Mean posterior predictive functions (left) and covariance functions (right) of both the regular GP model (dashed red line) and the HSGP model for different values of the boundary factor c, with a large enough fixed number of basis functions.

will no longer be fixed but rather estimated in both regular GP and HSGP models. Figure 3.3 shows the functional posterior predictions and the covariance function obtained after fitting the data, for varying m and c. Figure 3.4 shows the root mean square error (RMSE) of the HSGP models, computed against the regular GP model. Figure 3.5 shows the estimated lengthscale and marginal variance for the regular GP model and the HSGP models. Looking at the RMSEs in Figure 3.4, we can conclude that the optimal choice in terms of precision and computations would be $m = 15$ basis functions and a boundary factor between $c = 1.5$ and $c = 2.5$. Further, the choice of $m = 10$ and $c = 1.5$ could still be an accurate enough choice. We may also come to the same conclusion by looking at the posterior predictions and covariance function plots in Figure 3.3. From these results, some general conclusions may be drawn:

- As c increases, m has to increase as well (and vice versa).
- There exists a minimum c below which a close approximation will never be achieved regardless of m.

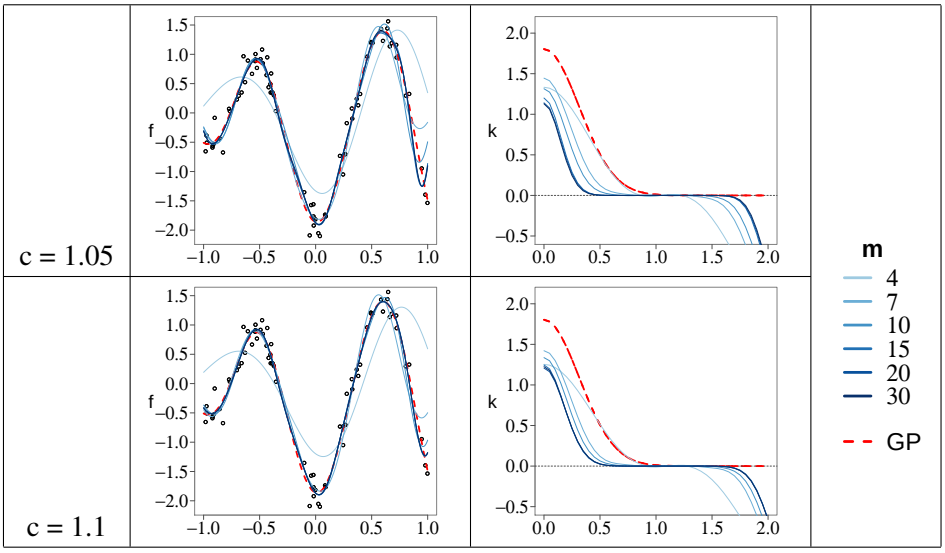


Figure 3.3: (continued on next page)

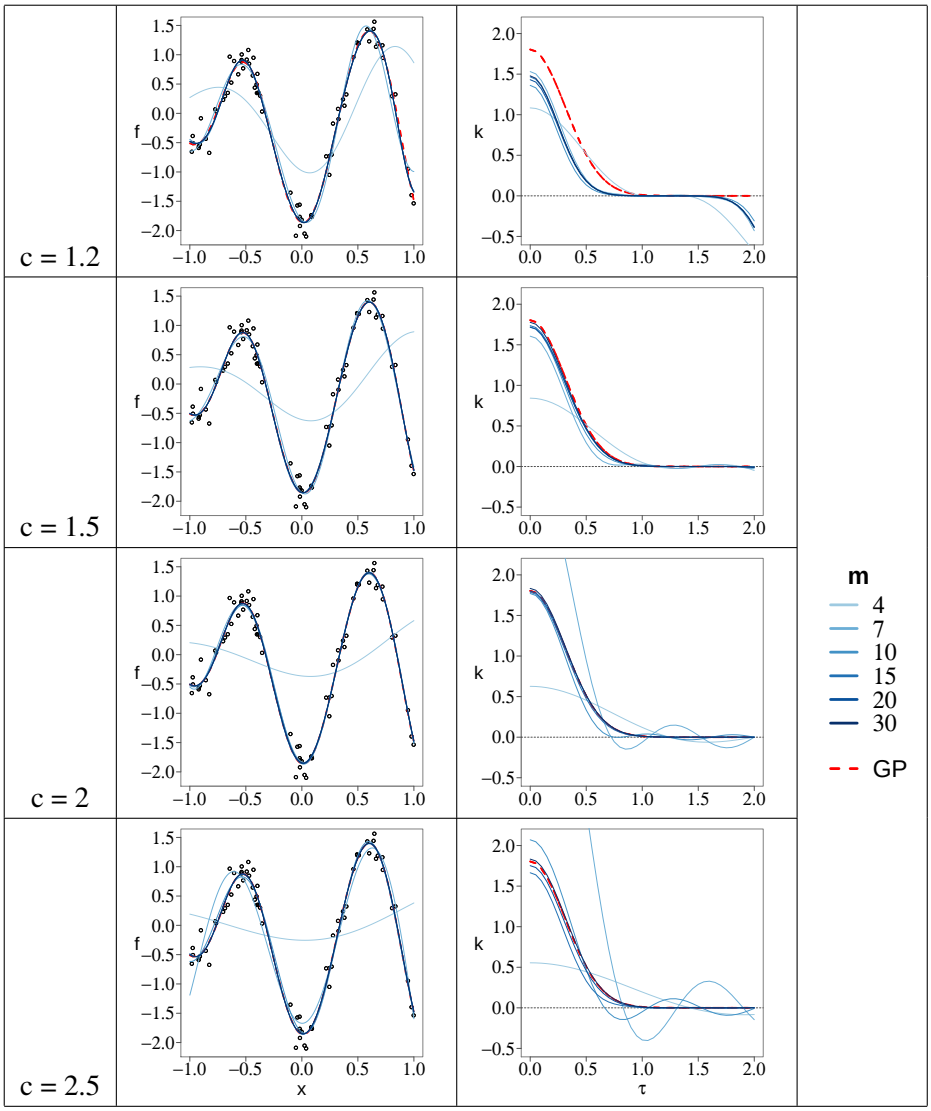


Figure 3.3: Mean posterior predictive functions (left) and covariance functions (right) of both the regular GP model and the HSGP model for different number of basis functions m and for different values of the boundary factor c.

Additionally, there is a clear relation of the number of basis functions m and the boundary factor c with the lengthscale ℓ of the approximated function. Figures 3.6 and 3.7 depicts how these three factors interact with each other in relation to a close

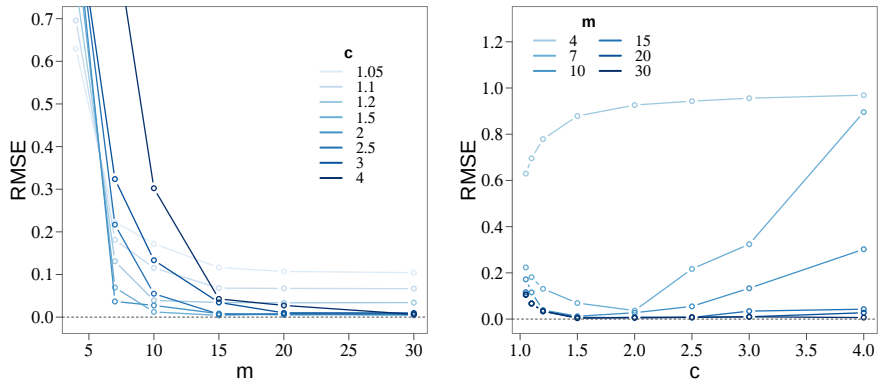


Figure 3.4: Root mean square error (RMSE) of the proposed HSGP models computed against the regular GP model. RMSE versus the number of basis functions m and for different values of the boundary factor c (left). RMSE versus the boundary factor c and for different values of the number of basis functions m (right).

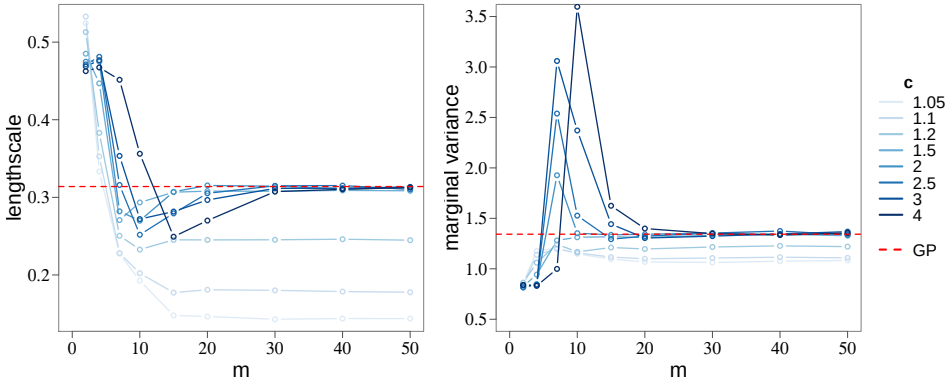


Figure 3.5: Estimated lengthscale (left) and marginal variance (right) parameters of both regular GP and HSGP models, plotted versus the number of basis functions m and for different values of the boundary factor c.

approximation of the HSGP model, in the cases of a GP with squared exponential covariance function and Matérn (ν=3/2) covariance function, respectively, and a single input dimension. More precisely, for a given GP model (with a squared exponential covariance function) with lengthscale ℓ and given a boundary factor c, Figure 3.6 shows the minimum m required to achieve a close approximation in terms of satisfying equation (3.14). Similarly for Figure 3.7 in the case of a Matérn (ν=3/2) covariance function. We have considered an approximation to be a close enough when the difference between densities of the approximate covariance function and the exact covariance function, ε in equation (3.14), is below 1% of the density of the exact covariance function,

$$\frac{\varepsilon}{\int k(\tau)\, d\tau} < 0.01.$$

Alternatively, these figures could be understood as providing the minimum c that we should use for given ℓ and m. Of course, we may also read it as providing the minimum ℓ that can be closely approximated given m and c. We obtain the following main conclusions:

- As ℓ increases, c and m required for a close enough approximation decrease.
- The lower c, the smaller m can and ℓ must be to achieve a close approximation.
- For a given ℓ there exist a minimum c under which a close approximation is never going to be achieved regardless of m. This fact can be appreciated in the Figure as the contour lines which represent c have an end in function of ℓ (valid c are restricted in function of ℓ).

As stated above, Figures 3.6 and 3.7 provide the minimum lengthscale that can be closely approximated given m and c. This information serves as a powerful diagnostic tool in determining if the obtained accuracy is acceptable. As the lenghscale ℓ controls the wiggliness of the functional relationship, it strongly influences the difficulty of obtaining accurate inference about the function from the data. Basically, if the lengthscale estimate is accurate, we can expect the HSGP approximation to be accurate as well. Thus, having obtained an estimate $\hat{\ell}$ of ℓ from the HSGP model based on prespecified m and c, we can check whether or not $\hat{\ell}$ exceeds the minimum lengthscale provided in Figure 3.6. If $\hat{\ell}$ exceeds this recommended minimum lengthscale, the approximation should be close enough. If, however, it does not exceed it, the approximation may be inaccurate and m should be increased or c decreased. We may also use this diagnostic in an iterative procedure. Starting from some initial guess of ℓ, we can choose initial values for m and c and fit an

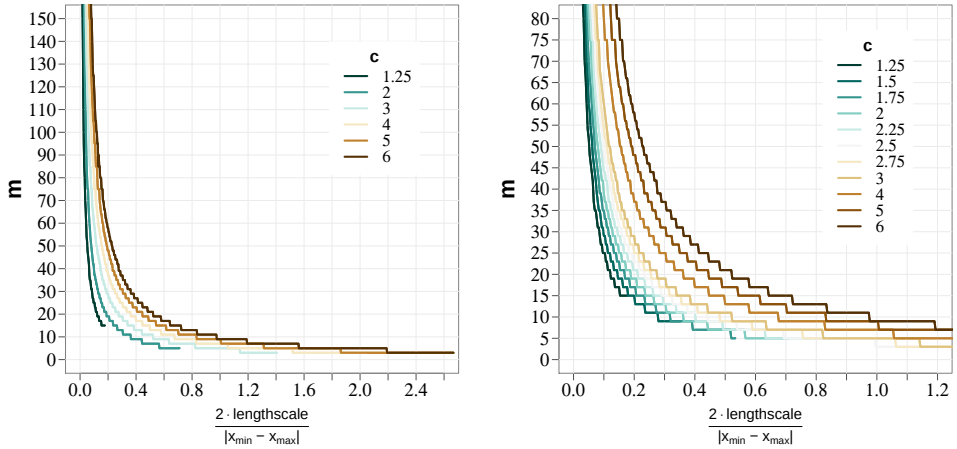


Figure 3.6: Relation among the minimum number of basis functions m, the boundary factor c ($c = \frac{L}{S}$) and the lengthscale normalized by the half-range of the data ($\frac{\ell}{S}$), in the case of a squared exponential covariance function. The right-side plot is a zoom in of the left-side plot.

HSGP model, then check the approximation accuracy, and, if not accurate enough because the estimated $\hat{\ell}$ is below the minimum lengthscale provided by Figure 3.6, repeat the process while increasing m or decreasing c. Note that, as commented before, c can not be decreased as much as desired because it is restricted to the lengthscale.

If we look back to the conclusions drawn from Figures 3.4 and 3.5, where $m = 10$ basis functions and a boundary factor of $c = 1.5$ were enough to closely approximate a function with $\ell = 0.3$, we can recognize that these conclusions also matches those obtained from Figure 3.6.

Figures 3.6 and 3.7 were build for a GP with a unidimensional covariance function, which result in a surface depending on three variables, m, c and ℓ. An equivalent figure for a GP model with a two-dimensional covariance function would result in a surface depending on four variables, m, c, ℓ_1 and ℓ_2, which can not be graphically represented. More precisely, in the multi-dimensional case, whether the approximation is close enough might depend only on the ratio between wiggliness in every dimensions. For instance, in the two-dimensional case it would depend on the ratio between ℓ_1 and ℓ_2 and could be graphically represented. Future research will focus on building useful graphs or analytical models that provide these relations in multi-dimensional cases. However, as an approximation, we can use the unidimensional GP conclusions in Figures 3.6 and 3.7 to check the accuracy by

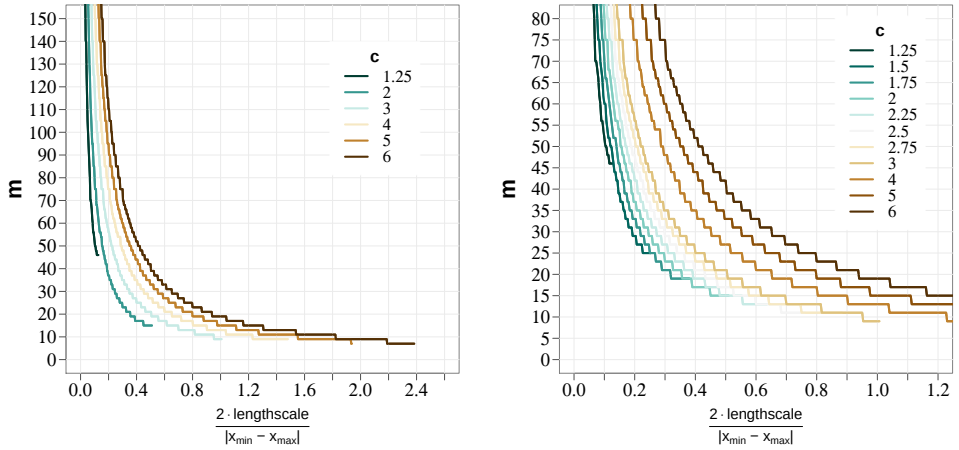


Figure 3.7: Relation among the minimum number of basis functions m, the boundary factor c ($c = \frac{L}{S}$) and the lengthscale normalized by the half-range of the data ($\frac{\ell}{S}$), in the case of a Matèrn (ν=3/2) covariance function. The right-side plot is a zoom in of the left-side plot.

analyze individually the different dimensions of a multidimensional GP model.

3.6.2 Comparing lengthscale estimates

In this example, we make a comparison of the lengthscale estimates obtained from the regular GP and HSGP models. We also have a look at those recommended minimum lengthscales provided by Figure 3.6.

For this analysis, we will use various datasets consisting of noisy draws from a GP prior model with a squared exponential covariance function and varying lengthscale values. Different values of the number of basis functions m are used when estimating the HSGP models, and the boundary factor c is set to a valid and optimum value in every case.

Figure 3.8 shows the posterior predictions of both regular GP and HSGP models fitted to those datasets. The lengthscale estimates as obtained by regular GP and HSGP models are depicted in Figure 3.9. As noted previously, an accurate estimate of the lengthscale can be a good indicator of a close approximation of the HSGP model to the regular GP model. Further, Figure 3.10 shows the root mean square error (RMSE) of the HSGP models, computed against the regular GP models, as a function of the lengthscale and number of basis functions.

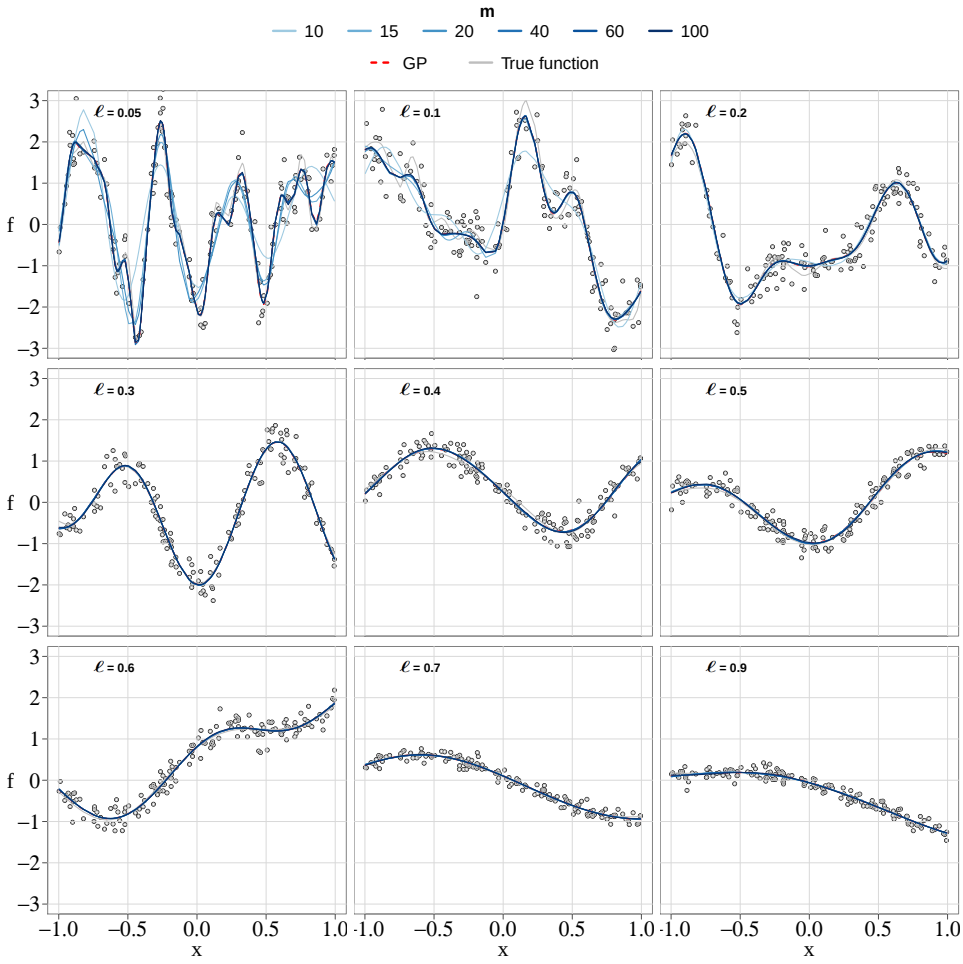


Figure 3.8: Mean posterior predictions of both the regular GP model (dashed red line) and the HSGP model for different number of basis functions m, fitted over various datasets drawn from squared exponential GP models with different characteristic lengthscales (ℓ) and same marginal variance (α) as the data-generating functions (*true function*).

Comparing the accuracy of the lengthscale in Figure 3.9 to the RMSE in Figure 3.10, we see that they agree closely with each other for medium lengthscales. That is, a good estimation of the lengthscale implies a small RMSE. This is no longer true for very small or large lengthscales. In small lengthscales, even very small inaccuracies may have a strong influence on the posteriors predictions and thus on

the RMSE. In large lengthscales, larger inaccuracies change the posterior predictions only little and may thus not yield large RMSEs. The dashed black line in Figure 3.9 represents the minimum lengthscale that can be closely approximated under the given condition, according to the results presented in Figure 3.6. We observe that whenever the estimated lengthscale exceeds the minimially estimable lengthscale, the RMSE of the posterior predictions is small (see Figure 3.10). Conversely, when the estimated lengthscale is smaller than the minimally estimable one, the RMSE becomes very large.

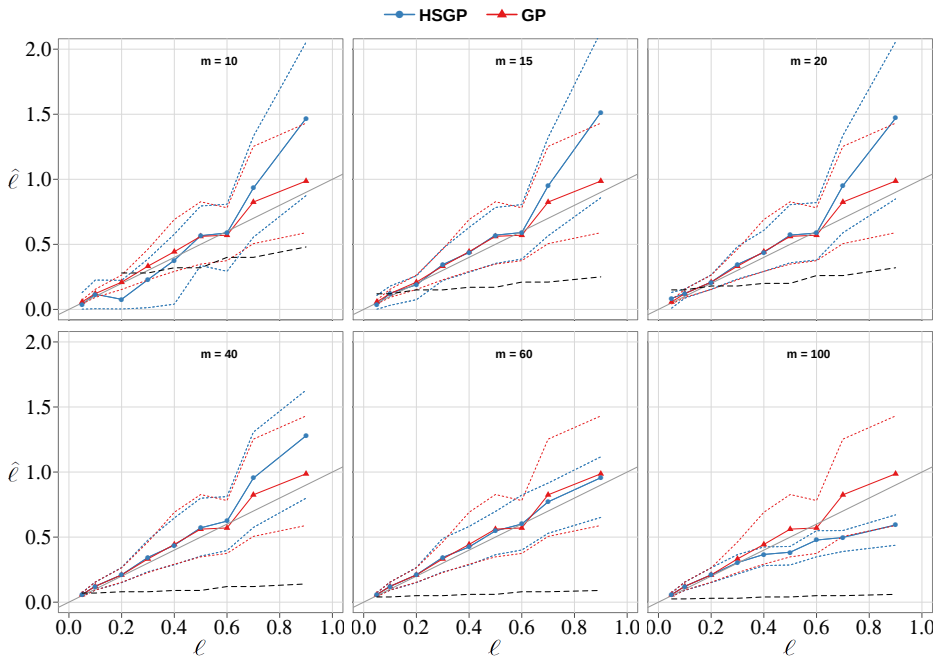


Figure 3.9: Data-generating functional lengthscales (ℓ), of the various datasets illustrated in Figure 3.8, versus the corresponding lengthscale estimates ($\hat{\ell}$) from the regular GP and HSGP models. 95% credible intervals of the lengthscale estimates are plotted as dot lines. The different plots represent the use of different number of basis functions m in the HSGP model. The dashed black line represents the recommended minimum lengthscales provided by Figure 3.6 that can be closely approximated by the HSGP model in every case.

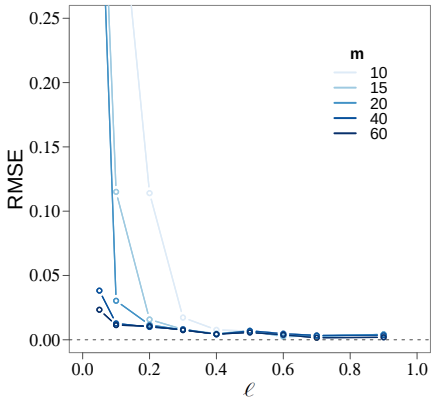


Figure 3.10: RMSE of the HSGP models with different number of basis functions m, for the various datasets with different wiggly effects (ℓ).

3.7 Low-rank Gaussian process with a periodic covariance function

A GP model with a periodic covariance function does no fit under the framework of the HSGP approximation covered so far in this study. However, it do has a low-rank representation. In this section, we first give a brief presentation of the results from Solin and Särkkä [2014], where the authors obtain an approximate linear representation of a periodic squared exponential covariance function based on expanding the periodic covariance function into a series of stochastic resonators. Secondly, we analyze the accuracy of this approximation and, finally, we derive the GP model with this approximate periodic square exponential covariance function.

The periodic squared exponential covariance function takes the form

$$k(\boldsymbol{\tau}) = \sigma^2 \exp\Big(- \frac{2\sin^2\left(\omega_0 \frac{\boldsymbol{\tau}}{2}\right)}{\ell^2}\Big), \tag{3.16}$$

where σ^2 is the magnitude scale of the covariance, ℓ is the characteristic lengthscale of the covariance, and ω_0 is the angular frequency defining the periodicity.

In Solin and Särkkä [2014], the authors come to a cosine series expansion for the periodic covariance function (3.16) as follows,

$$k(\tau) = \sigma^2 \sum_{j=0}^{J} \tilde{q}_j^2 \cos(j\omega_0\tau), \tag{3.17}$$

which comes basically from a Taylor series representation of the periodic covariance function. The coefficients $\tilde{q}_j^2$ of the previous expression are

$$\tilde{q}_j^2 = \frac{2}{\exp\left(\frac{1}{\ell^2}\right)} \sum_{j=0}^{\lfloor \frac{J-j}{2} \rfloor} \frac{(2\ell^2)^{-j-2}}{(j+i)!i!}, \tag{3.18}$$

where $j = 1, 2, \cdots, J$, and $\lfloor \cdot \rfloor$ denotes the floor round-off operator. For the index $j = 0$, the coefficient is

$$\tilde{q}_0^2 = \frac{1}{2}\frac{2}{\exp\left(\frac{1}{\ell^2}\right)} \sum_{j=0}^{\lfloor \frac{J-j}{2} \rfloor} \frac{(2\ell^2)^{-j-2}}{(j+i)!i!}. \tag{3.19}$$

Note that the covariance in equation (3.17) is a Jth order truncation of a Taylor series representation. As Solin and Särkkä [2014] argue, this approximation converges to equation (3.16) when $J \to \infty$.

An upper bounded approximation to the coefficients $\tilde{q}_j^2$ and $\tilde{q}_0^2$ can be obtained by taking the limit $J \to \infty$ in the sub-sums in the corresponding equations (3.18) and (3.19), and thus leading to the following variance coefficients:

$$\begin{aligned} \tilde{q}_j^2 &= \frac{2\mathrm{I}_j(\ell^{-2})}{\exp\left(\frac{1}{\ell^2}\right)}, \\ \tilde{q}_0^2 &= \frac{\mathrm{I}_0(\ell^{-2})}{\exp\left(\frac{1}{\ell^2}\right)}, \end{aligned} \tag{3.20}$$

for $j = 1, 2, \cdots, J$, and where the $\mathrm{I}_\alpha(z)$ is the modified Bessel function [Abramowitz and Stegun, 1970] of the first order of α. This approximation implies that the requirement of a valid covariance function is relaxed and only an optimal series approximation is required [Solin and Särkkä, 2014]. A more detailed explanation and mathematical proofs of this approximation of a periodic covariance function can be found in Solin and Särkkä [2014].

In order to assess the accuracy of this representation as a function of the number of cosine terms J considered in the approximation, an empirical evaluation is carried out in a similar way than that in Section 3.6 of this work. Thus, Figure 3.11 shows the minimum number of terms J required to achieve a close approximation to the exact periodic squared exponential kernel as a function of the lengthscale of the kernel. We have considered an approximation to be close enough in terms of satisfying equation (3.14) with $\varepsilon = 0.5\%$. Notice that since this is a series expansion of sinusoidal functions, the approximation does not depend on any boundary condition.

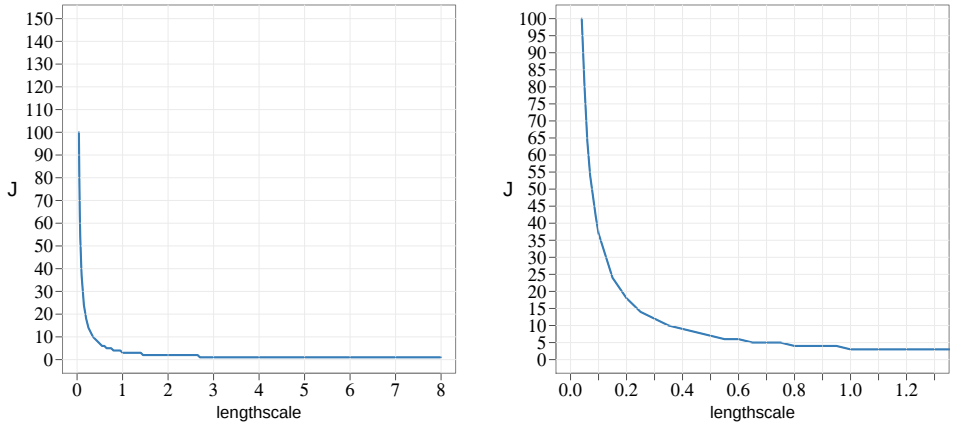


Figure 3.11: Relation among the minimum number of terms J in the approximation and the lengthscale (ℓ) of the periodic squared exponential covariance function. The right-side plot is a zoom in of the left-side plot.

The function values of a GP model with this low-rank representation of the periodic exponential covariance function can be easily derived. Considering the identity

$$\cos(j\omega_0(x-x')) = \cos(j\omega_0 x)\cos(j\omega_0 x') + \sin(j\omega_0 x)\sin(j\omega_0 x'),$$

the covariance $k(\tau)$ in equation (3.17) can be re-writting as

$$k(x,x') = \sigma^2\Big(\sum_{j=0}^{J}\tilde{q}_j^2\cos(j\omega_0 x)\cos(j\omega_0 x') + \sum_{j=1}^{J}\tilde{q}_j^2\sin(j\omega_0 x)\sin(j\omega_0 x')\Big) \quad (3.21)$$

where $\tau = x - x'$. With this approximation for the periodic squared exponential covariance function $k(x,x')$, the approximate GP model $f(x) \sim \mathcal{GP}(0, k(x,x')$ equivalently leads to a linear representation of $f(\cdot)$ via

$$f(x) \approx \sigma\Big(\sum_{j=0}^{J}\tilde{q}_j\cos(j\omega_0 x)\beta_j + \sum_{j=1}^{J}\tilde{q}_j\sin(j\omega_0 x)\beta_{J+1+j}\Big), \quad (3.22)$$

where $\beta_j \sim \text{Normal}(0,1)$, with $j = 1,\ldots,2J+1$. The cosine $\cos(j\omega_0 x)$ and sinus $\sin(j\omega_0 x)$ terms do not depend on the covariance hyperparameters ℓ. The only dependence on the hyperparameter ℓ is through the coefficients $\tilde{q}_j$, which are

J-dimensional. The computational cost of this approximation scales as $O\big(n(2J+1)+(2J+1)\big)$, where n is the number of observations and J the number of cosine terms.

The parameterization in equation (3.22) is naturally in the non-centered parameterization form with independent prior distribution on β_j, which makes the posterior inference easier.

3.8 Case study I: 1D Simulated data

This example consists of a simulated dataset with $n = 250$ ($i = 1, \ldots, n$) single draws from a Gaussian process prior with a Matérn(ν=3/2) covariance function and hyperparameters marginal variance $\alpha = 1$ and lengthscale $\ell = 0.15$, with corresponding inputs values $\boldsymbol{x} = (x_1 \ldots, x_n)$ with $x_i \in [-1, 1] \subset \mathbb{R}$. To form the final noisy dataset $\boldsymbol{y}$, Gaussian noise $\sigma = 0.2$ was added to the GP draws.

The regular GP model for fitting this simulated dataset $\boldsymbol{y}$ can be written as follows,

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{f} + \boldsymbol{\epsilon} \\ \boldsymbol{\epsilon} &\sim \mathcal{N}(0, \sigma^2 I) \\ f(x) &\sim \mathcal{GP}(0, k(x, x', \theta)), \end{aligned}$$

where I represents the identity matrix and $\boldsymbol{f} = \{f(x_i)\}_{i=1}^n$ represents the underlying function values to the noisy data. The previous formulation corresponds to the latent form of a GP model. The function $f : \mathbb{R} \to \mathbb{R}$ is a GP prior with a Matérn(ν=3/2) covariance function k. Saying that the function $f(\cdot)$ follows a GP model is equivalent to say that $\boldsymbol{f}$ is multivariate Gaussian distributed with covariance matrix K, where $K_{ij} = k(x_i, x_j, \theta)$, with $i, j = 1, \ldots, n$.

A more computationally efficient formulation of a GP model with Gaussian likelihood, and for probabilistic inference using sampling methods such as HMC, would be its marginalized form,

$$\boldsymbol{y} \sim \mathcal{N}(0, K + \sigma^2 I),$$

where the function values $\boldsymbol{f}$ have been integrated out, yielding a lower-dimensional parameter space over which to do inference, reducing the time of computation and improving the sampling and the effective number of samples.

In the HSGP model, the latent function values $f(x)$ are approximated as in

equation (3.9),

$$f(x) \approx \sum_{j=1}^{m} \left(S(\sqrt{\lambda_j}) \right)^{1/2} \phi_j(x) \beta_j,$$

with the spectral density S as a function of $\sqrt{\lambda_j}$,

$$S(\sqrt{\lambda_j}) = \alpha^2 \frac{4\sqrt{3}^3}{\ell^3} (\frac{3}{\ell^2} + \lambda_j)^{-2},$$

and eigenvalues λ_j and eigenfunctions ϕ_j ,

$$\lambda_j = \left(\frac{j\pi}{2L} \right)^2,$$
$$\phi_j(x) = \sqrt{\frac{1}{L}} \sin \left(\sqrt{\lambda_j}(x + L) \right).$$

In the previous equations, L is the boundary and m the number of basis functions. The parameters β_j are $\mathcal{N}(0, 1)$ distributed, and α and ℓ are the marginal variance and lengthscale parameters, respectively, of the approximate covariance function.

In order to do model comparison, in addition to the regular GP model and HSGP model, a splines-based model is also fitted using the thin plate regression splines approach in Wood [2003] and implemented in the R-package *mgcv* [Wood, 2015]. A Bayesian approach is used to fit this spline model using the R-package *brms* [Bürkner et al., 2017].

Figure 3.12 shows the posteriors predictive distributions of the three models, the regular GP, the HSGP with $m = 80$ basis functions and boundary factor $c = 1.2$ $\big(L = c \cdot 1 = 1.2$ (equation (3.15))$\big)$, and the splines model with 80 knots. The true data-generative function and the noisy observations are also plotted. The sample observations are plotted as circles and the out-of-sample or test data, which have not been taking part on training the models, are plotted as crosses. The test data located at the extremes of the plot are used for assessing model extrapolation, and the test data located in the middle are used for assessing model interpolation. The posteriors of the three models, regular GP, HSGP and Splines, are pretty similar within the interpolation input space. However, when extrapolating the splines solution clearly differs from the regular GP and HSGP models as well as the actual observations.

In order to assess the performance of the models as a function of the number of basis functions and number of knots, different models with different number of basis functions for the HSGP model, and different number of knots for the splines model, have been fitted. Figure 3.13 shows the standardized root mean squared error (SRMSE) for interpolation and extrapolating data as a function of the number

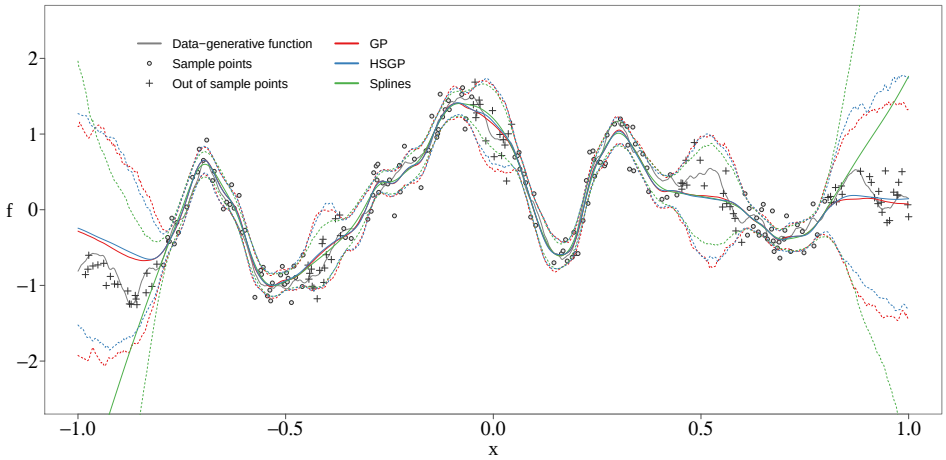


Figure 3.12: Posterior predictive means of the proposed HSGP model, the regular GP model, and the Splines model. 95% credible intervals are plotted as dashed lines.

of basis functions and knots. The SRMSE is computed against the data-generating function. From Figures 3.12 and 3.13, it can be seen a close approximation of the HSGP model to the regular GP model for interpolating and extrapolating data. However, the splines model does not extrapolate data properly. Both models show roughly similar interpolating performance.

Figure 3.14 shows computational times, in seconds per iteration (iteration of the HMC sampling method), as a function of the number of basis functions m, for the HSGP model, and knots, for the splines model. The computational time is represented in the y-axis of the figure, which is in a logarithmic scale. The HSGP model is on average roughly 400 times faster than the regular GP, in the particular case of applying over this dataset.

The Stan model codes for the exact GP, the approximate GP and the splines models of this case study can be found through the following link to the author's GitHub repository:

`https://github.com/gabriuma/Doctoral_thesis/tree/master/Case-study-I_1D-Simulated-data`

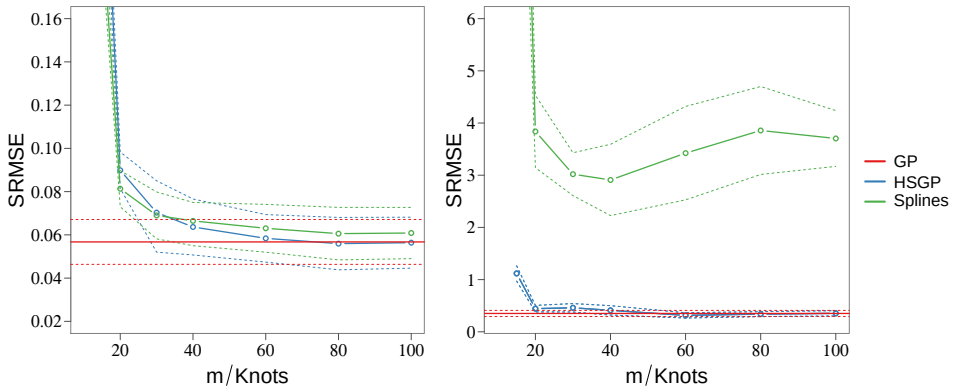


Figure 3.13: Standardized root mean square error (SRMSE) of the different methods against the data-generating function. SRMSE for interpolation (left) and SRMSE for extrapolation (right). The standard deviation of the mean of the SRMSE is plotted as dashed lines.

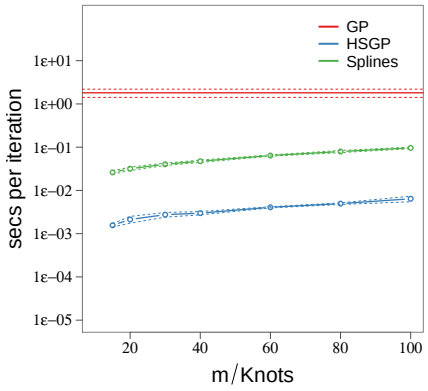


Figure 3.14: Computational time (y-axis), in seconds per iteration (iteration of the HMC sampling method), as a function of the number of basis functions m, for the HSGP model, and knots, for the Splines model. The y-axis is on a logarithmic scale. The standard deviation of the computational time is plotted as dashed lines.

3.9 Case study II: Birthday data

This example is an analysis of patterns in birthday frequencies in a dataset containing records of all births in the United States on each day during the period 1969–1988. The model decomposes the number of births along all the period in longer-term trend effects, patterns during the year, day-of-week effects, and special days effects. The special days effects cover patterns such as possible fewer births on Halloween, Christmas or new year, and excess of births on Valentine's Day or the days after Christmas (due, presumably, to choices involved in scheduled deliveries, along with decisions of whether to induce a birth for health reasons). This analysis was originally addressed in Gelman et al. [2013]. The total number of days within the period is $T = 7305$ ($t = 1, \dots, T$), then a regular GP model is unfeasible to be fitted on this dataset as we know inference scales $O(T^3)$ in covariance matrix inversion. Therefore, an approximate approach has to be used to fit a GP model on this data. We will use the HSGP model developed in this Chapter, as well as the low-rank GP model with a periodic covariance function introduced in Section 3.7 which is based on expanding the periodic covariance function into a series of stochastic resonators [Solin and Särkkä, 2014].

Let's denote y_t as the number of births of day t. The observational model is a normal model with parameters the mean function $\mu(t)$ and noise variance σ^2,

$$y_t \sim \mathcal{N}(\mu(t), \sigma^2).$$

The mean function $\mu(t)$ will be defined as an additive model in the form:

$$\mu(t) = f_1(t) + f_2(t) + f_3(t) + f_4(t). \tag{3.23}$$

The component $f_1(t)$ represents the long-term trends modeled by a GP with squared exponential covariance function,

$$f_1(t) \sim \mathcal{GP}(0, k_1), \qquad k_1(t, t') = \sigma_1^2 \exp\left(-\frac{1}{2}\frac{(t-t')^2}{\ell_1^2} \right),$$

which means the function values $\boldsymbol{f}_1 = \{f_1(t)\}_{t=1}^T$ are multivariate Gaussian distributed with covariance matrix K_1, where $K_{1_{t,s}} = k_1(t, s)$, with $t, s = 1, \dots, T$.

The component $f_2(t)$ represents the yearly smooth seasonal pattern, using a periodic squared exponential covariance function (with period 365.25 to match the average length of the year) in a GP model,

$$f_2(t) \sim \mathcal{GP}(0, k_2), \qquad k_2(t, t') = \sigma_2^2 \exp\left(-\frac{2\sin^2(\pi(t-t')/365.25}{\ell_2^2} \right),$$

which means the function values $\boldsymbol{f}_2 = \{f_2(t)\}_{t=1}^T$ are multivariate Gaussian distributed with covariance matrix K_2, where $K_{2_{t,s}} = k_2(t,s)$, with $t, s = 1, \ldots, T$.

The component $f_3(t)$ represents the weekly smooth pattern using a periodic squared exponential covariance function (with period 7 of length of the week) in a GP model,

$$f_3(t) \sim \mathcal{GP}(0, k_3), \qquad k_3(t,t') = \sigma_3^2 \exp\left(- \frac{2\sin^2(\pi(t-t')/7}{\ell_3^2} \right),$$

which means the function values $\boldsymbol{f}_3 = \{f_3(t)\}_{t=1}^T$ are multivariate Gaussian distributed with covariance matrix K_3, where $K_{3_{t,s}} = k_3(t,s)$, with $t, s = 1, \ldots, T$.

The component $f_4(t)$ represents the special days effects, modeled as a horse-shoe prior model [Piironen et al., 2017]:

$$f_4(t) \sim \mathcal{N}(0, \lambda_t^2 \tau^2), \qquad \lambda_t^2 \sim \mathcal{C}^+(0,1).$$

A horse-shoe prior allows for sparse distributed effects. Its global parameter τ pulls all the weights (effects) globally towards zero, while the thick half-Cauchy tails for the local scales λ_t allow some of the weights to escape the shrinkage. Different levels of sparsity can be accommodated by changing the value of τ: with large τ all the variables have very diffuse priors with very little shrinkage towards zero, but letting $\tau \to 0$ will shrink all the weights $f_4(t)$ to zero [Piironen and Vehtari, 2016].

GP priors have been defined over the components $f_1(t)$, $f_2(t)$ and $f_3(t)$. Then, low-rank representations of the GP priors have to be used in the modeling and inference. The component $f_1(t)$ will be approximated using the HSGP model. Thus, the function values $f_1(t)$ are approximated as in equation (3.9), with the squared exponential spectral density S as in equation (3.3), and eigenvalues λ_j and eigenfunctions ϕ_j as in equations (3.7) and (3.8).

The year effects $f_2(t)$ and week effects $f_3(t)$, as they use a periodic covariance function, they do no fit under the main framework of the HSGP approximation covered in this chapter. However, they do have a representation based on expanding periodic covariance functions into a series of stochastic resonators (Section 3.7). Thus, the functions $f_2(t)$ and $f_3(t)$ are approximated as in equation 3.22, with variance coefficients $\tilde{q}_j^2$ as in equation 3.20.

For the component $f_1(t)$, $m = 30$ basis functions and a boundary factor $c = 1.5$ were used. The lengthscale estimate $\hat{\ell}_1$, for this component, normalized by half of the range of the input x_1, is bigger than the minimum lengthscale reported by Figure 3.6 as a function of m and c. Which means that the used number of basis functions and boundary factor are suitable values for modeling accurately the input effects.

For the components $f_2(t)$ and $f_3(t)$, $J = 10$ cosine terms were used. The

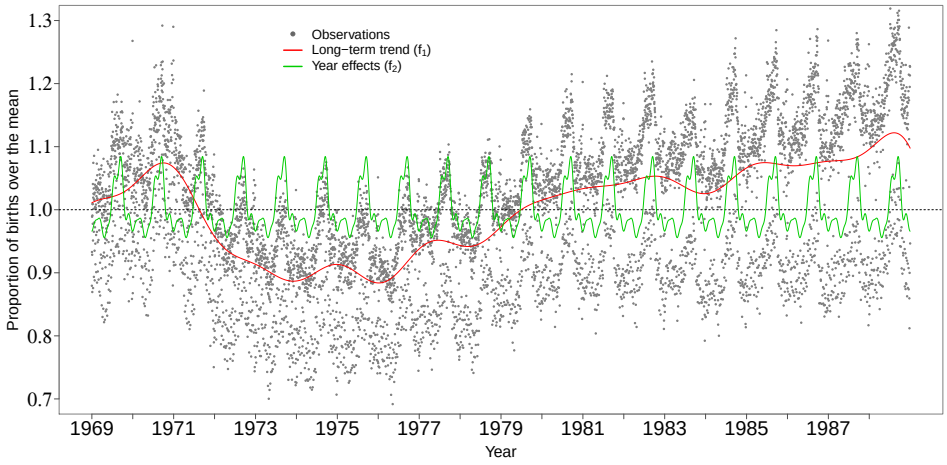


Figure 3.15: Posterior means of the long-term trend ($f_1(\cdot)$) and year effects pattern ($f_2(\cdot)$) for the whole series.

lengthscales estimates $\hat{\ell}_2$ and $\hat{\ell}_3$, for the GP components $f_2(t)$ and $f_3(t)$, respectively, are bigger than the minimum lengthscale reported by Figure 3.11 as function of the number of cosine terms J, which means that the approximations are accurate enough.

Figure 3.15 shows the posterior means of the long-term trend $f_1(t)$ and year patterns $f_2(t)$ for the whole period, jointly with the observed data. Figure 3.16 show the process for one year (1972) only. In this figure, the special days effects $f_4(t)$ in the year can be clearly represented. The posterior means of the the function $\mu(t)$ and the components $f_1(t)$ (long-term trend) and $f_2(t)$ (year pattern) are also plotted in this Figure 3.16. Figure 3.17 show the process in the month of January of 1972 only, where the week pattern $f_3(t)$ can be clearly represented. The mean of the the function $\mu(t)$ and components $f_1(t)$ (long-term trend), $f_2(t)$ (year pattern) and $f_4(t)$ (special-days effects) are also plotted in this Figure 3.17.

The Stan model code for the approximate GP model of this case study can be found in:

`https://github.com/gabriuma/Doctoral_thesis/tree/master/Case-study-II_Birthday-data`

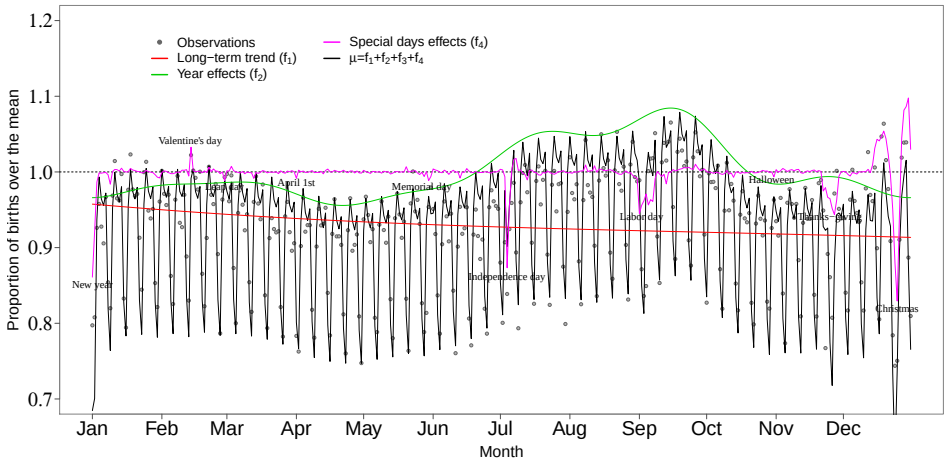


Figure 3.16: Posterior means of the function $\mu(\cdot)$ for the year 1972 of the series. The special days effects pattern ($f_4(\cdot)$) in the year is also represented, as well as the long-term trend ($f_1(\cdot)$) and year effects pattern ($f_2(\cdot)$).

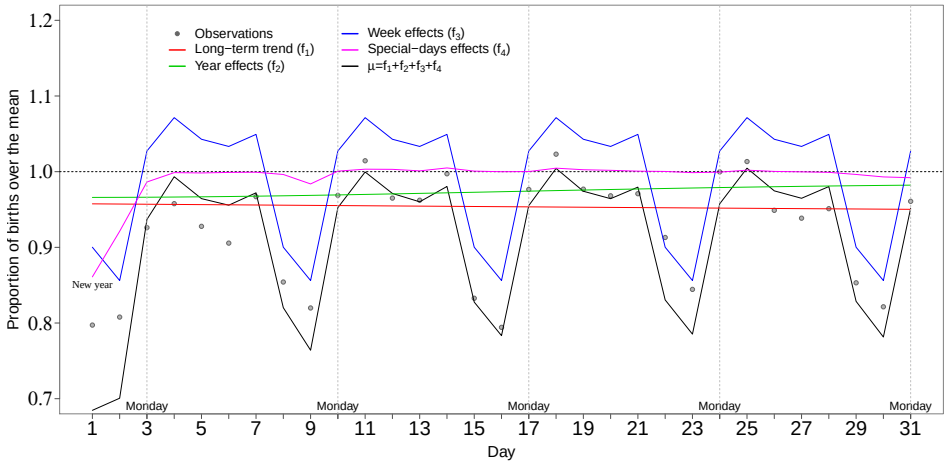


Figure 3.17: Posterior means of the function $\mu(\cdot)$ for the month of January of 1972. The week effects pattern ($f_3(\cdot)$) in the month is also represented, as well as the long-term trend ($f_1(\cdot)$), year effects pattern ($f_2(\cdot)$) and special days effects pattern ($f_4(\cdot)$).

3.10 Case study III: Diabetes data

The next example presents an epidemiological study of diabetes disease. The study aims to relate the probability of suffering from diabetes to some risk factors. The data contains $n = 392$ individuals ($i = 1, \ldots, n$) from which the binary variable of suffering ($y_i = 1$) or not suffering ($y_i = 0$) from diabetes have been observed. The matrix $X = [\boldsymbol{x}_1 \ \boldsymbol{x}_2 \ \cdots \ \boldsymbol{x}_n]^\top \in \mathbb{R}^{n\times 4}$, with $\boldsymbol{x}_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}) \in \mathbb{R}^4$, contains the risk factors, *Glucose* (x_{i1}), *Pregnancy* (x_{i2}), *Age* (x_{i3}) and *BMI* (x_{i4}), per individual i. The observational model is a Bernoulli model with parameter the probability p_i of suffering from diabetes per observation i,

$$y_i \sim \text{Bernoulli}(p_i).$$

The goal is to estimate the probability p_i as a function of the risk factors, which function $f(\cdot) : \mathbb{R}^4 \to \mathbb{R}$ is modeled as a Gaussian process with a multivariate squared exponential covariance function k depending on the matrix X of risk factors and hyperparameters $\theta = \{\alpha, \ell\}$, and related to the probabilities p_i through the *logit* link function,

$$\begin{aligned} p_i &= \text{logit}(f(\boldsymbol{x}_i)) \\ f(\boldsymbol{x}) &\sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}, \theta). \end{aligned}$$

Saying that the function $f(\cdot)$ follows a GP model is equivalent to say that $\boldsymbol{f} = \{f(\boldsymbol{x}_i)\}_{i=1}^n$ are multivariate Gaussian distributed with covariance matrix K, where $K_{ij} = k(x_i, x_j, \theta)$, with $i, j = 1, \ldots, n$. The hyperparameters α and ℓ represent the marginal variance and lengthscale, respectively, of the GP process. Notice that a scalar lengthscale is considered in the multivariate covariance function.

In the HSGP model with D input dimensions, the function $f(\boldsymbol{x})$ evaluated at input vector $\boldsymbol{x} \in \mathbb{R}^D$ is approximated as in equation (3.13), with the D-dimensional (with a scalar lengthscale) squared exponential spectral density S as in equation (3.3) and the D-vector of eigenvalues $\boldsymbol{\lambda}_j$ and the multivariate eigenfunctions ϕ_j as in equations (3.11) and (3.12), respectively.

In order to do model comparison, in addition to the regular GP and HSGP models, a D-dimensional splines-based model is also fitted using a cubic spline basis penalized by the conventional integrated square second derivative cubic spline penalty [Wood, 2017] and implemented in the R-package *mgcv*. A Bayesian approach is used to fit this spline model using the R-package *brms*.

Figure 3.18 shows the mean posterior predictions of probabilities (p_i) of the three models, the regular GP, the HSGP and the splines, fitted over the dataset with the 2 input dimensions *Glucose* and *Pregnancy* ($D = 2$). The binary observations

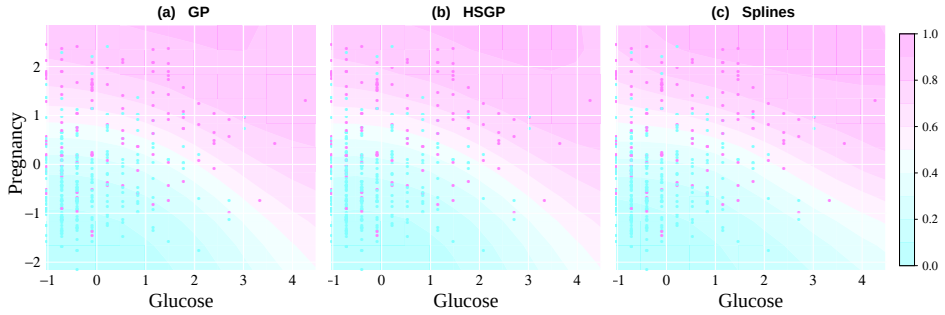


Figure 3.18: (a) Mean posterior predictive functions of the GP model. (b) Mean posterior predictive functions of the HSGP model. (c) Mean posterior predictive functions of the splines (SP) model. Samples observations of suffering (red points) and not suffering (blue points) form the disease are plotted.

y_i are also plotted in the plots as colored points. For the HSGP model, $m_1 = 20$ and $m_2 = 20$ basis functions for each dimension, respectively, were used, which lead to a total of 400 multivariate basis functions. A boundary factor for each dimension $c_1 = 4$ and $c_2 = 4$ were used. For the splines model, 20 knots per dimension were used.

In order to assess the performance of the models as a function of the boundary factor, the number of basis functions and knots, different models with different number of basis functions and boundary factor for the HSGP model and different number of knots for the splines model have been fitted. In all models, the same boundary factor, number of basis functions and knots per dimension were used. Figure 3.19 shows the expected log predictive density (ELPD) (ELPD was defined in equation 2.3 of chapter 2) as a function of the boundary factor c and the number of univariate basis functions m, for the HSGP model, and knots, for the splines model. The ELPD is computed over the actual observations by cross-validation. Basically, with slightly differences, all models show similar performances, due to the fact that the process is very smooth with a relatively very large lengthscale estimate $\ell = 4.51$. Even though, a slight pattern of performance improvement can be appreciated as the boundary factor c increases, which fact is because small boundary factors are not allowed when large lengthscales (Figure 3.6).

Figure 3.20 shows the computational times of the different models, regular GP, HSGP and splines, fitted over the dataset, with 2 input dimensions, 3 input dimensions and 4 input dimensions, as a function of the boundary factor c and number of univariate basis functions m and knots. We can appreciate the significant increase of computational time with higher dimensions for the HSGP and splines models.

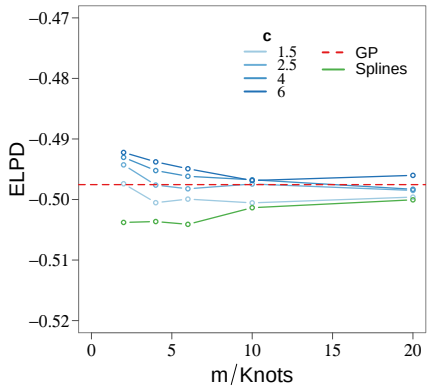


Figure 3.19: Expected log predictive density (ELPD) of the different methods as a function of the boundary factor c and the number of basis functions m, for the HSGP model, and knots, for the splines model.

This fact reveals that choosing optimal values for the number of basis functions and boundary factor, looking at the recommendations and diagnosis provided by Figure 3.6, is essential to avoid a excessive computational time, especially in high input dimensionality. It is interesting to be noticed that considering more than 10 knots per dimension in the splines model with 3D is not allowed for an amount of 392 observations. Similarly, just the computation of the input data for the splines model in a 4D input space is computationally very expensive.

The Stan model codes for the exact GP, the approximate GP and the splines models of this case study can be found in:

`https://github.com/gabriuma/Doctoral_thesis/tree/master/Case-study-III_Diabetes-data`

3.11 Case study IV: Spatio-temporal land-use classification

The next example presents an spatio-temporal classification in land-use of plots between 2006 and 2015 in a part of the territory of Valencia in Spain dedicated to growing citrus fruits. A sampling set consists of $N = 200$ plots with known class. The data is recorded in a time series of $T = 5$ years (2006, 2008, 2010, 2012, 2015) within the period. The class of each parcel i and time t is stored by a categorical variable y_{it} representing the $K = 5$ different possible classes ($k = 1, \ldots, K$):

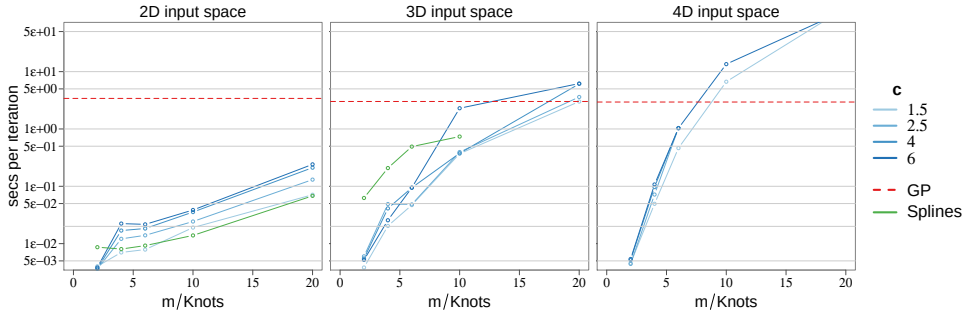


Figure 3.20: Time of computation in seconds per iteration (iteration of the HMC sampling method) of the different models fitted over the dataset, with 2 input dimensions (left) and 3 input dimensions (center) and 4 input dimensions (right), as a function of the boundary factor c and number of basis functions m, for the HSGP model, and knots, for the splines model.

$k = 1$, adult independent citrus fruits; $k = 2$, aligned citrus fruits; $k = 3$, irregular citrus fruits; $k = 4$, abandoned citrus fruits; $k = 5$, young citrus fruits.

A bunch of 52 characteristic variables was available for every parcel and time. These variables were computed from satellite color images and cadastral map by using the software FETEX for automatic descriptive feature extraction from image-objects [Ruiz et al., 2011]. These variables concern spectral intensities and empirical semivariogram of the pixels within a plot, as well as descriptive statistics of the shape of the plots.

Due to the fact that 52 input variables are too high-dimensional for a multivariate HSGP model, which computational cost scales as $O(nm^D + m^D)$, with m the number of basis functions and D the number of input variables, the multivariate HSGP model will be formulated as an additive HSGP model.

As it is known, the computational demand of a multivariate HSGP model component increases quickly with the number of dimensions, so we should avoid high-dimensional HSGP components in the additive model. Original input variables are highly correlated, which would imply the use of high-order interaction components in the additive model to achieve accurate model performance. Therefore, instead of using the original variables as inputs, we use their principal components, which are expect to be linearly uncorrelated. Using the principal components as inputs helps, in principle, not to have to use as many high-order interaction components in the additive model.

The principal components (PCs) will be used jointly with the time variable as inputs in the classifying additive HSGP model. Let's denote the matrix

$X = [\boldsymbol{x}_{11} \cdots \boldsymbol{x}_{it} \cdots \boldsymbol{x}_{NT}]^\top \in \mathbb{R}^{NT \times D}$, which contains the input vectors $\boldsymbol{x}_{it} \in \mathbb{R}^D$, $D = 53$ (52 PCs plus time) for the spatio-temporal observations (plots $i = 1, \ldots, 200$, and times $t = 1, \ldots, 5$).

The observational model is a multinomial model with parameters the vector of probabilities $\boldsymbol{p}_{it} = (p_{it,1}, \ldots, p_{it,K})$, where $p_{it,k}$ is the probability of belonging to class k per parcel i and time t,

$$y_{it} \sim \text{multinomial}(\boldsymbol{p}_{it}).$$

The goal is to estimate the vector of probabilities $\boldsymbol{p}_{it}$ as a function of the predictors, which is a multivariate function $f(\boldsymbol{x}_{it}) : \mathbb{R}^D \to \mathbb{R}^K$,

$$f(\boldsymbol{x}_{it}) = \big(f_1(\boldsymbol{x}_{it}), \ldots, f_K(\boldsymbol{x}_{it})\big),$$

which is related to the vector of probabilities $\boldsymbol{p}_{it}$ through the 'softmax' link function (equation 2.1 in chapter 2):

$$\boldsymbol{p}_{it} = \text{softmax}(f(\boldsymbol{x}_{it})).$$

Each individual function $f_k(\boldsymbol{x})$ is modeled as a first-order additive model plus the second-order additive effects between time input variable (x^D) and all the other inputs as follows:

$$f_k(\boldsymbol{x}) = \sum_{d=1}^{D} g_d(x^d) + \sum_{d=1}^{D-1} h_{d,D}(x^d, x^D). \tag{3.24}$$

The first-order components $\{g_d(x^d)\}_{d=1}^D$ in equation (3.24) are modeled as unidimensional HSGP models:

$$g_d(x^d) \sim \mathcal{HSGP}(x^d, S, \theta_{d,1}).$$

In the HSGP model, a first-order components $g_d(x^d)$, evaluated at input value $x^d \in \mathbb{R}$, is approximated as in equation (3.9) with the squared exponential spectral density S as in equation (3.3) and eigenvalues λ_j and eigenfunctions ϕ_j as in Equations (3.7) and (3.8), respectively.

The second-order components $\{h_{d,D}(x^d, x^D)\}_{d=1}^{D-1}$ in Equation (3.24) are modeled as two-dimensional HSGP models:

$$h_{d,D}(x^d, x^D) \sim \mathcal{HSGP}(x^d, x^D, S, \theta_{d,D}).$$

In the HSGP model, a second-order component $h_{d,D}(x^d, x^D)$, evaluated at in-

puts $x^d \in \mathbb{R}$ and $x^D \in \mathbb{R}$, is approximated as in equation (3.13) with the two-dimensional (with a scalar lengthscale) squared exponential spectral density S as in equation (3.3) and the D-vector of eigenvalues $\boldsymbol{\lambda}_j$ and the multivariate eigenfunctions ϕ_j as in equations (3.11) and (3.12), respectively.

The vector of hyperparameters $\theta_{d,1} = (\alpha_{d,1}, \ell_{d,1})$ contains the marginal variance $\alpha_{d,1}$ and lengthscale $\ell_{d,1}$ of the $g_d(x^d)$ model component. And, the vector of hyperparameters $\theta_{d,D} = (\alpha_{d,D}, \ell_{d,D})$ contains the marginal variance $\alpha_{d,D}$ and lengthscale $\ell_{d,D}$ of the $h_{d,D}(x^d, x^D)$ model component.

For the first-order components $g_d(x^d)$, $m = 15$ basis functions and a boundary factor $c = 2.5$ were used. For the second-order components $h_{d,D}(x^d, x^D)$, $m_1 = 15$ and $m_2 = 15$ basis functions for each dimension, respectively, were used, which lead to a total of 225 multivariate basis functions. A boundary factor for each dimension $c_1 = 2.5$ and $c_2 = 2.5$ were used. All the input variables were previously standardized.

In the case of the first-order components, the normalized lengthscale estimates $\left(\frac{2 \cdot \hat{\ell}_{d,1}}{|x^d_{max} - x^d_{min}|}\right)$ are all bigger than the minimum lengthscale reported by Figure 3.6 as a function of m, c. Which means that the used number of basis functions ($m = 15$) and boundary factor ($c = 2.5$) are suitable values for modeling accurately the input effects.

For the second-order components, the relationships between the number of basis functions, the boundary factor and the lengthscale is not available for the multivariate case. However, we can approximately analyze the lengthscale estimates of the second-order HSGP components analyzing each dimension separately as unidimensional HSGP models.

Table 3.1 shows the confusion matrix after fitting the model following a Q-fold cross-validation procedure, with $Q = 100$, over the training data. Thus, every fold contains 10 observations. The confusion matrix evaluates the rate of misclassification per class. Columns represent the true classes and rows represent the estimated classes. The values within the matrix correspond to the number of items that fall into every cell. The marginals of the columns (true classes) represent the percentage of misclassified items in relation to the 'truth', commonly known as the *omission error*. And the marginals of the rows (estimated classes) represent the percentage of misclassified items in relation to the estimates (classifier), commonly known as the *commission error*. The percentage in the down right cell of the matrix is the overall mean misclassification rate. As can be seen, there exist a high misclassification rate between classes $k = 1$ and $k = 2$, between classes $k = 1$ and $k = 3$, and between classes $k = 1$ and $k = 5$.

Estimate \ True	k = 1	k = 2	k = 3	k = 4	k = 5	
k = 1	90	39	14	3	11	42%
k = 2	46	301	8	2	3	16%
k = 3	8	4	59	4	1	19%
k = 4	5	2	6	342	5	5%
k = 5	8	2	1	0	38	19%
	42%	13%	32%	2%	34%	**17%**

Table 3.1: Confusion matrix after the Q-fold cross-validation procedure over the training data.

Further modeling considerations

In this case study, the temporal correlation structure has been taken into account jointly with other spatial covariates in a multivariate squared exponential covariance function. That implies that the covariance structure in the temporal dimension is modeled as a smooth and monotonically decreasing function in time. The grade of decay of the covariance in function of time depends on the relations observed in the data.

However, this prior assumption on the temporal covariance structure might be too simple for this practical case. In this application case, the class of an observation is expected to switch in time in different ways. Thus, the assumption of a smooth and monotonic covariance function in time might not be appropriate or at least too simple. In this sense, Figure 3.21 provides the first-order transition probabilities matrix between classes of the sampling dataset. A first-order transition probability means the probability of a specific class to change to another class when the time variable changes one unit, that is, the probability of change of a class in one interval of time. It can be seen that there is a significant probability of some of the classes to change in time. For example, class $k = 1$ has 0.64 of probability to change to class $k = 4$ when time changes from one time point to the next time point, or class $k = 5$ has 0.65 of probability to change to class $k = 2$ in one time interval. This suggests that a Markov chain distribution for the time dimension could be more appropriate. Thus, in further research, we propose modeling this practical case by specifying a Markov chain model for the transition probabilities of classes in time and a multivariate Gaussian process prior, with the spatial predictors, to relate the transition probabilities among plots (space).

The Stan model code for the approximate GP model of this case study can be found in:

Figure 3.21: Probabilities transition matrix between classes

`https://github.com/gabriuma/Doctoral_thesis/tree/master/Case-study-IV_Land-use-classification`

3.12 Conclusion

The GP model entails a complexity that is computationally intractable for many practical problems, and this problem especially becomes severe when we want to perform inference using sampling methods. In this Chapter, we have implemented and analyzed a novel approach for a low-rank representation of GPs, originally and theoretically developed by Solin and Särkkä [2018]. The method is based on a basis function approximation via Laplace eigenfunctions. The method has an attractive computational cost as this basically turns the regular GP model into a linear model, which is also an attractive property in modular probabilistic programming models.

The dominating cost per log density evaluation (during sampling) is $O(nm+m)$, which is a big benefit in comparison to $O(n^3)$ of a regular GP model. The design matrix is independent of hyperparameters and therefore only needs to be constructed once, at cost $O(nm)$. All dependencies on the kernel and the hyperparameters are through the prior distribution of the regression weights. The parameter posterior distribution is m-dimensional, where m is much smaller than the number of observations n, which is greatly reduced in comparison to regular GPs and this makes inference faster, especially when sampling methods are used. The drawbacks of the method are the boundary conditions and scaling with respect to the number

of dimensions, i.e. the number of basis functions m scales exponentially with the number of dimensions.

In this Chapter, we clearly presented the formulae of the method. We have shown how the method applies for all the stationary covariance functions as long as they can be represented in terms of their spectral densities. We focused on the analysis of the accuracy of the approximation in relation to the key factors of the method, the number of basis functions, the boundary condition of the Laplace eigenfunctions and the non-linearity of the function to be learned. .

Basically, if the lengthscale estimate characterizes accurately the non-linearity of the function to be learned, we can expect the HSGP approximation to be accurate as well. Thus, the performance of the method relies ultimately on the relationship among the number of basis functions, the boundary factor and the lengthscale of the function, which depends on the particular choice of the covariance function. We made recommendations for the values of these key factors based on the recognized relations among them. We provided useful graphs of these relations that will help the users to improve performance and save time of computation. We also diagnose if the chosen values for the number of basis functions and the box size are adequate to fit to the actual data.

In this work, we built this relationship for the squared exponential covariance function and Matern (ν=3/2) covariance functions. For the particular case of a periodic covariance function we have presented a related methodology based on a low-rank representation of a periodic kernel following the work of Solin and Särkkä [2014], and then we analyzed the performance of the approximation in relation to the number of basis functions.

Furthermore, we put the focus on showing how GPs can be easily used as modular components in probabilistic programming frameworks (i.e. Stan, WinBUGS and others) and can be used as latent functions in non-Gaussian observational models. We have shown several illustrative examples, simulated and real datasets, of the performance of the model, where we demonstrated the applicability and the implementation of the methodology, the reduction of the computation and the improvement in sampling efficiency. The Stan codes of these case studies have been provided through links to the author's GitHub repository.

The main drawback of this approach is that its computational complexity scales exponentially with the number of dimensions. Hence, in practice, input dimensionalities larger than 3 start to be too computationally demanding. In these cases, the proposed HSGP model can be used as components in an additive modeling scheme. In the spatio-temporal land use classification task performed in the case study IV (Section 3.11), we model the 52 input dimensions additively using 1D and 2D HSGP model components.

Future research will focus on constructing analytical models for the relationships

between the key factors of the number of basis functions, the boundary factor and the lengthscale of the function, depicted in Figures 3.6, 3.7 and 3.11, on which ultimately depend the performance of the approximation. This analytical models can be useful to automatize the diagnosis of the performance of the approximation. These relationships have been obtained, in the present study, for the unidimensional case only. So, another future research line will be focused on analyzing these relationships in the multidimensional case, building useful graphs or analytical models that encode these relationships in multidimensional cases.

Chapter 4

Additional (virtual) derivative observations to induce monotonicity and gradient to functions

In this chapter, we illustrate the use of derivative information as additional or virtual observations in the modeling, in order to constraint or control the dynamics of stochastic functions. The methodology is illustrated for GPs and semiparametric models based on splines functions. The aim is to ultimately reveal and make recommendations about the issues that can arise when using this methodology to induce monotonicity on functions, especially when using GP models. Based on prior knowledge of the functions, controlling or constraining the dynamics of the functions is especially useful when the data is scarce relative to the complexity of the model or it is very noisy, and also produces better model extrapolations. This is the case of the application to rock-art paintings addressed in Chapter 5.

4.1 Introduction

In modeling problems of learning stochastic functions from data, there is often a priori knowledge and/or additional information available concerning the function to be learned, which can be used to improve the performance of the modeling. This information might concern the derivative of the function with respect to an input variable, so the dynamics of the function can be controlled or constrained. Some-

times, prior knowledge or measurements of the value of the derivative (gradient) of the function may be available. And sometimes, such information can be regarded on the behavior of the function such as being monotonically increasing or decreasing with respect to some input variables.

Knowing or having measurements of the value of the derivative of the function implies that specific behaviour regarded to the gradient of the function with respect to an input variable can be incorporated to the modeling. In general, different orders of derivatives can be considered. This information can be available in many applications. For example, in industrial applications where the gradient of the process under study can be measured at specific input values, or in the case of modeling the degradation of some substance which degradation is expected to stabilize eventually.

The monotonicity assumption in general means that a function always increases (monotonically increasing) or decreases (monotonically decreasing) as a function of some input variable. There are several ways a stochastic function can be monotonic, e.g. the expected values of a function are monotonic or all samples from a stochastic function are monotonic. Monotonic functions arise in many applications, such as modeling the mortality rate as a function of the age, or modeling the treatment effects as a function of drug dose response, or modeling the degradation of a substance as a function of time and influence of adverse factors. Monotonicity can also be a useful prior information to gain efficiency in algorithms of function optimization [Li et al., 2017].

Modeling this information, gradients or monotonicity, has several advantages. Models with additional derivative information have stronger inductive bias, thus yielding better predictive performance and confidence intervals, especially for smaller data sets. Specifically, they can be expected to produce more reliable models, especially where the data is scarse, and produce better model extrapolations. Furthermore, additional derivative information can improve model interpretation.

In this work we use the term virtual observations instead of additional observations, in order to make it clear that they are artificial observations we introduce to induce specific information into the model.

The use of additional or virtual observations of the partial derivative to induce monotonicity on functions [Rasmussen, 2003, Riihimäki and Vehtari, 2010, Solak et al., 2003] has some interesting properties, but also has some issues. The use of virtual observations allows for imposing different monotonicity in different input dimensions. Furthermore, by placing virtual points appropriately, unimodality can be imposed with respect to an input variable [Andersen et al., 2017]. The use of virtual observations of the partial derivative also allows for imposing constrains concerning the gradient of the functions. Generally, derivative observations (of different orders) can be used to constrain the dynamics of the functions in different

ways. However, inducing monotonicity through virtual observations can arise with some practical issues, since the monotonicity information is included in the likelihood of the model through virtual observations instead of into the prior of the function. Which makes the posterior distribution of the function dependent on the number of the inducing points. Furthermore, this approach does not guarantee stochastic functions are either sampling-wise monotonic or even monotonic in expectation.

In this chapter, we illustrate the usage of derivative information as additional or virtual observations in the modeling, and also reveal the issues that can arise when using this methodology to induce monotonicity on functions.

The rest of the chapter is structured as follows. In Section 4.2, a brief overview of the related work is given. In Section 4.3, we describe the main contributions of this chapter. In Section 4.4 the general observational model is set. In Sections 4.5.1 and 4.5.2, the derivative processes of both Gaussian processes (GPs) and splines-based models, respectively, are derived and jointly modeled with their regular processes. Section 4.6 describes the implementation of zero-order constraints which concern to the function values of the regular process. Section 4.7.1 describes the implementation of first-order derivative constraints regarded the values of the derivative (gradients) of the function. In Section 4.7.2, first-order derivative constraints regarded the the sign of the derivatives for monotonicity are described. Section 4.9 illustrates the main issue of using additional or virtual observations for monotonicity. Finally, in Section 4.10 brief conclusions are given.

4.2 State of the art

Several methods have been proposed for monotonic regression in the literature. The predominant focus of the theoretical literature on monotone function estimation has been on the methodology of order-restricted inference, which is sometimes also known as isotonic regression [Barlow et al., 1972]. Neelon and Dunson [2004] approach isotonic regression and order-restricted inference for non-parametric models in a Bayesian analysis. Brezger and Steiner [2008] induces monotonicity on penalized B-splines imposing order- restriction by specifying truncated prior distributions in order to reject the undesired draws for the parameters in the MCMC sampling. Reich et al. [2011] makes a similar approach imposing order to the regression parameters by means of reparameterizing and constraining these parameters with application to a quantile regression model. Shively et al. [2009] proposes two approaches to obtain monotonic functions, the first using a modified characterization of the smooth monotone functions proposed in Ramsay [1998] that allows for unconstrained estimation, and the second using constrained prior distributions for

the regression coefficients to ensure monotonicity.

However, imposing monotonicity by construction in non-parametric GP prior models is more difficult. Generic GP prior models do not restrict the function values to be monotonically increasing or decreasing with respect to input variables. Few approaches to induce monotonicity in GP models can be found. Recently, Andersen et al. [2018] proposes a monotonic model based on applying non-linear transformation of a GP and then using Hilbert space methods to approximate the model to make inference tractable.

In addition to monotonic regression imposed by construction, monotonicity can be expressed in terms of the sign of the partial derivative of the functions [Riihimäki and Vehtari, 2010, Solak et al., 2003]. Furthermore, gradients of the functions can also be expressed in terms of the value of the partial derivatives. In order to do that the derivative process of the model has to be derived and jointly modeled with the regular process. Thus, jointly modeling observations of the partial derivative and regular observations.

Linearity of parametric or semiparametric models makes feasible the use of their derivatives as additional observations jointly with the regular observations [Rasmussen, 2003]. In the same way, the derivative of a (mean-square differentiable) GP function remains a GP because differentiation is a linear operator [Solak et al., 2003]. This makes it possible to use derivative observations jointly with regular observations in a GP model, by extending the covariance function accordingly to include the covariances between the process and its partial derivatives [Solak et al., 2003]. Riihimäki and Vehtari [2010] proposed a method for inducing monotonicity in functions using GPs by using additional observations of the sign of the derivative of the process in specific locations in the input space. A similar approach of using additional observations for inducing monotonicity to neural network models was proposed by Sill [1998]. In Lorenzi and Filippone [2018] the authors use the same idea to include constraints concerning the derivative of the functions in deep probabilistic models [Lorenzi and Filippone, 2018].

The use of additional (virtual) observations in multi-dimensional input spaces has the drawback that the number of observations can considerably increase and the computation becomes very heavy. In this case, approximate inference methods must be used [Gelman et al., 2013] instead of sampling methods such as MCMC (Markov chain Monte Carlo). Another option is to settle for a small number of virtual observations, placed appropriately, as long as the forced function is monotonic. Riihimäki and Vehtari [2010] propose a method for placing the virtual points based on the probability of function being negative at the input points, and thus optimize the computational complexity.

4.3 Contributions of the chapter

In this chapter, we illustrate the use of derivative information as additional (virtual) observations in the modeling. We illustrate this in two different modeling frameworks: a GP model [Riihimäki and Vehtari, 2010] and a semiparametric splines-based model. The derivative process of the model is derived and then both the regular and derivative processes are jointly modeled. In this way, we can encode derivative information in the model by defining observational models for the derivative observations: observations of the value of the derivative (gradient) and observations of the sign of the derivative for monotonicity.

Finally, we analyze the issue that arises when using virtual derivative sign observations to induce monotonicity on functions. The problem is that the posterior functions depend on the number and locations of the virtual observations in the input space. When the number of virtual points is too large, the posteriors tend to be overly smoothed. However, if the function is smooth, this problem can be avoided in practice choosing a few virtual points only and placing them appropriately.

4.4 General observational model

We consider a continuous stochastic process based on a collection of observations $\boldsymbol{y} = (y_1, \ldots, y_N)$ of the regular process, defined at the subset $A = \{i : i = 1, \ldots, N\}$ of observational indices, where $y_i \in \mathbb{R}$ is the value of the observation $i \in A$. And an associated matrix of inputs $X_A = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ \cdots\ \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times D}$, where $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id}, \ldots, x_{iD}) \in \mathbb{R}^D$ denotes the vector of input values for the ith observation, with $d = 1, \ldots, D$ denoting the indices for the inputs variables.

We adopt the model

$$y_i = f_i + e_i,$$

where f_i is the value of the latent function $f : \mathbb{R}^{\mathrm{D}} \to \mathbb{R}$ underlying to the observations and evaluated at $\boldsymbol{x}_i$, i.e. $f_i = f(\boldsymbol{x}_i)$, and e_i is a Gaussian noise term. So, the observational model for the data can be written as follows:

$$p(\boldsymbol{y}|\boldsymbol{f}) = \prod_{i \in A} \mathcal{N}(y_i|f_i, \sigma^2), \tag{4.1}$$

where $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N))$ and σ^2 is the noise variance.

4.5 Latent function models with derivatives

4.5.1 Gaussian processes with derivatives

In this section we derive the first-order partial derivative process of a multivariate GP prior model, and jointly model both the regular and the derivative processes of a GP. GPs are prior models for multi-dimensional functions (Chapter 3.4).

The function f in equation (4.1) is assumed to follow a zero mean GP prior, $f \sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}'))$, where $k : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ is the covariance function. Let $\boldsymbol{f} = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N))$, then the resulting prior distribution for $\boldsymbol{f}$ is a multivariate Gaussian distribution $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$, where $\boldsymbol{K}$ is the covariance matrix, with $K_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, with $i, j = 1, \ldots, N$.

We consider a stationary and separable exponentiated quadratic covariance function which depends on a set of hyperparameters θ. Thus, the element (i, j) of the covariance matrix K is:

$$K(X;\theta)_{ij} = \alpha^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{1}{\rho_d^2}(x_{i,d} - x_{j,d})^2\right),$$

where θ contains the parameters α and $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_D)$. The hyperparameter α is the prior standard deviation of the latent Gaussian process $\boldsymbol{f}$. The lengthscale hyperparameter ρ_d control the smoothness of the covariance function or rate of decay of the correlation in the direction of the dth predictor (input variable), so that, the larger the ρ_d the smoother the correlation function and the smoother the posterior functions for $\boldsymbol{f}$; Although the scale of ρ_d is dependent on the scale of the input data along the dimension d. The magnitude α and lengthscale ρ_d must be strictly positive parameters.

As differentiation is a linear operator the derivative of a mean-square differentiable GP model is still a GP. This makes it possible to jointly model a function and its derivatives using GPs, by extending the covariance function accordingly to include the covariances between the process and its partial derivatives. Hence, we can write the joint prior distribution for regular function values $\boldsymbol{f}$ and partial derivatives of function values $\boldsymbol{f}'$ as follows

$$p(\boldsymbol{f}, \boldsymbol{f}'|X, X*, \theta) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}' \end{bmatrix} \middle| 0, \begin{bmatrix} K_{f,f}(X,\theta) & K_{f,f'}(X, X^*, \theta) \\ K_{f',f}(X, X^*, \theta) & K_{f',f'}(X^*, \theta) \end{bmatrix}\right), \quad (4.2)$$

where $\boldsymbol{f}'$ denotes the values of the partial derivatives of latent function f with respect to some input dimension evaluated at inputs X^* associated with the derivative

observations. The covariance matrix is extended to include the covariances between observations and partial derivatives ($K_{f,f'}$ and $K_{f',f}$) and the covariances between partial derivatives ($K_{f',f'}$). The covariance between a partial derivative and a function value is given by

$$\operatorname{Cov}\left[\frac{\partial f^i}{\partial x_{i,g}}, f^j\right] = \alpha^2 \exp\Big(-\frac{1}{2}\sum_{d=1}^{D}\rho_d^{-2}(x_{i,d}-x_{j,d})^2\Big) \times \big(-\rho_g^{-2}(x_{i,g}-x_{j,g})\big),$$

and the covariance between partial derivatives

$$\operatorname{Cov}\left[\frac{\partial f^i}{\partial x_{i,g}}, \frac{\partial f^j}{\partial x_{j,h}}\right] = \alpha^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\rho_d^{-2}(x_{i,d}-x_{j,d})^2\right) \times \rho_g^{-2}\left(\zeta_{gh}-\rho_h^{-2}(x_{i,h}-x_{j,h})(x_{i,g}-x_{j,g})\right).$$

In the previous equation, ζ denotes the Kronecker Delta function where $\zeta_{gh}=1$ if $g=h$, and 0 otherwise [Riihimäki and Vehtari, 2010].

We can now combine the joint prior distribution in equation (4.2) with an observation model, $p(\boldsymbol{m}|\boldsymbol{f}')$, for the partial derivatives observations $\boldsymbol{m}$, to encode information about the partial derivatives of f into the model.

4.5.2 Splines model with derivatives

In this section, we derive the first-order derivative process of a penalized cubic-splines model in the one-dimensional input space. We focus on semiparametric regression models using penalized thin-plate splines [Ruppert et al., 2003]. Specifically, we focus on the penalized thin-plate splines as presented by Crainiceanu et al. [2005].

Due to the number of parameters in multivariate splines models is high, in the sake of simplicity, we consider the process in the one-dimensional case only, such that the latent function f in equation (4.1) is a 1D-function, $f:\mathbb{R}\to\mathbb{R}$.

Let $\boldsymbol{f}=(f(\boldsymbol{x}_1),\ldots,f(\boldsymbol{x}_N))$ be the latent function values underlying the noisy observations $\boldsymbol{y}$, where f_i is the value of the latent function f evaluated at the input value x_i, i.e. $f_i=f(x_i)$. Firstly, we assume the latent function f is a thin-plate cubic-splines function:

$$f(x_i)=\beta_1+\beta_2 x_i+\sum_{k=1}^{K}Z_{ik}u_k=\boldsymbol{H}_i.\boldsymbol{\beta}+\boldsymbol{Z}_i.\boldsymbol{u}, \tag{4.3}$$

where $\boldsymbol{H}_i. = (1, x_i)$ is the row-vector of linear function values of the ith observations, $\boldsymbol{\beta} = (\beta_1, \beta_2)^\top$ is the column-vector of linear coefficients, $\boldsymbol{Z}_i. = (Z_{i1}, \ldots, Z_{iK}) \in \mathbb{R}^K$ is the row-vector of cubic-splines values of the ith observation, and $\boldsymbol{u} = (u_1, \ldots, u_K)^\top \in \mathbb{R}^K$ is the column-vector of splines coefficients, with K the order of the splines model and number of knots. The kth element of the vector $\boldsymbol{Z}_i.$ of cubic-splines function values is

$$\begin{aligned} Z_{ik} = \phi(x_i, \kappa_k) = \\ \frac{1}{4}\left((\kappa_k - 0.5)^2 - \frac{1}{12}\right)\left((x_i - 0.5)^2 - \frac{1}{12}\right) \\ -\frac{1}{24}\left(\left(|x_i - \kappa_k| - 0.5\right)^4 - \frac{1}{2}\left(|x_i - \kappa_k| - 0.5\right)^2\right), \end{aligned} \tag{4.4}$$

which is the one-dimensional cubic-splines function ϕ [Nievergelt, 1993] evaluated at the input value x_i and the pre-fixed knot κ_k corresponding to the kth spline function, with $k = 1, ..., K$.

Following, we briefly derive the penalized representation of the thin-plate splines of equation (4.3). A more detailed explanation of this approach can be found in Crainiceanu et al. [2005]. In order to derive the penalized representation, let first formulate the complete latent model for the observations $\boldsymbol{y}$:

$$\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\epsilon} = H\boldsymbol{\beta} + Z\boldsymbol{u} + \boldsymbol{\epsilon},$$

where $\boldsymbol{y}$ and $\boldsymbol{f}$ denote the column-vectors of observations and latent function values, respectievely. $H \in \mathbb{R}^{N\times 2}$ denotes the matrix of linear function values with ith row $\boldsymbol{H}_i.$, $Z \in \mathbb{R}^{N\times K}$ denotes the matrix of cubic-spline function values with ith row $\boldsymbol{Z}_i.$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2)$ is the Gaussian noise term. To perform the penalization of the model in order to avoid overfitting, we can minimize:

$$\frac{1}{\sigma^2}||\boldsymbol{y} - H\boldsymbol{\beta} + Z\boldsymbol{u}||^2 + \frac{1}{\sigma_u^2}\boldsymbol{u}^\top \Sigma \boldsymbol{u},$$

in which the covariance of the vector $\boldsymbol{u}$ is $\mathrm{cov}(\boldsymbol{u}) = \sigma_u^2\Sigma^{-1}$, the covariance of the residuals $\boldsymbol{\epsilon}$ is linear Gaussian $\mathrm{cov}(\boldsymbol{\epsilon}) = \sigma^2 I$, and $\Sigma \in \mathbb{R}^{K\times K}$ is the matrix of

penalization with element (q, k):

$$\begin{aligned}\Sigma_{qk} = \phi(\kappa_q, \kappa_k) = \\ \frac{1}{4}\left((\kappa_k - 0.5)^2 - \frac{1}{12}\right)\left((\kappa_q - 0.5)^2 - \frac{1}{12}\right) \\ -\frac{1}{24}\left(\left(|\kappa_q - \kappa_k| - 0.5\right)^4 - \frac{1}{2}\left(|\kappa_q - \kappa_k| - 0.5\right)^2\right),\end{aligned}$$

that is the cubic-splines function ϕ evaluated at the pre-fixed knots κ_q and κ_k, corresponding to the qth and kth spline function, respectively, with $q, k = 1, ..., K$.

If we use the reparametrization $\boldsymbol{u} = \Sigma^{-1/2}\boldsymbol{b}$, we obtain an equivalent model representation of the penalized thin-plate splines in the form of linear mixed models [Brumback et al., 1999]:

$$\boldsymbol{y} = \boldsymbol{f} + \boldsymbol{\epsilon} = H\boldsymbol{\beta} + Z\Sigma^{-1/2}\boldsymbol{b} + \boldsymbol{\epsilon},$$

where the covariance of the vector splines coefficients $\boldsymbol{b}$ is now linear Gaussian $\mathrm{cov}(\boldsymbol{b}) = \sigma_b^2 I$.

Thus, we represent the latent function f as penalized thin-plate cubic-splines in the form of linear mixed models:

$$f(x_i) = \boldsymbol{H}_{i.}\boldsymbol{\beta} + \boldsymbol{Z}_{i.}\Sigma^{-1/2}\boldsymbol{b}, \tag{4.5}$$

where $\boldsymbol{b} = (b_1, \ldots, b_K)^\top \in \mathbb{R}^K$ is the column-vector of penalized splines coefficients.

In practice, the inverse of the square root of the matrix Σ is often computed as $\Sigma^{-1/2} = UD^{-1/2}V^\top$, where $UDV^\top$ is the singular value decomposition of the matrix Σ, where D is the rectangular diagonal matrix with the singular values, U and V are the matrices with the left-singular vectors and the right-singular vectors of Σ, respectively.

The partial derivative of the penalized cubic-splines function $f(x_i)$ in (4.5), with respect to the input variable takes the form

$$f'(x_i) = \frac{\partial f(x_i)}{\partial x_i} = \beta_2 + \frac{\partial \boldsymbol{Z}_{i.}}{\partial x_i}\Sigma^{-1/2}\boldsymbol{b}, \tag{4.6}$$

where $\frac{\partial \boldsymbol{Z}_{i.}}{\partial x_i} = (\frac{\partial Z_{i1}}{\partial x_i}, \ldots, \frac{\partial Z_{iK}}{\partial x_i}) \in \mathbb{R}^K$ is the row-vector of derivative cubic-splines values of the ith observation, where the element $\frac{\partial \boldsymbol{Z}_{ik}}{\partial x_i}$ is the partial derivative of the

cubic splines function Z_{ik} in equation (4.4):

$$\frac{\partial Z_{ik}}{\partial x_i} = \frac{\phi(x_i, \kappa_k)}{\partial x_i} = \frac{1}{4}\left((\kappa_k - 0.5)^2 - \frac{1}{12}\right) 2(x_i - 0.5) - \frac{1}{24}\left(4\big(|x_i - \kappa_k| - 0.5\big)^3 - \big(|x_i - \kappa_k| - 0.5\big)\right), \quad (4.7)$$

evaluated at the input value x_i and knot κ_k.

4.6 Function value constraint (zero-order constraint)

Before going through the implementation of constraints regarding the derivative of the function in the next sections, in this section we illustrate the implementation of constraints regarding the regular values of the function. We can consider sets of actual or virtual observations to be used as constraining observations, and these can be considered noisy or with the absence of noise.

Let $\boldsymbol{y}_B = (y_1, \ldots, y_J)$ be the set of constraining observations, with associated inputs $X_B = [\boldsymbol{x}_1 \ \boldsymbol{x}_2 \ \cdots \ \boldsymbol{x}_J]^\top \in \mathbb{R}^{J \times D}$ and defined at the set $B = \{i : i = 1, \ldots, J\}$ of observation indices.

If the constraining observations are considered noisy, we can incorporate this into the model by using a Gaussian likelihood with some variance $\tau^2 > 0$,

$$p(\boldsymbol{y}_B | \boldsymbol{f}_B) = \prod_{i \in B} \mathcal{N}(y_i | f(\boldsymbol{x}_i), \tau^2),$$

On the other hand, if the constraining observations are considered with the absence of noise, e.g. constrainting the function to pass through exactly these observations, the Dirac delta function δ as the observational model can be considered:

$$p(\boldsymbol{y}_B | \boldsymbol{f}_B) = \prod_{i \in B} \delta(y_i - f(\boldsymbol{x}_i)), \quad (4.8)$$

where $\boldsymbol{f}_B$ denotes the function values $f(\boldsymbol{x}_i)$ at the subset B of points.

In the case of the GP prior model for the latent function (Section 4.5.1), the function values $\boldsymbol{f}_B$ follow the marginalized distribution of equation (4.2), $p(\boldsymbol{f}_B | X_B, \theta) = \mathcal{N}\big(\boldsymbol{f}_B | K_{f_B, f_B}(X_B, \theta)\big)$. In the case of the splines model for the latent function (Section 4.5.2), the function f follows the model in equation (4.5).

Notice that generally, we can also consider the constraining observations as a subset of the actual observations $\boldsymbol{y}$, instead of considering them as a new set of (virtual) observations.

4.7 First-order derivative constraint

In this section, we illustrate the implementation of first-order derivative constraints to the function based on virtual observations of both the value of the partial derivative (gradient) and the sign of the partial derivative for monotonicity.

4.7.1 First-order derivative value (gradient) constraint

We consider a set $\boldsymbol{m} = (m_1, \ldots, m_L)$ of virtual observations of the partial derivative of the function, with associated inputs $X_C = [\boldsymbol{x}_1 \ \boldsymbol{x}_2 \ \cdots \ \boldsymbol{x}_L]^\top \in \mathbb{R}^{L \times D}$ and defined at the set $C = \{i : i = 1, \ldots, L\}$ of observation indices. Where m_i represents an artificial observation of $f'_g(\boldsymbol{x}_i) = \frac{\partial f(\boldsymbol{x}_i)}{\partial x_{i,g}}$, where $f'_g : \mathbb{R}^D \to \mathbb{R}$ is the partial derivative function with respect to the gth input variable of interest.

An observation model for these observations, $p(\boldsymbol{m}|\boldsymbol{f}'_C)$, can be used to encode this derivative information of f (provided by $\boldsymbol{m}$) into the model.

If the derivative observations $\boldsymbol{m}$ are noisy, we can incorporate this into the model by using a Gaussian likelihood with some variance $\tau^2 > 0$,

$$p(\boldsymbol{m}|\boldsymbol{f}'_C) = \prod_{i \in C} \mathcal{N}(m_i|f'_g(\boldsymbol{x}_i), \tau^2).$$

On the other hand, if we want to consider the absence of noise on these observations, e.g. the function to meet exactly these derivative values, the Dirac delta function δ as the observational model can be considered:

$$p(\boldsymbol{m}|\boldsymbol{f}'_C) = \prod_{i \in C} \delta(m_i - f'_g(\boldsymbol{x}_i)),$$

with $\boldsymbol{f}'_C = (f'_g(\boldsymbol{x}_1), \ldots, f'_g(\boldsymbol{x}_L))$. In the case of the GP prior model for the latent function, the partial derivative function values $\boldsymbol{f}'_C$ follow the marginalized distribution of equation (4.2), $p(\boldsymbol{f}'_C|X_C, \theta) = \mathcal{N}\big(\boldsymbol{f}'_C|K_{f'_C, f'_C}(X_C, \theta)\big)$. In the case of the splines model for the latent function, the partial derivative function f'_g follows the model in equation (4.6).

4.7.2 Monotonicity constraint

We consider a set $\boldsymbol{z} = (z_1, \ldots, z_Q)$ of virtual observations of the sign of the partial derivative of the function, with associated inputs $X_E = [\boldsymbol{x}_1 \ \boldsymbol{x}_2 \ \cdots \ \boldsymbol{x}_Q]^\top \in \mathbb{R}^{Q \times D}$, at the set $E = \{i : i = 1, \ldots, Q\}$ of observation indices. Where $z_i \in \{1, -1\}$ represents an artificial observation of $\operatorname{sign}(f'_g(\boldsymbol{x}_i))$, with $z_i = 1$ means that the partial derivative of the function is positive (increasing function) at the given data point, and $z_i = -1$ means that the partial derivative is negative (decreasing function).

The probit function $\Phi : \mathbb{R} \to (0, 1)$ can be used as a likelihood for the signs of the partial derivatives to encode this derivative information of f (provided by $\boldsymbol{z}$) into the model:

$$p(\boldsymbol{z}|\boldsymbol{f}'_E) = \prod_{i \in E} \Phi\Big(z_i \cdot f'_g(\boldsymbol{x}_i) \cdot \frac{1}{v}\Big). \tag{4.9}$$

The function Φ in equation (4.9) is the standard Normal cumulative distribution function and $v > 0$ is a parameter controlling the strictness of the constraint. When v approaches zero ($v \to 0$), the function Φ approaches a step function (Figure 4.1).

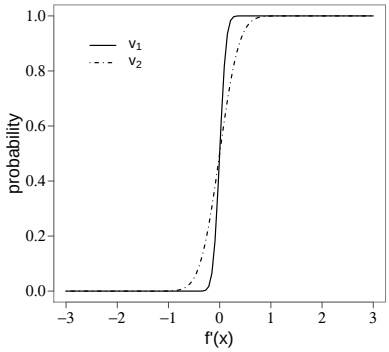


Figure 4.1: Standard normal cumulative distribution function (Probit function) for derivative values $f'(x)$, for two different values v_1 and v_2 of the parameter v, with $v_1 < v_2$.

4.8 Likelihood and posterior distributions

In this section, we illustrate the joint likelihood and joint posterior distribution for the actual observations $\boldsymbol{y}_A$, the virtual observations $\boldsymbol{y}_B$, the virtual derivative value observations $\boldsymbol{m}$ and the virtual derivative sign observations $\boldsymbol{z}$. Let $\boldsymbol{y}_B$ and $\boldsymbol{m}$ be considered with the absence of noise and models in equations (4.8) and (4.9), respectively.

Thus, the joint likelihood of these vectors of observations given $\boldsymbol{f}$, $\boldsymbol{f}_B$, $\boldsymbol{f}'_C$, $\boldsymbol{f}'_E$ and σ results:

$$
\begin{aligned}
p(\boldsymbol{y},&\boldsymbol{y}_B, \boldsymbol{m}, \boldsymbol{z}|\boldsymbol{f}, \boldsymbol{f}_B, \boldsymbol{f}'_C, \boldsymbol{f}'_E, \sigma) = \\
&p(\boldsymbol{y}|\boldsymbol{f}, \sigma)p(\boldsymbol{y}_B|\boldsymbol{f}_B)p(\boldsymbol{m}|\boldsymbol{f}'_C)p(\boldsymbol{z}|\boldsymbol{f}'_E) = \\
&\Big(\prod_{i\in A}\mathcal{N}(y_i|f(\boldsymbol{x}_i), \sigma^2)\Big)\Big(\prod_{j\in B}\delta(y_j - f(\boldsymbol{x}_j))\Big) \\
&\times \Big(\prod_{l\in C}\delta(m_l - f'_g(\boldsymbol{x}_l))\Big)\Big(\prod_{n\in E}\Phi\big(z_n \cdot f'_g(\boldsymbol{x}_n)\cdot\frac{1}{v}\big)\Big). \qquad (4.10)
\end{aligned}
$$

Posterior distribution in the case of a GP prior for the latent function f

The posterior joint distribution of parameters given the data, which is proportional to the likelihood and prior distributions results:

$$
\begin{aligned}
&p(\boldsymbol{f}, \boldsymbol{f}_B, \boldsymbol{f}'_C, \boldsymbol{f}'_E, \theta, \sigma|\boldsymbol{y}, \boldsymbol{y}_B, \boldsymbol{m}, \boldsymbol{z}) \propto \\
&\quad p(\boldsymbol{y}, \boldsymbol{y}_B, \boldsymbol{m}, \boldsymbol{z}|\boldsymbol{f}, \boldsymbol{f}_B, \boldsymbol{f}'_C, \boldsymbol{f}'_E, \sigma)p(\boldsymbol{f}, \boldsymbol{f}_B, \boldsymbol{f}'_C, \boldsymbol{f}'_E|X, X_B, X_C, X_E, \theta)p(\theta)p(\sigma),
\end{aligned}
$$

where $p(\boldsymbol{y}, \boldsymbol{y}_B, \boldsymbol{m}, \boldsymbol{z}|\boldsymbol{f}, \boldsymbol{f}_B, \boldsymbol{f}'_C, \boldsymbol{f}'_E, \sigma)$ is the likelihood of the model in equation (4.10) and $p(\boldsymbol{f}, \boldsymbol{f}_B, \boldsymbol{f}'_C, \boldsymbol{f}'_E|X, X_B, X_C, X_E, \theta)$ is the joint GP prior for regular, $\boldsymbol{f}$ and $\boldsymbol{f}_B$, and derivative, $\boldsymbol{f}'_C$ and $\boldsymbol{f}'_E$, latent function values following the equation (4.2). $p(\theta)$ contains the priors for the hyperparameters magnitud α and lengthscales $\boldsymbol{\rho}$ of the GP model. $p(\sigma)$ is the prior for the noise variance σ of the model. Based on prior knowledge of the magnitude of these parameters, we set, for example, positive half-normal prior distributions (zero-mean normal distribution limited to the positive input domain $[0, \infty)$) for the hyperparameters α, $p(\alpha) = \mathcal{N}^+(\alpha|0, 10)$, and σ, $p(\sigma) = \mathcal{N}^+(\sigma|0, 10)$, and gamma distributions for the hyperparameters $\boldsymbol{\rho}$, $p(\rho_d) = \mathrm{Gamma}(\rho_d|1, 0.1)$ for all d.

Thus, the joint posterior distribution can be expressed as:

$$
\begin{aligned}
&p(\boldsymbol{f}, \boldsymbol{f}_B, \boldsymbol{f}'_C, \boldsymbol{f}'_E, \theta, \sigma | \boldsymbol{y}, \boldsymbol{y}_B, \boldsymbol{m}, \boldsymbol{z}) \propto \\
&\Big(\prod_{i \in A} \mathcal{N}(y_i | f_i, \sigma^2)\Big)\Big(\prod_{j \in B} \delta(y_j - f_j)\Big)\Big(\prod_{l \in C} \delta(m_l - f'_l)\Big)\Big(\prod_{n \in E} \Phi\big(z_n \cdot f'_n \cdot \frac{1}{v}\big)\Big) \\
&\times \mathcal{N}\left(\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}_B \\ \boldsymbol{f}'_C \\ \boldsymbol{f}'_E \end{bmatrix} | 0, \begin{bmatrix} K_{f,f} & K_{f,f_B} & K_{f,f'_C} & K_{f,f'_E} \\ K_{f_B,f} & K_{f_B,f_B} & K_{f_B,f'_C} & K_{f_B,f'_E} \\ K_{f'_C,f} & K_{f'_C,f_B} & K_{f'_C,f'_C} & K_{f'_C,f'_E} \\ K_{f'_E,f} & K_{f'_E,f_B} & K_{f'_E,f'_C} & K_{f'_E,f'_E} \end{bmatrix}\right) \\
&\times \mathcal{N}^+(\alpha | 0, 1)\Big(\prod_{d=1}^{D} \mathrm{Gamma}(\rho_d | 1, 0.1)\Big)\mathcal{N}^+(\sigma | 0, 1). \qquad (4.11)
\end{aligned}
$$

Posterior distribution in the case of a splines model for the latent function f

The posterior joint distribution of parameters given the data, which is proportional to the likelihood and prior distributions results:

$$
\begin{aligned}
p(\boldsymbol{f}, \boldsymbol{f}_B, \boldsymbol{f}'_C, \boldsymbol{f}'_E, \boldsymbol{\beta}, \boldsymbol{b}, \sigma | \boldsymbol{y}, \boldsymbol{y}_B, \boldsymbol{m}, \boldsymbol{z}) \propto & \\
p(\boldsymbol{y}, \boldsymbol{y}_B, \boldsymbol{m}, \boldsymbol{z} | \boldsymbol{f}, \boldsymbol{f}_B, \boldsymbol{f}'_C, \boldsymbol{f}'_E, \sigma)\, g(\boldsymbol{f}, \boldsymbol{f}_B | \boldsymbol{\beta}, \boldsymbol{b}, H, Z, \Sigma) & \\
\cdot\, h(\boldsymbol{f}'_C, \boldsymbol{f}'_E | \boldsymbol{\beta}, \boldsymbol{b}, \tfrac{\partial H}{\partial x}, \tfrac{\partial Z}{\partial x}, \Sigma)\, p(\boldsymbol{\beta})\, p(\boldsymbol{b})\, p(\sigma), &
\end{aligned}
$$

where $p(\boldsymbol{y}, \boldsymbol{y}_B, \boldsymbol{m}, \boldsymbol{z} | \boldsymbol{f}, \boldsymbol{f}_B, \boldsymbol{f}'_C, \boldsymbol{f}'_E, \sigma)$ is the likelihood of the model in equation (4.10), $g(\boldsymbol{f}, \boldsymbol{f}_B | \boldsymbol{\beta}, \boldsymbol{b}, H, Z, \Sigma)$ is the splines model for regular $\boldsymbol{f}$ and $\boldsymbol{f}_B$ observations in equation (4.5) and $h(\boldsymbol{f}'_C, \boldsymbol{f}'_E | \boldsymbol{\beta}, \boldsymbol{b}, \frac{\partial H}{\partial x}, \frac{\partial Z}{\partial x}, \Sigma)$ is the splines model for derivative $\boldsymbol{f}'_C$ and $\boldsymbol{f}'_E$ observations in equation (4.6). $p(\boldsymbol{\beta})$ and $p(\boldsymbol{b})$ contain the priors for the linear and splines coefficients $\boldsymbol{\beta}$ and $\boldsymbol{b}$, respectively. $p(\sigma)$ is the prior for the noise variance σ of the model. Based on prior knowledge of the magnitude of these parameters, we define, for example, normal prior distributions for the linear coefficients $\boldsymbol{\beta}$, $p(\boldsymbol{\beta}) = \mathcal{N}(\boldsymbol{\beta} | 0, 10I)$, and for the splines coefficients $\boldsymbol{b}$, $p(\boldsymbol{b}) = \mathcal{N}(\boldsymbol{b} | 0, 10I)$, where I is the identity matrix, and positive half-normal prior distributions for the noise variance σ, $p(\sigma) = \mathcal{N}^+(\sigma | 0, 10)$.

Thus, the joint posterior distribution can be expressed as:

$$p(\boldsymbol{f}, \boldsymbol{f}_B, \boldsymbol{f}'_C, \boldsymbol{f}'_E, \boldsymbol{\beta}, \boldsymbol{b}, \sigma | \boldsymbol{y}, \boldsymbol{y}_B, \boldsymbol{m}, \boldsymbol{z}) \propto$$
$$\Big(\prod_{i\in A} \mathcal{N}(y_i|f_i, \sigma^2)\Big)\Big(\prod_{j\in B} \delta(y_j - f_j)\Big)\Big(\prod_{l\in C} \delta(m_l - f'_l)\Big)\Big(\prod_{n\in E} \Phi\big(z_n \cdot f'_n \cdot \frac{1}{v}\big)\Big)$$
$$\times \mathcal{N}(\beta_1|0,1)\mathcal{N}(\beta_2|0,1)\Big(\prod_{k=1}^{K} \mathcal{N}(b_k|0,1)\Big)\mathcal{N}^+(\sigma|0,1). \tag{4.12}$$

4.9 Issues of using virtual derivatives observations for monotonicity

In this section, we analyze the overly smoothing effect of using virtual derivatives observations for inducing monotonicity on functions. We analyze this effect on both of using a GP prior and a splines model for the latent function f. With this purpose, the models are fitted to a simulated dataset from the *inverse function*, $f(x) = \frac{-1}{x}$, with additive Gaussian noise. The simulated dataset consists of $N = 30$ single draws $\boldsymbol{y} = (y_1, \ldots, y_N)$ from the inverse function, with corresponding inputs values $\boldsymbol{x} = (x_1, \ldots, x_N)$ with $x_i \in [-1.5, 1.5] \subset \mathbb{R}$. To form the final noisy dataset $\boldsymbol{y}$, Gaussian noise $\sigma = 1.0$ was added to the function draws. The first 20 data points are used for training the model and the last 10 as testing data points for visually assessing the extrapolation performance of the models, as can be appreciated in Figures 4.2 and 4.3.

We consider the observational model for these set of regular observations as in equation (4.1), in a 1D input space and latent function $f : \mathbb{R} \to \mathbb{R}$.

In order to induce monotonicity on the function, we consider a set $\boldsymbol{z} = (z_1, \ldots, z_M)$ of virtual observations of the sign of the partial derivative of the function, where M denotes the number of virtual observations considered, with associated inputs $\boldsymbol{w} = (w_1, \ldots, w_M)$, with $w_i \in [-1.5, 1.5]$, at the set $E = \{i : i = 1, \ldots, M\}$ of observation indices. Where $z_i \in \{1, -1\}$ represents an artificial observation of $\text{sign}(f'_g(w_i))$. The observational model for this vector $\boldsymbol{z}$ of observations is as in equation (4.9). The parameter v that controls the strictness of the constraint is set to 10^{-4}.

In the case of using a GP prior model for the latent function f, as developed in Section 4.5.1, the joint prior distribution of regular $\boldsymbol{f}$ and derivative $\boldsymbol{f}'$ function values is as in equation (4.2). Thus, the marginal posterior distribution of $\boldsymbol{f}$ can be computed integrating out the joint posterior distribution $p(\boldsymbol{f}, \boldsymbol{f}', \theta, \sigma | \boldsymbol{y}, \boldsymbol{z})$ over the

hyperparameters $\theta = (\rho, \alpha)$ and σ:

$$
\begin{aligned}
p(\boldsymbol{f}) = \int & p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{f}, \boldsymbol{f}', \sigma)\, p(\boldsymbol{f}, \boldsymbol{f}'|\boldsymbol{x}, \boldsymbol{w}, \theta)\, p(\rho)\, p(\alpha)\, p(\sigma)\ \mathrm{d}(\alpha)\ \mathrm{d}(\rho)\ \mathrm{d}(\sigma) = \\
& \int \Big(\prod_{i \in A} \mathcal{N}(y_i|f(x_i), \sigma^2) \Big) \Big(\prod_{j \in E} \Phi\big(z_j \cdot f'(w_j) \cdot \frac{1}{v}\big) \Big) \\
& \times \mathcal{N}\left(\left[\begin{array}{c} \boldsymbol{f} \\ \boldsymbol{f}' \end{array} \right] |0, \left[\begin{array}{cc} K_{f,f}(\boldsymbol{x}, \theta) & K_{f,f'}(\boldsymbol{x}, \boldsymbol{w}, \theta) \\ K_{f',f}(\boldsymbol{w}, \boldsymbol{x}, \theta) & K_{f',f'}(\boldsymbol{w}, \theta) \end{array} \right] \right) \\
& \times \mathcal{N}^+(\alpha|0, 10)\ \mathrm{Gamma}(\rho|1, 0.1)\, \mathcal{N}^+(\sigma|0, 10)\, \mathrm{d}(\alpha)\, \mathrm{d}(\rho)\, \mathrm{d}(\sigma).
\end{aligned}
$$

Figure 4.2-(a) shows the mean posterior functions for this model with a GP prior for the function f. In red color, the posterior function without virtual observations of the sign of the derivative for inducing monotonicity is plotted. In graduated blue colors, the posterior functions with different number of virtual observations (different subsets E of virtual points) are plotted. The virtual points are placed evenly over the input space. As commented above the last 10 regular observations of the dataset (plotted as crosses in Figure 4.2-(a)) are used for testing model extrapolation of the predictive posterior functions. Figure 4.2-(b) shows a scatter plot with the posterior mean of the derivatives per pair of points when using different number of virtual observations for monotonicity. This graph aims to illustrate the correlation of the derivatives between pair of points of the function. The closer to the diagonal the higher the correlation. Figure 4.2-(c) shows a zooming of Figure 4.2-(b).

In the case of using a splines model for the latent function f, as developed in Section 4.5.2, the marginal posterior distribution of function values $\boldsymbol{f}$ can be computed integrating out the joint posterior distribution $p(\boldsymbol{f}, \boldsymbol{f}', \boldsymbol{\beta}, \boldsymbol{b}, \sigma|\boldsymbol{y}, \boldsymbol{z})$ over the hyperparameters $\boldsymbol{\beta}$, $\boldsymbol{b}$ and σ:

$$
\begin{aligned}
p(\boldsymbol{f}) = \int & p(\boldsymbol{y}, \boldsymbol{z}|\boldsymbol{f}, \boldsymbol{f}', \sigma)\, g(\boldsymbol{f}|\boldsymbol{\beta}, \boldsymbol{b}, H, Z, \Sigma)\, h(\boldsymbol{f}'|\boldsymbol{\beta}, \boldsymbol{b}, \tfrac{\partial H}{\partial w}, \tfrac{\partial Z}{\partial w}, \Sigma) \\
& \cdot p(\boldsymbol{\beta})\, p(\boldsymbol{b})\, p(\sigma)\, \mathrm{d}(\boldsymbol{\beta}) \Big(\prod_{k}^{K} \mathrm{d}(b_k) \Big)\, \mathrm{d}(\sigma) = \\
& \int \Big(\prod_{i \in A} \mathcal{N}(y_i|f(x_i), \sigma^2) \Big) \Big(\prod_{j \in E} \Phi\big(z_j \cdot f'(w_j) \cdot \frac{1}{v}\big) \Big) \mathcal{N}(\beta_1|0, 1) \mathcal{N}(\beta_2|0, 1) \\
& \times \Big(\prod_{k}^{K} \mathcal{N}(b_k|0, 1) \Big) \mathcal{N}^+(\sigma|0, 1)\, \mathrm{d}(\beta_1)\, \mathrm{d}(\beta_2)\, \mathrm{d}(b_1) \cdots \mathrm{d}(b_K)\, \mathrm{d}(\sigma),
\end{aligned}
$$

where $f(x_i)$ and $f'(w_j)$ follow the model in equations (4.5) and (4.6), respectively. Similarly to Figure 4.2 explained above, in Figures 4.3 the posterior mean function (Figure 4.3-(a)) and the mean derivatives per pair of points (Figure 4.3-(b)) for different number of virtual points for inducing monotonicity are plotted.

We can see in Figures 4.2-(a) and 4.3-(a) how the posterior distribution depends on the number and locations of virtual points; different posteriors are obtained with different numbers of virtual points. Furthermore, monotonicity of the posterior functions is not guaranteed, especially when the number of virtual points is too small. Furthermore, the posterior functions tend to be overly smoothed as the number of virtual points increases, especially when using a GP prior for f (Figure 4.2-(a)). In the case of splines, the functions seem not to be so overly smoothed and fit better to the dynamics of the process (Figure 4.3-(a)). The correlation between pair of derivatives values is higher for the use of a GP prior for function f since points are closer to the diagonal in Figure 4.2-(c) than in Figure 4.3-(c).

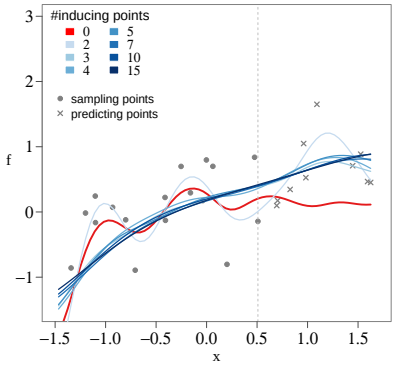
In monotonic functions, the function values at two arbitrary locations (x, x') can never be independent: $x' > x$ implies $f(x') \geq f(x)$. When using virtual observations to induce monotonicity, the assumption of independence between function values $f(x)$ y $f(x')$ is violated. In GP functions, in addition, monotonic functions do not have a characteristic lengthscale and, as the function values at the virtual points can no be independent, the lengthscale tends to increase and posterior functions are smoother and flatter.

However, monotonic functions with a GP prior to function f provides reliable model extrapolation since it has a stronger inductive bias than the model without virtual points for monotonicity, as can be seen in Figure 4.2-(a). On the other hand, monotonic functions with a splines model for function f extrapolate considerably worse than with GP priors.

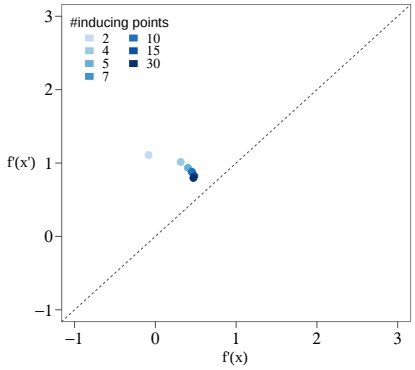
4.10 Conclusion

In this chapter, we have illustrated the use of derivative information in the modeling as additional observations. This methodology can be used to control or constrain the dynamics of stochastic functions, such as gradients, monotonicity or unimodality properties of the functions. We have illustrated this for GP prior models and splines models for stochastic functions.

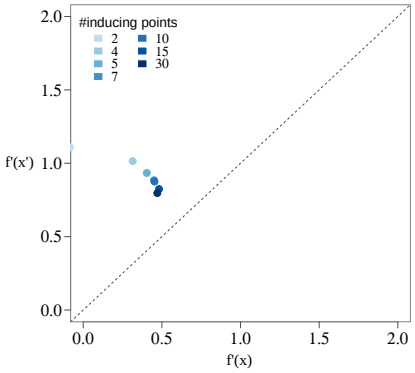
Furthermore, we have analyzed the main issues that can arise when using many virtual derivative sign observations for monotonicity. First, the likelihood and posterior distribution depend on the number and locations of the virtual points. And secondly, the correlation between pairs of points is larger as the number of virtual



(a) Mean posterior functions of the regular GP (red line) and the monotonic GP with different number of inducing points (virtual points) (blue lines).

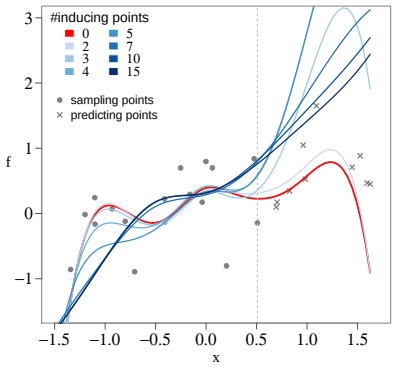


(b) Derivative posterior means per pair of points (x, x') for monotonic GP with different number of inducing points (virtual points).

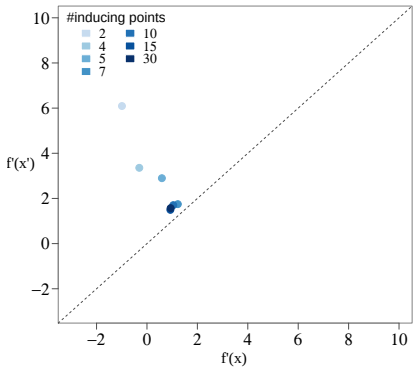


(c) Zooming of Figure 4.2(b).

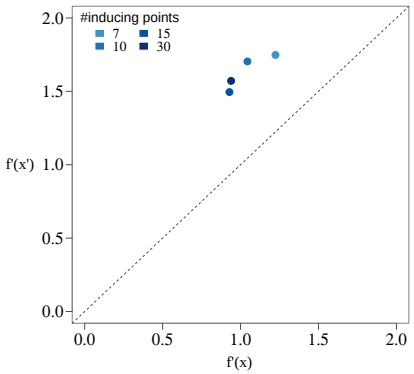
Figure 4.2



(a) Mean posterior functions of the regular splines (red line) and the monotonic splines with different number of inducing points (virtual points) (blue lines).



(b) Derivative posterior means per pair of points (x, x') for monotonic splines with different number of inducing points (virtual points).



(c) Zooming of Figure 4.3(b).

Figure 4.3

points for monotonicity increases, causing overly smoothed posterior distributions, especially when using GPs. In a Bayesian perspective, the monotonicity constraint should be specified in the prior rather than in the likelihood function through virtual observations.

Placing appropriately the virtual points for monotonicity is highly recommended, and placing as less virtual points as possible, as long as the posterior function is monotonic, is also recommended. These considerations apply for monotonic functions with both GP and splines models for the latent function f, although this is extremely recommendable for GPs.

While the use of virtual observations for monotonicity has some issues, as studied in this chapter, it has some interesting properties such as to be a flexible procedure in order to induce different monotonicity in different parts of the input space or in different input dimensions, as long as this is combined with an appropriate placement of the virtual points. Furthermore, it has been demonstrated in the present study how the use of virtual points for monotonicity helps for extrapolating data, especially with GPs.

Finally, as future research we propose the systematic study of using additional information of the functions such as constraints regarding the function values or constraints regarding the gradients of the function, jointly with the virtual observations for monotonicity. This might help for minimizing the over-smoothing effect of using virtual points for monotonicity.

Chapter 5

Application to rock art paintings: Models with derivative information for modeling microfading spectrometry measurements

In this chapter, an applied study on a specific problem is carried out. The goals of this applied study are to model and make predictions of microfading spectrometry (MFS) measurements for new unobserved spatial locations on the surface of rock art paintings. The main modeling aspects that this application requires to be solved is the definition of spatio-temporal correlation structures for the data and the specification of modeling constraints in order to control the dynamics (derivative properties) of the predicted functions. A GP (introduced in Chapter 3) with derivatives (introduced in Chapter 4) is the model used in the application. Furthermore, with the aim of comparison, the problem is also solved by formulating a spatially correlated splines time-series model, with the consideration of model derivatives. A univariate splines model with derivative information was introduced in Chapter 4 and, in this chapter, we illustrate how to correlate different univariate splines models, e.g. spatially located splines time-series models, by correlating their splines coefficients.

5.1 Summary

Microfading spectrometry (MFS) is a method for assessing the photostabilty of cultural heritage objects to light fading, allowing real time monitoring of spectral response, i.e. color changes of the surface of the material. The MFS technique provides measurements of the surface under study, where each point of the surface gives rise to a time-series that represents potential spectral (color) changes due to sunlight exposition over time. Thus, MFS measurements can be seen as observations of an underlying spatio-temporal stochastic process.

Color fading is expected to be non-decreasing as a function of time and stabilize eventually. These properties can be expressed in terms of the partial derivatives of the functions. In this work we propose two modeling approaches, a spatio-temporal GP model and a spatially correlated splines-based time-series model. Both models take the derivative information into account by jointly modeling the regular process and its derivative process, in order to force the functions to fit the properties of non-decreasing and stabilize eventually as a function of time.

A spatio-temporal structure in a GP is easily implemented in a multi-dimensional covariance function in the GP prior. However, two-way (space and time) splines model already becomes highly parameterized and hardly interpretable. Instead, we correlate the different (spatially located) splines time-series by correlating their coefficients across time-series by defining prior distribution with covariance structure, e.g. GP prior, on them.

A multivariate GP prior model is the more natural way to deal with multi-dimensional (space and time) data but requires high computational expense in matrix inversion as it increases rapidly with the spatio-temporal dimensionality. On the other hand, a spatially correlated time-series model, which correlates the splines time-series through correlating their splines coefficients, requires less computation, since it depends on the order K of the splines functions instead of the time dimensionality T of the GP model (usually $K \ll T$). Furthermore, different levels of complexity of the covariance structure, with different computational expenses, can be defined.

We fitted the two proposed models to MFS data collected from the surface of prehistoric rock art paintings. We demonstrated that the colorimetric variables are useful for predicting the color fading time-series for new unobserved spatial locations. Furthermore, constraining the model using derivative observations and derivative sign observations for monotonicity was shown to be beneficial in terms of both predictive performance and application-specific interpretability. Even though a multivariate GP prior model is the more natural way for modeling this type of data, we demonstrate that a spatially correlated time-series model is good enough for modeling this study case, achieving similar results to the GP model with the

advantage of a considerable reduction of computation.

5.2 Introduction

Prehistoric rock art paintings are exposed to environmental elements, which can accelerate their degradation, increasing the risk of losing such a valuable piece of information about past societies. Apart from and in addition to many other factors, exposure to sunlight can have adverse effects on these systems due to thermal and photochemical degradation of the historic materials, and changes in the spectral properties of the materials is one of its main effects which is mainly related to the physicochemical properties of the materials [del Hoyo-Meléndez et al., 2015, Díez-Herrero et al., 2009]. In this study we focused on the study and documentation of the degree of color changing/fading of paintings, patinas and host rock which is crucial for the conservation of these systems [Cassar et al., 2001].

It is known that materials with higher light sensitivity usually experience a rapid color change during the early stages of exposure, followed by a slower rate after maximum fading has occurred, assuming total disappearance of the atoms (chromophore) of the molecules of the substance that produces the color at this second stage of the fading [Feller et al., 1986, Giles, 1965, Giles et al., 1968, Johnston-Feller et al., 1984]. Thus, fading can not decrease with time and it is expected to stabilize in the long term. Materials can show different times to saturation depending on their physicochemical properties and concentration of chromophores.

The MFS technique is a method for assessing the susceptibility of cultural heritage objects to light fading [Columbia et al., 2013, Ford, 2011, Ford and Druzik, 2013, Tse et al., 2010]. Each measured point of the surface under study gives rise to a time-series that represents potential color fading due to light exposition over time [del Hoyo-Meléndez and Mecklenburg, 2010, Whitmore et al., 1999]. Thus, MFS measurements can be seen as observations of an underlying spatio-temporal stochastic process. MFS time-series represents potential color fading from the current state of the materials.

The MFS instrument is very sensitive to movement and glossy surface effects, occasionally causing extremely large fluctuations in color fading values registered during measurements. Furthermore, collected data can be easily contaminated by changes in the illumination conditions when performed in outside environments, as it is the case of the surface of rock art paintings. These large fluctuations and possible systematic noise effects in the observations can cause that models do not satisfy those properties of monotonicity and long-term stabilization of color fading over time. Thus, in order to meet these properties and to ensure reliable properties

for color fading estimates in new unobserved locations, it is recommended to include additional information in the models.

Furthermore, existing lightfastness studies on these systems have been limited to analyze the few measured points on the surface of the rock art paintings due to the difficulty to set up the instrument, especially under harsh conditions.

So, in this paper, we propose two reliable modeling frameworks: one based on Gaussian processes (GPs) and the other on spatially correlated splines time-series. In both models, the regular process is jointly modeled with the derivative process. These models aim at rigorously extending the analysis of MFS color-fading in many other unobserved points on the surface of rock art paintings. Forecasting potential color-fading in every point of the surface of rock art paintings will be an important and useful information in order to achieve further successful conservation actions on these systems.

5.3 State of the art

Functional data usually refers to independent realizations of a functional random variable that takes values in a continuous space [Ramsay and Silverman, 2007]. Time-series of observations (e.g. color-fading time-series functions) might be the most common case of functional data in 1D, $f(x) : x \in \mathbb{R} \rightarrow \mathbb{R}$, but spatially distributed observations can also be seen as functional data in a 2D space, $f(x) : x \in \mathbb{R}^2 \rightarrow \mathbb{R}$, or spatio-temporal observations can also be seen as functional data in a 3D space, $f(x) : x \in \mathbb{R}^3 \rightarrow \mathbb{R}$.

In order to construct a model useful for making predictions of new functional data as a function of new values of the variables in the input space, the process must be considered as an structured process with dependence among observations.

Correlated functional models consider the observed functional data as non-independent functions [Delicado et al., 2010]. A popular approach for correlated space-time functional data consists in three-way (spatial (2D) and temporal (1D)) penalized splines models [Wood, 2003] with different basis constructions based on Kronecker products [Currie et al., 2006, Lee and Durbán, 2011] or additive basis components [Kneib and Fahrmeir, 2006]. Aguilera-Morillo et al. [2017] propose a mixture of the functional regression model for functional response and penalized spline spatial regression. However, in general, these models have a large number of parameters and become hardly interpretable.

Another and powerful approach consists of considering the space-time structured observations as stochastic realizations of a GP prior with a spatio-temporal covariance function. GP [Neal, 1999, Rasmussen and Williams, 2006] is a natural and flexible non-parametric prior model for multi-dimensional functions and with

multivariate predictors (input variables) in each dimension. Furthermore, GP is sufficiently flexible to model complex phenomena since it allows possible non-linear effects and can handle interactions between input variables implicitly. GPs are easy to generalize/change to new models by changing the covariance function. For a review of the different covariance functions in Gaussian processes, see Rasmussen and Williams [2006]. In a separable form, the space-time covariance function is a result of two independent processes, space and time [Banerjee et al., 2014]. In a non-separable form, the covariance function models space-time interaction [Cressie and Huang, 1999, De Iaco et al., 2002]. However, for parametric or semiparametric models, it is not so easy and natural the generalization to spatio-temporal covariance structures. For instance, the two-way splines model already becomes highly parameterized and hardly interpretable. As an advantage, the time of computation for these semiparametric models is faster.

A geostatistical approach for spatio-temporal data is the kriging approach for 1D-functional data [Delicado et al., 2008, Giraldo et al., 2010], with the functional data consisting of the time-series of the data. The spatial correlation of the time-series is modeled in the covariance function. The dimension of the covariance matrix is $N \times N$ and the matrix inversion is a $O(N^3)$ operation, with N the number of spatial locations. Although this approach requires less computation that the spatio-temporal GPs, it has the drawback of being a quite less flexible model in the spatio-temporal structure since the same spatial structure is defined for the whole time-series. A related approach can be found in Baladandayuthapani et al. [2008] where the spatial correlation between the time-series is modeled by defining GPs with a spatial covariance function across the time-series functional coefficients. This construction allows for modeling different covariance structures for the splines coefficients.

Regarding the methods for inducing monotonicity on functions and using derivative information, they were already discussed in Chapter 4.

5.4 Objectives and methodology of the study

In this chapter, a specific application aiming at modeling and predicting MFS color fading time-series for new unobserved spatial locations on the surface of rock art paintings is presented. The main motivation of this study is to construct a model that exploits to the maximum the correlation structure of the data in order to extend the analysis and make useful predictions, in an scenario of a short set of sampling observations, as it is the case of MFS measurements on rock art paintings. In fact, in the present study case only 13 measured locations on the surface were collected. A multivariate GP prior model is the more natural way to accomplish this objective

for this type of data. Nevertheless, and in addition to GPs, in this work we also formulate a spatially correlated splines time-series model in order to make model comparison.

Thus, in this study we apply two modeling approaches for correlated functional data in order to accomplish the objectives and compare their results:

1. A Gaussian process model with a multi-dimensional covariance function.

   A multi-dimensional (e.g. space and time) covariance function is the key element of a GP as it encodes the functional relationship and defines the correlation structure which characterizes the correlation between function values at different inputs. Furthermore, the covariance function in a GP can be extended to jointly model the covariances of regular and derivatives observations, increasing thus the predictive capacity of the model and, at the same time, guaranteeing that the predictions are monotonically non-decreasing and stabilize eventually as a function of time.

2. A spatially correlated splines time-series model.

   The color-fading time-series are modeled as penalized cubic-splines functions. The spatial correlation of the time-series is established by defining multivariate Gaussian process priors distributions over the splines coefficients across time-series. The derivative of a semiparametric model such as a splines model is still a linear model and can be used as an additional observation together with actual observations, thus encoding derivative information of the functions into the modeling and induce monotonicity and long-term stabilization as a function of time.

Apart from the regular observations $\boldsymbol{y}$, we want to include partial derivative observations and first order constraints in the model. Color degradation is expected to be non-decreasing as a function of time and stabilize in the long term. These properties can be expressed in terms of the first order partial derivative of the functions.

In both modeling frameworks, the regular process is jointly modeled with the derivative process. This makes it possible to use derivative observations jointly with regular observations. Derivative observations of both the sign and the values of the partial derivatives are used to induce monotonicity (non-decreasing) and long-term saturation, respectively, as a function of time. As studied in Chapter 4, can be practical issues with the approach of inducing monotonicity through the use of additional (virtual) derivative sign observations. These issues are due to the fact that the monotonicity information is included in the likelihood of the model through virtual observations instead of into the prior of the function, which makes

the posterior distribution of the function dependent on the number and location of the virtual points. When the number of virtual points is too large, the posteriors tend to be overly smoothed. However, if the function to be learned is smooth, this problem can be avoided in practice choosing a few virtual points only and placing them appropriately.

On the other hand, in order to force the functions to be zero at the starting points of every time-series, observations with the absence of noise are used at these points.

Physicochemical data for all the points on the surface as inputs to predict new curves are hard to obtain. Instead, image color values can be used as surrogate input variables to construct and evaluate the correlation, since these image color variables are known to be related to the physicochemical properties of the imaged material [Malacara, 2011]. A multivariate covariance function in a GP allows modeling trichromatic image color variables jointly with spatial distances as inputs to evaluate the covariance structure of the data.

In order to conduct model evaluation and comparison, the same models but without derivative information are also fitted. Cross-validation procedures are conducted to compute the *posterior predictive checks*, the *expected log predictive density* and the *mean square predictive errors* in order to do model checking and assessment of the predictive performance.

The rest of the chapter is structured as follows. Section 5.5 describes the case study and the available data in detail. Sections 5.7 and 5.8 focus on the modeling and inference formulation of both modeling proposals. Section 5.10 describes the model checking and model selection procedures. Section 5.11 describes the results of applying the proposed models on the data set. Section 5.12 discusses about the results and modeling. Finally, Section 5.13 presents a brief conclusion of the work.

5.5 Case study and data description

In this practical case, we seek to evaluate the degree of color fading over time and space on rock art paintings caused by direct solar irradiation. The goal is to construct a model for the set of MFS data and make predictions at new unobserved locations under the surface of rock art paintings as a function of new input values.

The study area is located in cova Remigia rock art shelter, Castellón (Spain). Some of its paintings, which are included in UNESCO's World Heritage List, are exposed to environmental factors, including the natural daylight depending on the time of the day and the season of the year. It is well-known that exposure to sunlight can have adverse effects on these systems due to thermal and photochemical degradation of the historic materials [del Hoyo-Meléndez et al., 2015].

Each measured point on the surface gives rise to a time-series that represents

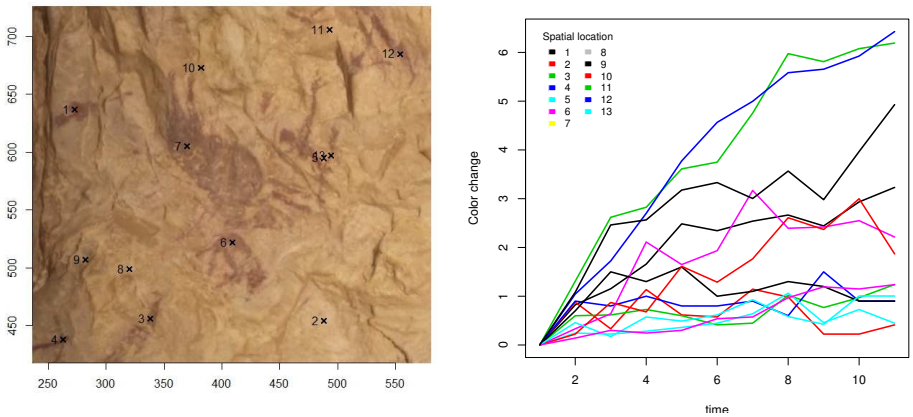


Figure 5.1: Image with the points where the MFS observations were measured in Shelter V of Cova Remigia (left), and observed MFS time-series (right).

color fading over time. Color fading can be described in terms of CIE color differences ΔE^*_{ab} [Malacara, 2011]. The MFS measurements have a duration of 10 minutes once setup. The sampling frequency will be once per minute, such that the resulting time-series will consist of 11 time points $(t = 1, \ldots, 11)$. Thus, the spatio-temporal MFS data set consist of 13 observed locations on the surface $(N = 13)$. Figure 5.1-left shows their pixel locations on a color image of the study area; each location incorporates color fading time-series of observations (Figure 5.1-right). The available input variables for every spatial location are the three image color variables, Hue (H), Saturation (S) and Intensity (I), and the two spatial coordinates S_x and S_y. In the temporal dimension, the input variable is the collection of time points, $t = 1, ..., 11$, of the MFS time series. Table 5.1 presents summary statistics for the input variables H, S, I, S_x, and S_y. The input variables H, S, and I have been re-scaled by dividing by their standard deviations. In the case of S_x and S_y, they were jointly re-scaled by dividing by their common standard deviation.

Table 5.1: Summary statistics of the input variables

	H	S	I	S_x	S_y
Mean	5.255	9.704	5.155	3.549	4.969
Std. Dev.	1.0	1.0	1.0	0.732	0.674
1st Qu.	4.359	9.042	4.603	2.910	4.387
3rd Qu.	5.913	10.273	5.772	4.187	5.551

Due to the large fluctuations present in the measurements conducted on these

rock art systems, some modeling issues can arise, such as not starting at zero, not being monotonically increasing or not stabilizing in the long term as a function of time, which are properties assumed for the color fading curves, as discussed above.

Thus, the functions must be constrained to be zero at the subset $B = \{(i,t) : t = 1\}$ of starting points of every time-series, that is, the time-series must start in zero.

Most of the rock art painting systems analyzed so far show stabilization at or before the 10 minutes of MFS monitoring measurements. Therefore, at the subset $C = \{(i,t) : t = 11\}$ of ending sampling points of every time-series, the functions will be constrained to reach a stabilization as a function of time.

Furthermore, the functions are guaranteed to be monotonically non-decreasing as a function of time when the partial derivative is non-negative. Thus, virtual points for monotonicity will be appropriately placed at two time points of every time-series, denoted by subset $E = \{(i,t) : t \in \{7, 10\}\}$ of observation indices. We only use two virtual points in every time-series to prevent the posterior functions to be overly smoothed, as studied in Chapter 4. At the same time, the use of two virtual points only will probably be enough to ensure monotonicity in expectation for the entire time-series, since the time-series functions to be learned are expected to be smooth.

In the case of the spatially correlated splines models, each time-series is modeled as a cubic-splines function. The order, or number of spline knots K, of the cubic-splines functions is set to $K = 3$ ($k = 1, \ldots, K$) and placed uniformly through the time points variable.

A comparison with the same models but without derivative information is carried out and evaluated in terms of both predictive performance and application-specific interpretability.

The actual equivalency of the exposure time used in MFS in years depends on the hours and intensity of sunlight that affects the paintings on a changing daily basis. Without proper monitoring of light, this equivalency is difficult to obtain, so this aspect of the research was not considered in the current study.

5.6 Observational model

The MFS dataset consist of observations of a continuous stochastic process represented as a matrix of spatio-temporal output observations $Y \in \mathbb{R}^{N \times T}$ where the element y_{it} denotes the observation at the ith location (row) at time t (column), with $i = 1, \ldots, N$ representing the spatial locations and $t = 1, \ldots, T$ the time points. As described in previous Section 5.5, the MFS dataset consists of $N = 13$ spatial locations and $T = 11$ time points.

We consider separable input variables in the spatial and temporal dimensions of

the input space. Thus, a matrix of inputs variables $X_S = [\boldsymbol{x}_{S_1}\ \boldsymbol{x}_{S_2}\ \cdots\ \boldsymbol{x}_{S_N}]^\top \in \mathbb{R}^{N\times 5}$ is associated to the observations in the spatial input dimension, where

$$\boldsymbol{x}_{S_i} = (H_i, S_i, I_i, S_{x_i}, S_{y_i}) \in \mathbb{R}^5 \tag{5.1}$$

is the row-vector of inputs values for the ith spatial location. And, a vector of inputs variables $\boldsymbol{x}_T = (t_1, \ldots, t_T)^\top \in \mathbb{R}^T$ is associated to the observations in the temporal input dimension, where an element t_l is the lth time point, with $l = 1, \ldots, T$.

We adopt the model $y_{it} = f_{it} + e_{it}$, where f_{it} is the value of the function $f : \mathbb{R}^6 \to \mathbb{R}$ underlying to the observations and evaluated at the input values $\boldsymbol{x}_{S_i}$ and t, i.e. $f_{it} = f(\boldsymbol{x}_{S_i}, t)$. And e_{it} is the Gaussian noise. Let $F \in \mathbb{R}^{N\times T}$ the matrix of latent function values.

So, the observational model for the data Y given the latent function values F can be written as follows

$$p(Y|F) = \prod_{i,t} \mathcal{N}(y_{it}|f_{it}, \sigma^2).$$

5.7 Latent Gaussian process model with derivative information

First of all, let's join the X_S and X_T matrices of input variables as follows,

$$X = [X_S \otimes \boldsymbol{I}_1 \ \ X_T \otimes \boldsymbol{I}_2] \in \mathbb{R}^{NT\times 6},$$

where the operator $\otimes$ denotes the Kronecker product, and $\boldsymbol{I}_1 \in \mathbb{R}^T$ and $\boldsymbol{I}_2 \in \mathbb{R}^N$ are identity column-vectors. A row (it) of matrix X,

$$\boldsymbol{x}_{it} = (\boldsymbol{x}_{S_i}, t) = (H_i, S_i, I_i, S_{x_i}, S_{y_i}, t) \in \mathbb{R}^6, \tag{5.2}$$

is the vector of spatio-temporal input values at ith spatial location and tth time point.

In addition, let us re-arrange the matrices of spatio-temporal observations Y and function values F into the column-vectors of NT-dimensionality $\boldsymbol{y} \in \mathbb{R}^{NT}$ and $\boldsymbol{f} \in \mathbb{R}^{NT}$, respectively.

5.7.1 Separable spatio-temporal Gaussian process model

The latent function f is assumed to follow a zero mean GP prior

$$f \sim \mathcal{GP}(0, \alpha^2\, k_S(\boldsymbol{x}_S, \boldsymbol{x}'_S)\, k_T(t, t')), \tag{5.3}$$

where $k_S(\boldsymbol{x}_S, \boldsymbol{x}'_S) : \mathbb{R}^5 \times \mathbb{R}^5 \to \mathbb{R}$ and $k_T(t, t') : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ are the covariance functions for the spatial and temporal input dimensions, respectively, and $\boldsymbol{x}_S$ and $\boldsymbol{x}'_S$ denote a pair of spatial input vectors and t and t' denote a pair of time points. The parameters α is the overall magnitude of the GP prior model.

We consider a stationary and separable exponentiated quadratic covariance function for both k_S and k_T covariance functions. Thus, the covariance between the pair (i, j) of spatial locations is:

$$k_S(\boldsymbol{x}_{S_i}, \boldsymbol{x}_{S_j}) = \exp\left(-\frac{1}{2}\sum_{d=1}^{5}\frac{1}{\rho_d^2}(x_{S_{i,d}} - x_{S_{j,d}})^2\right),$$

where $x_{S_{id}}$ and $x_{S_{jd}}$ are the dth spatial input values for the ith and jth observations, respectively, and ρ_d is the lengthscale associated to the dth input variable in the spatial dimension, $d = 1, \ldots, 5$, which controls the rate of decay of the correlation in the direction of the dth input variable. Let us collect these lengthscale parameters into the vector $\boldsymbol{\rho}_S = (\rho_1, \ldots, \rho_5)$. The variables $x_{S_{i4}} = S_{x_i}$ and $x_{S_{i5}} = S_{y_i}$ share the same lengthscale, such that $\rho_4 = \rho_5$, which makes the covariance function dependent on the Euclidean distance between spatial coordinates. Lengtshscales are strictly positive parameters. And, the covariance between the pair (t_l, t_n) of time points is:

$$k_T(t_l, t_n) = \exp\left(-\frac{1}{2\rho_T^2}(t_l - t_n)^2\right),$$

where t_l and t_n are the lth and nth time points, respectively, and ρ_T is the lengthscale associated to the time input variable.

Let $\boldsymbol{f} \in \mathbb{R}^{NT}$ a column-vector where the element (i, t) is the value of the GP latent function f in equation (5.3) evaluated at the inputs $\boldsymbol{x}_{it}$, i.e. $\boldsymbol{f}_{it} = f(\boldsymbol{x}_{it})$. Then the resulting prior distribution for $\boldsymbol{f}$ is a multivariate Gaussian distribution $\boldsymbol{f} \sim \mathcal{N}(\boldsymbol{\mu}, \alpha^2\boldsymbol{K})$, where $\boldsymbol{K} \in \mathbb{R}^{NT\times NT}$ is the spatio-temporal covariance matrix with element $K_{(i,l),(j,n)} = k_S(\boldsymbol{x}_{S_i}, \boldsymbol{x}_{S_j})\, k_T(t_l, t_n)$, with $i, j = 1, \ldots, N$ and $l, n = 1, \ldots, T$. Thus the element $((i, l), (j, n))$ of the covariance matrix $\boldsymbol{K}$ is:

$$\begin{aligned} K(X_S, \boldsymbol{\rho}_S, X_T, \boldsymbol{\rho}_T)_{(i,l),(j,n)} = \\ \exp\left(-\frac{1}{2}\sum_{d=1}^{5}\frac{1}{\rho_d^2}(x_{S_{i,d}} - x_{S_{j,d}})^2\right)\exp\left(-\frac{1}{2\rho_T^2}(t_l - t_n)^2\right) = \\ \exp\left(-\frac{1}{2}\Big(\sum_{d=1}^{5}\frac{1}{\rho_d^2}(x_{S_{i,d}} - x_{S_{j,d}})^2 + \frac{1}{\rho_T^2}(t_l - t_n)^2\Big)\right) \end{aligned} \tag{5.4}$$

The previous equation (5.4) can also be expressed in terms of the vector of inputs $\boldsymbol{x}_{it}$ in equation (5.2)

$$K(X, \boldsymbol{\rho})_{(i,l),(j,n)} = \exp\left(-\frac{1}{2}\sum_{d=1}^{6}\frac{1}{\rho_d^2}(x_{il,d} - x_{jn,d})^2\right)$$

with vector of lenghtscales $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_6) = (\boldsymbol{\rho}_S, \rho_T)$. Notice that the concatenation of the vector $\boldsymbol{\rho}_S$ and the single parameter ρ_T have been denoted by $(\boldsymbol{\rho}_S, \rho_T)$.

The overall magnitude α of the GP prior represents the prior standard deviation of the latent GP function $\boldsymbol{f}$.

An exponentiated quadratic function assumes stationarity with respect to the input variables, and using a Kronecker product implies separability with respect to the spatial and temporal input dimensions.

5.7.2 Gaussian process with derivatives

The joint prior distribution for the latent function values $\boldsymbol{f}$ and partial derivative function values $\boldsymbol{f}'$, using a mean-square differentiable GP model, can be denoted as follows:

$$p(\boldsymbol{f}, \boldsymbol{f}'|X, X^*, \theta) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}' \end{bmatrix}\middle| 0, \begin{bmatrix} K_{f,f}(X,\theta) & K_{f,f'}(X, X^*, \theta) \\ K_{f',f}(X, X^*, \theta) & K_{f',f'}(X^*, \theta) \end{bmatrix}\right), \tag{5.5}$$

where $\boldsymbol{f}'$ denotes the values of the partial derivatives of latent function f with respect to some input dimension evaluated at inputs X^* associated with the derivative observations. The covariance matrix is extended to include the covariances between observations and partial derivatives ($K_{f,f'}$ and $K_{f',f}$) and the covariances between partial derivatives ($K_{f',f'}$), in a similar way as explained in Section 4.5.1 of Chapter 4.

The joint prior distribution in equation (5.5) can be combined with an observation model, $p(\boldsymbol{m}|\boldsymbol{f}')$, for the partial derivatives observations $\boldsymbol{m}$, to encode information about the partial derivatives of the function f into the model. In this work, we consider two types of derivative observations:

- observations of the value of a partial derivative of the function with respect to the time input variable and denoted by $m_{it} \in \mathbb{R}$ (m_{it} represents an artificial observation of $f'(\boldsymbol{x}_{it}) = \frac{\partial f(\boldsymbol{x}_{it})}{\partial t}$),
- and observations of the sign of a partial derivative of the function with respect to

the time input variable and denoted by $z_{it} \in \{1, -1\}$ (z_{it} represents an artificial observation of $\text{sign}(f'(\boldsymbol{x}_{it}))$), where $z_{it} = 1$ means that the partial derivative of the function is positive at the given data point and $z_{it} = -1$ means that the partial derivative is negative (decreasing function).

5.7.3 A zero-order constraint for time-series to start at zero

In order to force the time-series functions to start at zero, a zero-order constraint can be specified by using a set of virtual observations equal to zero,

$$\boldsymbol{y}_B = \{0 : (i,t) \in B\}, \tag{5.6}$$

at the subset $B = \{(i,t) : t = 1\}$ of starting points of every time-series. And the Dirac Delta function δ can be used as an observational model for these observations:

$$p(\boldsymbol{y}_B|\boldsymbol{f}_B) = \prod_{i,t\in B} \delta(y_{it} - f(\boldsymbol{x}_{it})),$$

where $\boldsymbol{f}_B$ denotes the GP latent function values $f(\boldsymbol{x}_{it})$ at the subset B of points. While the rest of observations, i.e. the observations $\boldsymbol{y}$ at the subset $A = \{(i,t) : t > 1\}$ of points and denoted by $\boldsymbol{y}_A$, are considered to be contaminated with Gaussian noise:

$$p(\boldsymbol{y}_A|\boldsymbol{f}_A, \sigma^2) = \prod_{i,t\in A} \mathcal{N}(y_{it}|f(\boldsymbol{x}_{it}), \sigma^2),$$

where $\boldsymbol{f}_A$ denotes the GP latent function values $f(\boldsymbol{x}_{it})$ at the subset A.

5.7.4 A first-order derivative constraint for time-series to stabilize in the long-term

In order to impose a saturation constraint for long-term stabilization of the time-series, a set of virtual observations m_{it} of the value of the partial derivative of the function with respect to the time input variable equal to zero can be considered:

$$\boldsymbol{m} = \{0 : (i,t) \in C\}, \tag{5.7}$$

where $C = \{(i,t) : t = 11\}$ is the subset of ending points of every time-series where to induce saturation of the function (as explained in Section 5.5). And the Dirac Delta function δ can be used as an observational model for these observations,

$$p(\boldsymbol{m}|\boldsymbol{f}'_C) = \prod_{i,t\in C} \delta(m_{it} - f'(\boldsymbol{x}_{it})),$$

where $f'(\boldsymbol{x}_{it}) = \frac{\partial f(\boldsymbol{x}_{it})}{\partial t}$ is the GP latent function partial derivative value at point (i,t), and $\boldsymbol{f}'_C$ denotes the partial derivative values $f'(\boldsymbol{x}_{it})$ at the subset C of points.

5.7.5 Monotonicity constraint for the time-series

The function is guaranteed to be non-decreasing as a function of time when the partial derivative is non-negative. This constraint can be specified by using a set of virtual observations of the sign of the partial derivative of the function with respect to the time input variable equal to one:

$$\boldsymbol{z} = \{1 : (i,t) \in E\}, \tag{5.8}$$

where $E = \{(i,t) : t = \{7, 10\}\}$ is the subset of desired points where to induce monotonicity (as explained in Section 5.5). The probit function $\Phi : \mathbb{R} \rightarrow (0,1)$ can be used as a likelihood for the signs of the partial derivatives,

$$p(\boldsymbol{z}|\boldsymbol{f}'_E) = \prod_{i,t \in E} \Phi\left(z_{it} \cdot f'(\boldsymbol{x}_{it}) \cdot \frac{1}{v}\right),$$

where $\boldsymbol{f}'_E$ denotes the partial derivative values $f'(\boldsymbol{x}_{it})$ at the subset E of points. The parameter $v > 0$ controls the strictness of the constraint, as seen in Section 4.7.2 of Chapter 4. In this work, we use $v = 10^{-4}$.

5.7.6 Likelihood function

The joint likelihood of regular observations $\boldsymbol{y} = (\boldsymbol{y}_A, \boldsymbol{y}_B)$, derivative value observations $\boldsymbol{m}$ and derivative sign observations $\boldsymbol{z}$, given latent functions $\boldsymbol{f} = (\boldsymbol{f}_A, \boldsymbol{f}_B)$ and $\boldsymbol{f}' = (\boldsymbol{f}'_E, \boldsymbol{f}'_C)$ and hyperparameter σ, results:

$$\begin{aligned} p(\boldsymbol{y},&\boldsymbol{m}, \boldsymbol{z}|\boldsymbol{f}, \boldsymbol{f}', \sigma) = \\ & p(\boldsymbol{y}_A|\boldsymbol{f}_A)\, p(\boldsymbol{y}_B|\boldsymbol{f}_B)\, p(\boldsymbol{z}|\boldsymbol{f}'_E)\, p(\boldsymbol{m}|\boldsymbol{f}'_C) = \\ & \Big(\prod_{i,t \in A} \mathcal{N}(y_{it}|f(\boldsymbol{x}_{it}), \sigma^2)\Big)\Big(\prod_{i,t \in B} \delta(y_{it} - f(\boldsymbol{x}_{it}))\Big) \\ & \times \Big(\prod_{i,t \in E} \Phi\big(z_{it} \cdot f'(\boldsymbol{x}_{it}) \cdot \frac{1}{v}\big)\Big)\Big(\prod_{i,t \in C} \delta(m_{it} - f'(\boldsymbol{x}_{it}))\Big). \end{aligned} \tag{5.9}$$

5.7.7 Posterior and predictive distributions

Bayesian inference is based on the posterior joint distribution of parameters given the data, which is proportional to the product of the likelihood and prior distributions,

$$p(\boldsymbol{f}, \boldsymbol{f}', \sigma | \boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z}) \propto p(\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z} | \boldsymbol{f}, \boldsymbol{f}', \sigma)\, p(\boldsymbol{f}, \boldsymbol{f}' | X, X^*, \boldsymbol{\rho}, \alpha)\, p(\boldsymbol{\rho})\, p(\alpha)\, p(\sigma),$$

where $p(\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z} | \boldsymbol{f}, \boldsymbol{f}', \sigma)$ is the likelihood of the model in equation (5.9) and $p(\boldsymbol{f}, \boldsymbol{f}' | X, X^*, \theta)$ is the joint Gaussian process prior of regular $\boldsymbol{f}$ and derivative $\boldsymbol{f}'$ latent functions values in equation (5.5). We set positive half-normal prior distributions (zero-mean normal distribution limited to the positive input domain $[0, \infty)$) for the hyperparameters α, $p(\alpha) = \mathcal{N}^+(\alpha | 0, 10)$, and σ, $p(\sigma) = \mathcal{N}^+(\sigma | 0, 10)$, and gamma distributions for the hyperparameters $\boldsymbol{\rho}$, $p(\rho_d) = \mathrm{Gamma}(\rho_d | 1, 0.1)$ for all d. These correspond to weakly informative prior distributions based on prior knowledge about the magnitude of the parameters. Thus, the joint posterior distribution can be expressed as depicted in equation (5.10).

$$\begin{aligned}
&p(\boldsymbol{f}, \boldsymbol{f}', \sigma | \boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z}) \propto \\
&\Big(\prod_{i,t \in A} \mathcal{N}(y_{it} | f(\boldsymbol{x}_{it}), \sigma^2) \Big) \Big(\prod_{i,t \in B} \delta(y_{it} - f(\boldsymbol{x}_{it})) \Big) \\
&\times \Big(\prod_{i,t \in E} \Phi\big(z_{it} \cdot f'(\boldsymbol{x}_{it}) \cdot \frac{1}{v}\big) \Big) \Big(\prod_{i,t \in C} \delta(m_{it} - f'(\boldsymbol{x}_{it})) \Big) \\
&\times \mathcal{N}\left(\begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{f}' \end{bmatrix} \middle| 0, \begin{bmatrix} K_{f,f}(X, \theta) & K_{f,f'}(X, X^*, \theta) \\ K_{f',f}(X, X^*, \theta) & K_{f',f'}(X^*, \theta)) \end{bmatrix} \right) \\
&\times \mathcal{N}^+(\alpha | 0, 1) \Big(\prod_{d=1}^{D} \mathrm{Gamma}(\rho_d | 1, 0.1) \Big) \mathcal{N}^+(\sigma | 0, 1). \qquad (5.10)
\end{aligned}$$

Predictive inference for new function values $\tilde{\boldsymbol{y}}$ for a new sequence of input values $\tilde{X}$ can be computed by integrating over the joint posterior distribution,

$$\begin{aligned}
&p(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{m}}, \tilde{\boldsymbol{z}} | \boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z}) = \\
&\quad \int p(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{m}}, \tilde{\boldsymbol{z}} | \tilde{\boldsymbol{f}}, \tilde{\boldsymbol{f}}', \sigma)\, p(\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{f}}' | \boldsymbol{f}, \boldsymbol{f}')\, p(\boldsymbol{f}, \boldsymbol{f}', \sigma | \boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z})\, \mathrm{d}\boldsymbol{f}\, \mathrm{d}\boldsymbol{f}'\, \mathrm{d}\sigma.
\end{aligned}$$

The main computational demands of this model comes from the covariance matrix inversion operation of optimizing the hyperparameters. In a spatio-temporal GP model this is a $O\big((NT)^3\big)$ computational demanding operation.

5.8 Spatially correlated time-series model with derivative information

5.8.1 Penalized thin-plate splines-based time-series model

We consider a penalized thin-plate cubic-splines function, as it was derived in equation (4.5) (Section 4.5.2 on Chapter 4), for modeling the latent function of each time-series of the spatio-temporal data. Thus, the ith time-series latent function $f_i : \mathbb{R} \to \mathbb{R}$, as a function of the time input variable, takes the form:

$$f_i(t) = \boldsymbol{H}_{t\cdot}\boldsymbol{\beta}_{\cdot i} + \boldsymbol{Z}_{t\cdot}\Omega^{-\frac{1}{2}}\boldsymbol{b}_{\cdot i}. \tag{5.11}$$

where the tth element of the ith time-series, f_{it}, is the value of the function f_i evaluated at the t time point variable, i.e. $f_i = f_i(t)$.

Notice that the following equivalences between formulation used in Section 4.5.2 and the one used in the actual section should be done:

- Index i in Section 4.5.2 is index t in the current section.

- Input value x_i in Section 4.5.2 is input value t in the current section.

Let $\boldsymbol{f}_i = (f_1(t), \ldots, f_N(t))$ the vector of latent functional values of the ith time-series, which takes the form:

$$\boldsymbol{f}_i = \left(H\boldsymbol{\beta}_{\cdot i} + Z\Omega^{-\frac{1}{2}}\boldsymbol{b}_{\cdot i}\right)^\top.$$

And, let $F = [\boldsymbol{f}_1 \ \boldsymbol{f}_2 \ \cdots \ \boldsymbol{f}_N]^\top \in \mathbb{R}^{N\times T}$ the matrix of spatio-temporal latent function values where $\boldsymbol{f}_i$ is the row-vector containing the ith time-series latent values. F can be expressed as:

$$F = \left(H\boldsymbol{\beta} + Z\Omega^{-\frac{1}{2}}\boldsymbol{b}\right)^\top,$$

where $\boldsymbol{\beta} \in \mathbb{R}^{2\times N}$ is the matrix of linear coefficients with ith column $\boldsymbol{\beta}_{\cdot i} = (\beta_{1i}, \beta_{2i})^\top$ of linear coefficients for the ith time-series, $\boldsymbol{b} = [\boldsymbol{b}_{\cdot 1}\, \boldsymbol{b}_{\cdot 2} \cdots \boldsymbol{b}_{\cdot N}] \in \mathbb{R}^{K\times N}$ is the matrix of splines coefficients with ith column $\boldsymbol{b}_{\cdot i} = (b_{1i}, \ldots, b_{Ki})^\top \in \mathbb{R}^K$ of penalized splines coefficients for the ith time-series, $H \in \mathbb{R}^{T\times 2}$ is the matrix of linear function values with tth row $\boldsymbol{H}_{t\cdot} = (1, t)$ of linear function values of the tth time point, $Z \in \mathbb{R}^{T\times K}$ is the matrix of cubic-spline function values with tth row $\boldsymbol{Z}_{t\cdot} = (Z_{t1}, \ldots, Z_{tK}) \in \mathbb{R}^K$ of cubic-splines function values of the tth time point following the equation (4.4) on Chapter 4, and Ω is the matrix of penalization as derived in Section 4.5.2 of Chapter 4.

5.8.2 Spatially correlating the splines-based time-series functions

In order to establish a correlation structure among time-series i, the matrix of spline coefficients $\boldsymbol{b} \in \mathbb{R}^{K \times N}$ is considered as a realization of a continuous stochastic process, and a GP prior model with a two-dimensional $\big($knots (K) and space $(N)\big)$ covariance function can be defined on them.

With the aim of simplifying the covariance structure, null covariances between splines coefficients belonging to a different order of splines can be considered, which is equivalent to define K independent Gaussian process priors, one for each kth row $\boldsymbol{b}_{k\cdot} = (b_{k1}, \ldots, b_{kN}) \in \mathbb{R}^N$ of matrix $\boldsymbol{b}$, such that

$$p(\boldsymbol{b}_{k\cdot}|X_S, \boldsymbol{\rho}_k) = \mathcal{N}\big(\boldsymbol{b}_{k\cdot}|0, \alpha_k C_k(X_S; \boldsymbol{\rho}_k)\big),$$

for $k = 1, \ldots, K$. C_k is the covariance function for the vector of coefficients $\boldsymbol{b}_{k\cdot}$. Hyperparameter α_k is the standard deviation of the Gaussian process which controls the overall scale or magnitude of the range of values of the vector $\boldsymbol{b}_{k\cdot}$. Thus, specific covariance structure for each kth vector of splines coefficients $\boldsymbol{b}_{k\cdot}$ can be specified.

Furthermore, in this work, we will simplify even more the covariance structure considering the same spatial structure for every vector of splines coefficients $\boldsymbol{b}_{k\cdot}$, i.e.,

$$p(\boldsymbol{b}_{k\cdot}|X_S, \boldsymbol{\rho}) = \mathcal{N}\big(\boldsymbol{b}_{k\cdot}|0, \alpha_k\, C(X_S; \boldsymbol{\rho})\big),$$

for $k = 1, \ldots, K$, and C is a common covariance function for every vector of coefficients $\boldsymbol{b}_{k\cdot}$.

The $N \times N$ covariance matrix C is computed by a squared exponential covariance function (Section 3.4 in Chapter 3), dependent on the vectors of input values $\boldsymbol{x}_{S_i}$ in equation (5.1) for every spatial location i, and the vector of lengthscale parameters $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_5)$,

$$C(X_S; \boldsymbol{\rho})_{(i,j)} = \exp\left(-\frac{1}{2}\sum_{d=1}^{5}\frac{1}{\rho_d^2}(\boldsymbol{x}_{S_{i,d}} - \boldsymbol{x}_{S_{j,d}})^2\right),$$

where $i, j = 1, \cdots, N$. The lengthscale parameters ρ_1, ρ_2 and ρ_3 correspond to the input variables H, S and I, respectively. The spatial coordinates input variables S_x and S_y are sharing the lengthscale parameter ($\rho_4 = \rho_5$), such that the covariance function depends on the (Euclidean) distance between spatial coordinates. Hyperparameter ρ_d controls the smoothness of the function for $\boldsymbol{b}_{k\cdot}$ in the direction of the dth predictor (input variable). The squared exponential covariance function is a stationary function with respect to the input variables.

5.8.3 Derivative of the penalized splines-based time-series model

The derivative function $f_i' : \mathbb{R} \rightarrow \mathbb{R}$ of the penalized cubic-splines function f_i in equation (5.11) with respect to the time input variable takes the form:

$$f_i'(t) = \frac{\partial f_i(t)}{\partial t} = \boldsymbol{\beta}_{2i} + \frac{\partial \boldsymbol{Z}_{t\cdot}}{\partial t} \Omega^{-\frac{1}{2}} \boldsymbol{b}_{\cdot i}, \tag{5.12}$$

where $\frac{\partial \boldsymbol{Z}_{t\cdot}}{\partial t} = (\frac{\partial Z_{t1}}{\partial t}, \ldots, \frac{\partial Z_{tK}}{\partial t}) \in \mathbb{R}^K$, is the row-vector of the derivative cubic-splines values at time t, where the elements $\frac{\partial \boldsymbol{Z}_{tk}}{\partial t}$ follow the partial derivative function derived in equation (4.7) (Section 4.5.2 of Chapter 4) with respect to the time input variable. Notice that in the terms $\frac{\partial \boldsymbol{Z}_{tk}}{\partial t}$ we use the same letter t to denote two different things, the index for the function $\boldsymbol{Z}_{tk}$ and the time input variable t with respect to which we derive the function $\boldsymbol{Z}_{tk}$.

5.8.4 A zero-order constraint for time-series to start at zero

Similarly to previous Section 5.7.3, in order to constraint the time-series to be zero at the set of starting points $B = \{(i,t) : t = 1\}$, a Dirac Delta function δ can be used as an observational model for the set $\boldsymbol{y}_B$ (in equation (5.6)) of virtual observations equal to zero:

$$p(\boldsymbol{y}_B | \boldsymbol{f}_B) = \prod_{i,t \in B} \delta(y_{it} - f_i(t)).$$

While the rest of the observations $\boldsymbol{y}_A$ (in Section 5.7.3) are considered to be contaminated with Gaussian noise:

$$p(\boldsymbol{y}_A | \boldsymbol{f}_A, \sigma^2) = \prod_{i,t \in A} \mathcal{N}(y_{it} | f_i(t), \sigma^2),$$

where $\boldsymbol{f}_B$ and $\boldsymbol{f}_A$ denote the function values $f_i(t)$ at the subset B and A (in Section 5.7.3) of points, respectively.

5.8.5 A first-order derivative constraint for time-series to stabilize in the long-term

Similarly to previous Section 5.7.4, in order to impose a saturation constraint for long-term stabilization of the time-series, i.e. at the subset $C = \{(i,t) : t = T\}$ of ending points of every time-series, a Dirac Delta function δ can be used as an observational model for the set $\boldsymbol{m}$ (in equation (5.7)) of virtual observations of the

time-partial derivative of the function equal to zero:

$$p(\boldsymbol{m}|\boldsymbol{f}'_C) = \prod_{i,t \in C} \delta(m_{it} - f'_i(t)),$$

with $\boldsymbol{f}'_C$ denotes the partial derivative function values $f'_i(t)$ at the subset C of points.

5.8.6 Monotonicity constraint for the time-series

Similarly to previous Section 5.7.5, in order to guarantee that time-series are non-decreasing as a function of time, a probit function Φ can be used as the observational model for the set $\boldsymbol{z}$ (in equation (5.8)) of virtual observations of the sign of the time-partial derivatives of the function:

$$p(\boldsymbol{z}|\boldsymbol{f}'_E) = \prod_{i,t \in E} \Phi\left(z_{it} \cdot f'_i(t) \cdot \frac{1}{v}\right), \tag{5.13}$$

where $E = \{(i,t) : t = \{7, 10\}\}$ is the subset of desired time points where to induce monotonicity, and $\boldsymbol{f}'_E$ denotes the derivative function values $f'_i(t)$ at subset E. As before, in this work, we use $v = 10^{-4}$.

5.8.7 Likelihood function

The joint likelihood of regular observations $\boldsymbol{y} = (\boldsymbol{y}_A, \boldsymbol{y}_B)$, derivative value observations $\boldsymbol{m}$ and derivative sign observations $\boldsymbol{z}$, given the parameters $\boldsymbol{\beta}$, $\boldsymbol{b}$ and σ results:

$$\begin{aligned}
&p(\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z}|\boldsymbol{\beta}, \boldsymbol{b}, \sigma) = \\
&\quad p(\boldsymbol{y}_A|\boldsymbol{f}_A)\, p(\boldsymbol{y}_B|\boldsymbol{f}_B)\, p(\boldsymbol{z}|\boldsymbol{f}'_E)\, p(\boldsymbol{m}|\boldsymbol{f}'_C) = \\
&\quad \Big(\prod_{i,t \in A} \mathcal{N}(y_{it}|f_i(t), \sigma^2)\Big)\Big(\prod_{i,t \in B} \delta\big(y_{it} - f_i(t)\big)\Big) \\
&\quad \times \Big(\prod_{i,t \in E} \Phi\big(z_{it} \cdot f'_i(t) \cdot \frac{1}{v}\big)\Big)\Big(\prod_{i,t \in C} \delta\big(m_{it} - f'_i(t)\big)\Big).
\end{aligned}$$

5.8.8 Posterior and predictive distributions

The posterior joint distribution of parameters given the data, which is proportional to the product of the likelihood and priors, is:

$$p(\boldsymbol{\beta}, \boldsymbol{b}, \sigma|\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z}) \propto p(\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z}|\boldsymbol{\beta}, \boldsymbol{b}, \sigma)\, p(\boldsymbol{\beta})\, p(\boldsymbol{b}|\alpha, \boldsymbol{\rho})\, p(\alpha)\, p(\boldsymbol{\rho})\, p(\sigma),$$

where $p(\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z}|\boldsymbol{\beta}, \boldsymbol{b}, \sigma)$ is the likelihood of the model and $p(\boldsymbol{b}|\alpha, \boldsymbol{\rho})$ is the GP prior for the splines coefficients $\boldsymbol{b}$. We set normal prior distributions for the linear coefficients $\boldsymbol{\beta}$, $p(\boldsymbol{\beta}) = \mathcal{N}(\alpha|0, 10I)$, where I denotes the identity matrix, and positive half-normal prior distributions for the hyperparameters α, $p(\alpha) = \mathcal{N}^+(\alpha|0, 10)$, and σ, $p(\sigma) = \mathcal{N}^+(\sigma|0, 10)$, and gamma distributions for the hyperparameters $\boldsymbol{\rho}$, $p(\rho_d) = \mathrm{Gamma}(\rho_d|1, 0.1)$ for all d. These correspond to weakly informative prior distributions based on prior knowledge about the magnitude of the parameters.

The joint predictive distribution of new output values $\tilde{\boldsymbol{y}}$ and $\tilde{\boldsymbol{y}}'$ for a new sequence of input values X^* can be computed by integrating out over the posterior distribution

$$p(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{m}}, \tilde{\boldsymbol{z}}|\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z}) = \int p(\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{m}}, \tilde{\boldsymbol{z}}|\boldsymbol{\beta}, \boldsymbol{b}, \sigma)\, p(\boldsymbol{\beta}, \boldsymbol{b}, \sigma|\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z})\, \mathrm{d}\boldsymbol{\beta}\, \mathrm{d}\boldsymbol{b}\, \mathrm{d}\sigma. \tag{5.14}$$

The main computational demands of this correlated splines model come from the covariance matrix inversion operation of optimizing the hyperparameters in the GP prior for the splines coefficients. The proposed model considers the same spatial covariance structure between splines coefficients belonging to different a order of the splines functions. This requires $O(N^3)$ computation expense in covariance matrix inversion, where N denotes the number of spatial locations.

5.9 Bayesian inference

To posterior distribution of interest $p(\boldsymbol{f}, \sigma|\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z})$ and $p(\boldsymbol{\beta}, \boldsymbol{b}, \sigma|\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z})$ are in general intractable. Hence, to estimate both parameter posterior distribution and posterior predictive distribution for this model, simulation methods and/or distributional approximations methods [Gelman et al., 2013] must be used. Simulating methods based on Markov chain Monte Carlo (MCMC) [Brooks et al., 2011] and, more recently, on Hamiltonian Monte Carlo (HMC) [Neal et al., 2011] are general sampling methods to obtain samples from the joint posterior distribution. In this study, HMC methods are used to make inference over the posterior and predictive distributions. Using sampling methods, such as HMC, the covariance matrix must be inverted in every step of the sampler. For large data sets where iterative simulation algorithms are too slow, modal and distributional approximation methods can be efficient and approximate alternatives [Gelman et al., 2013].

5.10 Model checking, predictive performance and model selection

The *posterior predictive checks*, which are also known as the *leave-one-out probability integral transformation* (LOO-PIT), can be used as a rigorous procedure in order to guarantee good model performance and ensure that the model is compatible with the observed data. They are based on computing the probability of new predictions to be lower or equal to their corresponding actual observations following a leave-one-out cross-validation procedure [Gelfand et al., 1992, Gelman et al., 2013]:

$$\text{LOO-PIT}_{(i,t)\in\mathfrak{D}} = P(\tilde{y}_{(i,t)\in\mathfrak{D}} \leq y_{(i,t)\in\mathfrak{D}}),$$

where $\mathfrak{D}$ is the subset of observation indices of new predictions in the cross-validation. $\tilde{y}_{(i,t)\in\mathfrak{D}}$ are the new observations from the predictive distribution at the subset $\mathfrak{D}$, and $y_{(i,t)\in\mathfrak{D}}$ are the actual observations at this subset $\mathfrak{D}$. The similarity or provenance of these probabilities from standard uniform distributions endorses these probabilities with the desirable property of having the same interpretation across models, which implies good fit to the data and good prediction [Bayarri and Berger, 2000]. Using simulation methods for estimating and predicting a Bayesian model, computing the probability of a predicted value being minor the observed one is straightforward through the collection of simulated values.

The *expected log predictive density* (ELPD) evaluates, by averaging over all the steps in the leave-one-out cross-validation procedure, how far new data is from the model while taking the posterior uncertainties into account. It is based on the log-density of an observation that does not take part in fitting the model, given the model [Vehtari et al., 2012]:

$$\text{ELPD} = \frac{1}{|\mathfrak{D}|} \sum_{(i,t)\in\mathfrak{D}} \ln(p(y_{it}|\boldsymbol{y}_{-\mathfrak{D}})),$$

where $|\mathfrak{D}|$ denotes the cardinality of the subset $\mathfrak{D}$ of observation indices in the cross-validation. $\boldsymbol{y}_{-\mathfrak{D}}$ denotes the dataset without the subset $\mathfrak{D}$ of observations, $\boldsymbol{y}_{-\mathfrak{D}} = \{y_{it} : (i,t) \notin \mathfrak{D}\}$.

The *mean square predictive error* (MSE) also evaluates, by averaging over all the steps in the leave-one-out cross-validation procedure, how far new data is from the model by using the distance (error) between the actual observation (y_{it}), that does not take part in fitting the model, and the predictive mean ($\tilde{y}_{it}$) from the model:

$$\text{MSE} = \frac{1}{|\mathfrak{D}|} \sum_{(i,t)\in\mathfrak{D}} (y_{it} - \tilde{y}_{it})^2.$$

Following a leave one observation out cross-validation scheme (denoted as *LOO-CV*), the subset $\mathfrak{D}$ of observation indices will be just a single observation (i,t), $\mathfrak{D} = \{(i,t)\}$. The LOO-PIT, ELPD, and MSE will be computed following the *LOO-CV* scheme. The LOO-PIT will essentially be useful for model checking, ensuring that the model is compatible with the data. The ELPD and MSE will evaluate the predictive performance of individual observations (i,t).

The end goal of this work is to predict complete color-fading time-series at new unobserved locations. In order to do that, a leave one location out cross-validation scheme (denoted as *LOLO-CV*) can be performed, where the subset of observation indices $\mathfrak{D}$ will be a complete time-series of a specific spatial location i, $\mathfrak{D} = \{(i,t) : t \in \{1,\ldots,T\}\}$. The statistics ELPD and MSE will be computed following the *LOLO-CV* scheme. Plots of predicted new time-series superimposed to their corresponding actual observations will be shown in order to visually evaluate the predictive performance. Model selection can be done by comparing the predictive performance between models using the ELPD and MSE statistics. The best model is who maximizes the ELPD and/or minimizes the MSE.

5.11 Experimental results

In this section, we present the results of fitting both of the models proposed, GPs and correlated splines, to the observed data and conducting the cross-validation procedures to asses the predictive performance of the models.

The posterior distributions and predictive distributions have been estimated by HMC sampling methods [Neal et al., 2011] using the Stan software [Carpenter et al., 2017]. Three simulation chains with different initial values have been launched. The convergence of the simulation chains was evaluated by the split-Rhat convergence diagnosis [Gelman et al., 2013] and the effective sample size of the chains. A value of 1 in the split-Rhat convergence statistic indicates convergence of the simulation chain, although conventionally accepted values of convergence would be between 1 and 1.1. In our case and for both models, a split-Rhat value lower than 1.05 has been obtained for all parameter simulation chains.

As commented in Section 5.5, the input variables H (Hue), S (Saturation) and I (Intensity) were previously re-scaled by dividing by their standard deviations. The S_x and S_y spatial coordinates were jointly re-scaled by dividing by their common standard deviation. The time input variable t was not re-scaled.

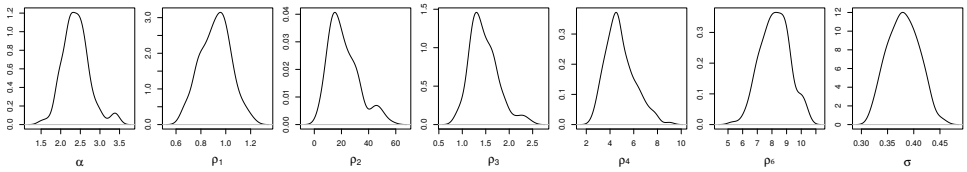


Figure 5.2: Posterior marginal density distributions for the hyperparameters: marginal variance α and lengthscales $\boldsymbol{\rho} = \{\rho_1, \rho_2, \rho_3, \rho_4, \rho_6\}$ of the GP prior, and residual noise σ of the model.

Table 5.2: Estimated posterior modes of the hyperparameters.

Input variables						Magnitud GP	Obs. noise
H	S	I	S_x	S_y	time		
ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	α	σ
0.95	18.3	1.3	4.5		8.2	2.4	0.37

5.11.1 Results of the spatio-temporal Gaussian process modeling approach

The estimated posterior marginal distributions of the hyperparameters: lengthscales $\boldsymbol{\rho}$ associated with the input variables, overall magnitude α of the latent GP function and the observation noise σ of the model, can be visualized in Figure 5.2, and their estimated modes are shown in Table 5.2.

The posterior means of the process $p(\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z})$ versus the input variables H, S and I, and the time points, can be visualized in Figure 5.13 in Appendix 5.A.

In Figure 5.3, the predictive distributions (predictive means and 95% pointwise credible intervals) of the regular process, $p(\tilde{\boldsymbol{y}}|\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z})$, evaluated at the sampling input space, are plotted as a function of time for specific spatial locations. Additionally, the predictive means without derivative information, $p(\tilde{\boldsymbol{y}}|\boldsymbol{y})$, are also plotted for comparison.

The predictive means of the derivative latent process, $p(\tilde{\boldsymbol{f}}'|\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z})$, are plotted as a function of time for specific spatial locations in Figure 5.4.

Figure 5.5 shows predictive distributions $p(\tilde{\boldsymbol{y}}_{i\cdot}|\boldsymbol{y}_{-i\cdot}, \boldsymbol{m}_{-i\cdot}, \boldsymbol{z}_{-i\cdot})$ of new time-series following the cross-validation scheme *LOLO-CV*, which is based on leaving the whole time-series observartions of the location i out of the training dataset. Predictive means and pointwise credible intervals for both the model with derivative information and the model without derivative information are plotted for comparison. The actual data of the time-series at predicted locations are also plotted to visually evaluate the predictions.

Table 5.3 shows the ELPD and MSE statistics computed by following the two

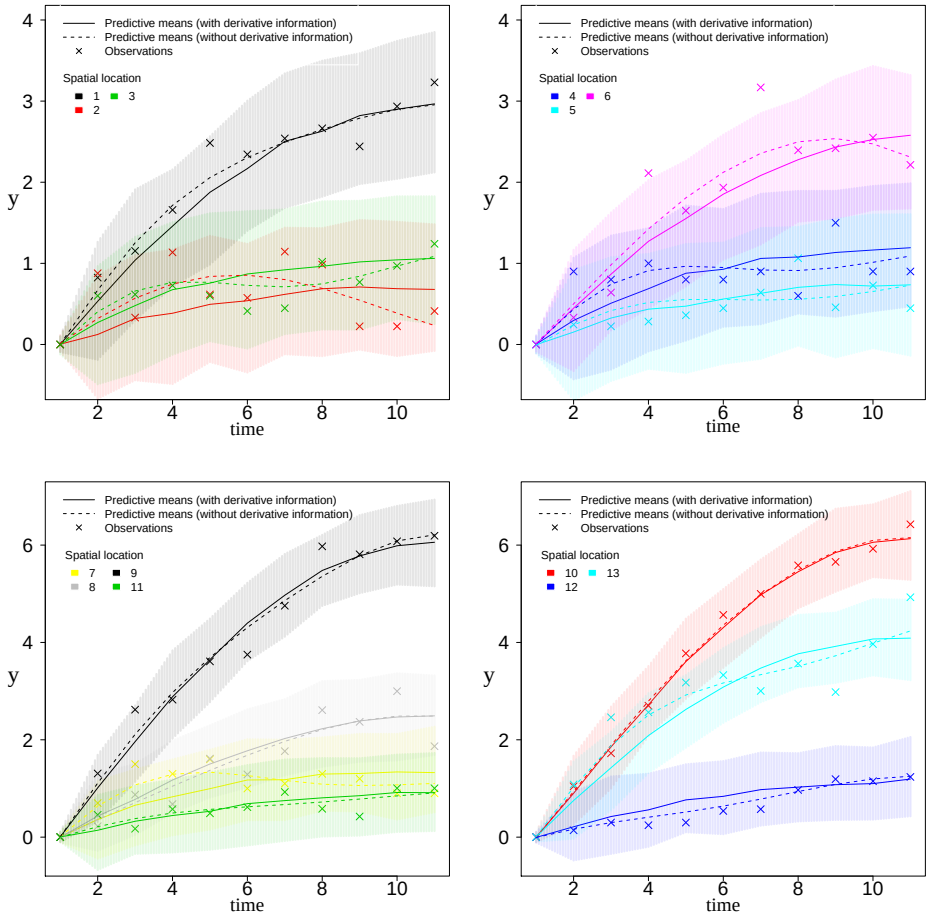


Figure 5.3: Predictive means and 95% pointwise credible intervals of the regular process at the sampling input space, with and without derivative information, plotted as a function of time and for specific spatial locations. The actual data $\boldsymbol{y}$ are also plotted as crosses.

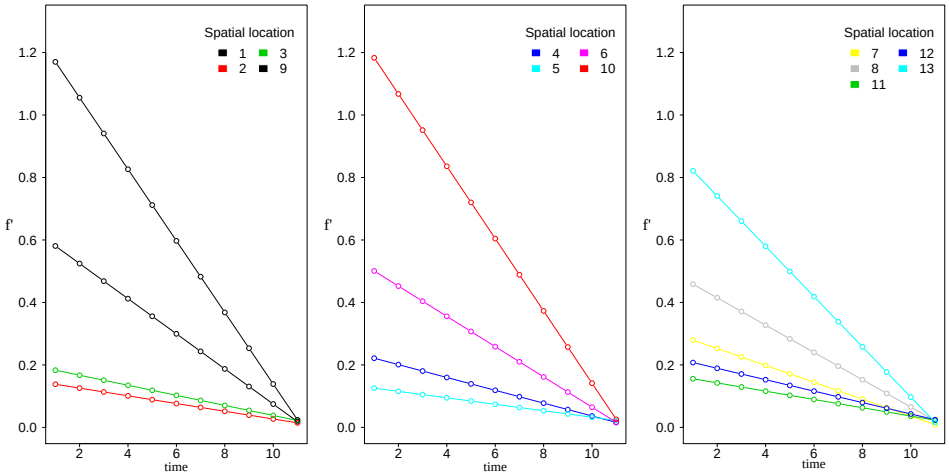


Figure 5.4: Posterior predictive means of the derivative latent functions plotted as a function of time for specific locations.

Table 5.3: The ELPD and MSE for the model with and without derivative information, computed by the two cross-validation scenarios, *LOO-CV* and *LOLO-CV*.

		with derivative information	without derivative information
LOO-CV	ELPD	-0.61	-0.78
	MSE	0.14	0.15
LOLO-CV	ELPD	-16.39	-29.61
	MSE	4.12	4.20

different cross-validation scenarios, the *LOO-CV* and *LOLO-CV* (as explained in Section 5.10), and for the model with and without derivatives.

Figure 5.6 shows the frequency histograms of the LOO probability integral transformation (LOO-PIT) by following the *LOO-CV*, for both models, with and without derivatives.

Additional information of the estimated joint covariance matrix K of the process is provided in Appendix 5.B. Images of the spatio-temporal covariance structure of both the regular and derivatives observations, and for both training and predicting data points, are depicted in Appendix 5.B.

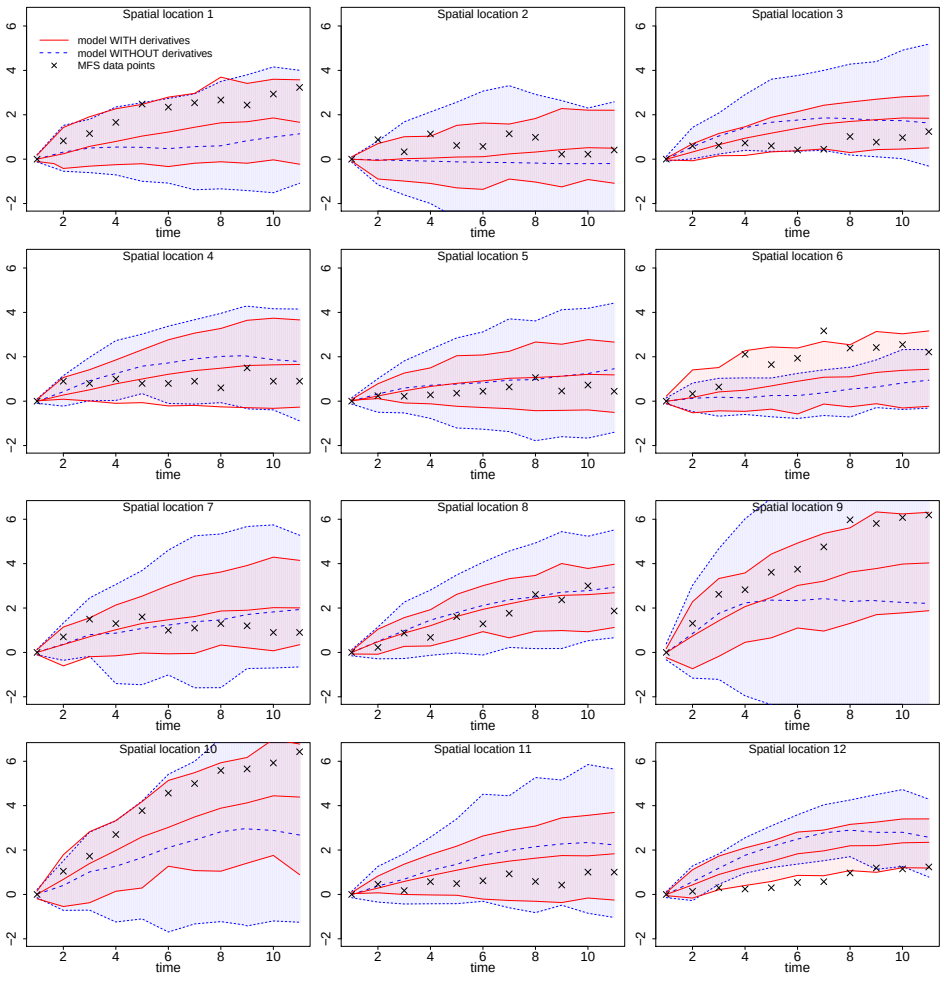


Figure 5.5: Predictive means and 95% pointwise credible intervals of new time-series at predicted locations in the leave-one location-out cross-validation procedure (*LOLO-CV*), using both models with and without derivative information. The actual MFS data time-series for every spatial location are plotted as crosses.

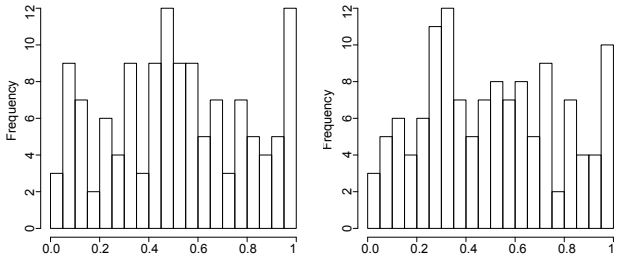


Figure 5.6: Frequency histograms of the LOO probability integral transformation (LOO-PIT) by following the *LOO-CV*, for both models, with (left) and without (right) derivative information.

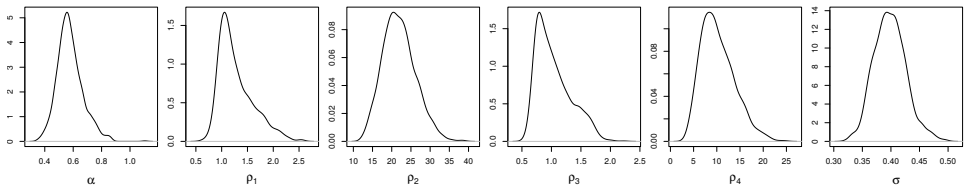


Figure 5.7: Posterior marginal density distributions of the hyperparameters: marginal variance α and lengthscales $\boldsymbol{\rho} = (\rho_1, \rho_2, \rho_3, \rho_4)$ of the GP prior for the splines coefficients, and residual noise σ of the model.

5.11.2 Results of the spatially correlated time-series modeling approach

The estimated posterior marginal distributions of the hyperparameters: lengthscales $\boldsymbol{\rho}$ associated to the input variables, overall magnitude α of the GP prior for the splines coefficients and the observation noise σ of the model, can be visualized in Figure 5.7, and their estimated modes are shown in Table 5.4.

Figure 5.8 shows the means and 95% pointwise credible intervals of predictive distributions, $p(\tilde{\boldsymbol{y}}|\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z})$, evaluated over the sampling input space and plotted as a function of time for specific spatial locations and superimposed to the actual observations $\boldsymbol{y}$. Additionally, the means of the predictive distribution of the model

Table 5.4: Estimated posterior modes of the hyperparameters.

Lengthscales associated to input variables					Magnitud GP prior of splines coefficients $\boldsymbol{b}$	Obs. noise
H	S	I	S_x	S_y		
ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	α	σ
1.10	20.0	0.85	7.9		0.58	0.38

Table 5.5: ELPD and MSE for both models, with and without derivative information, computed by the two cross-validation schemes, *LOO-CV* and *LOLO-CV*.

		with derivative information	without derivative information
LOO-CV	ELPD	-0.61	-0.78
	MSE	0.13	0.14
LOLO-CV	ELPD	-16.40	-33.39
	MSE	4.10	4.42

without the inclusion of the derivatives, $p(\tilde{\boldsymbol{y}}|\boldsymbol{y})$, are also plotted for comparison.

Table 5.5 shows the ELPD and MSE statistics computed by following the two different cross-validation schemes, *LOO-CV* and *LOLO-CV*, for both splines models, with and without derivatives.

In Figure 5.9, the predictive means of the derivative latent process $p(\tilde{\boldsymbol{f}}'|\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z})$ are plotted as a function of time for specific spatial locations.

Figure 5.10, shows the frequency histograms of the posterior predictive checks (LOO-PIT) by following the cross-validation scheme *LOO-CV*, for both splines models, with and without derivatives.

Figure 5.11 shows predictive distributions (predictive means and 95% pointwise credible intervals) of the proposed model with derivatives, $p(\tilde{\boldsymbol{y}}_i.|\boldsymbol{y}_{-i}., \boldsymbol{m}_{-i}., \boldsymbol{z}_{-i}.)$, and the model without derivatives, $p(\tilde{\boldsymbol{y}}_i.|\boldsymbol{y}_{-i}.)$, for new time-series by following the cross-validation scheme *LOLO-CV*. The actual data of the time-series $\boldsymbol{y}$ are also plotted to visually evaluate the predictions.

5.12 Discussion

The input variables were re-scaled in the same way in both modeling approaches, i.e. the GP model and the correlated splines model, so the lengthscales estimates in both models are comparable. The lengthscale parameters ρ_1 and ρ_3, corresponding to the variables H and I, respectively, are relatively small, in both modeling approaches, as can be seen in Table 5.2 for the GP model and in Table 5.4 for the splines model. This indicates that the posterior functions are non-linear with those variables or that the rate of decay of the correlation is moderately high. Therefore, variations in the input variables H and I imply a moderately quick decrease in the correlations allowing for the non-linear effects.

The variables S, S_x, and S_y have larger lengthscales, in both modeling approaches, as can be seen in Table 5.2 for the GP model and in Table 5.4 for the splines model. Notice that variables S_x and S_y share the same lengthscale ($\rho_4 = \rho_5$), such that the covariance function depends on the Euclidean distance between spatial

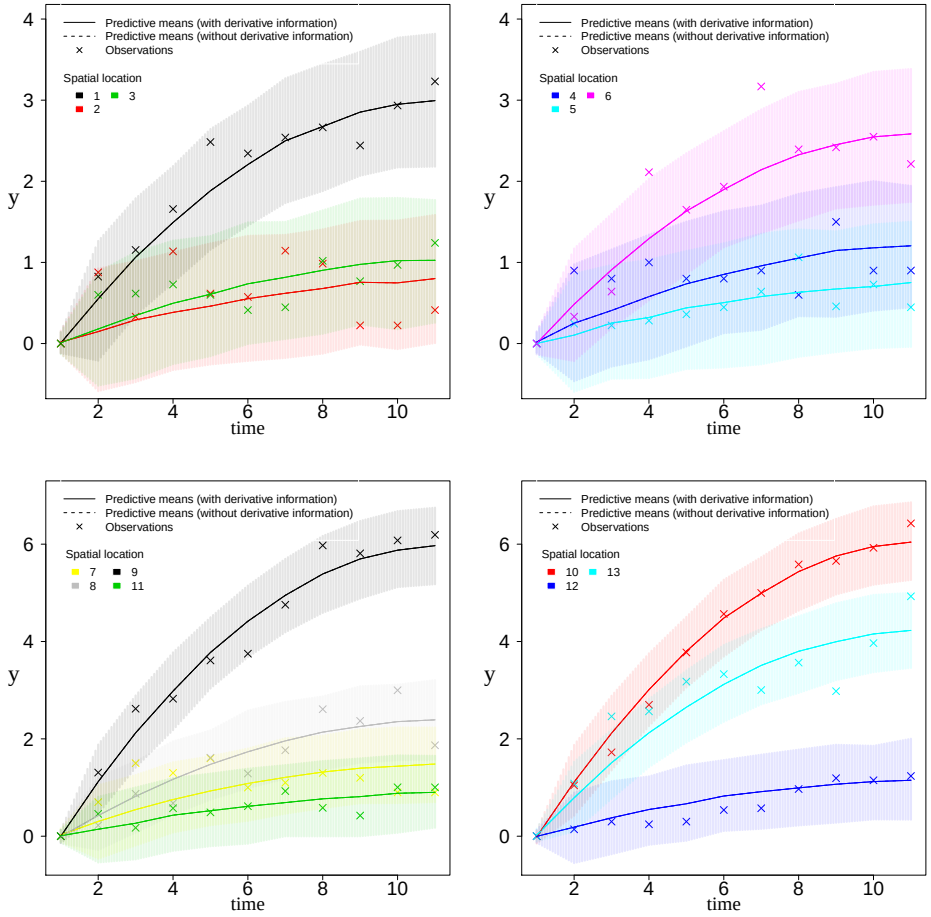


Figure 5.8: Predictive distributions (means and 95% pointwise credible intervals) of the regular process with and without derivative information, plotted as a function of time and for specific spatial locations. The actual data $\boldsymbol{y}$ are also plotted as crosses.

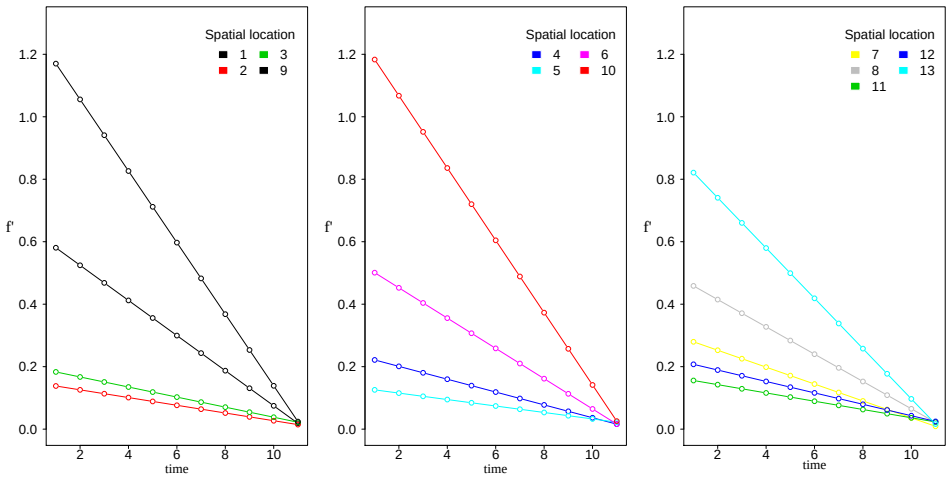


Figure 5.9: Posterior derivatives means of the derivative latent function plotted as a function of time for specific locations.

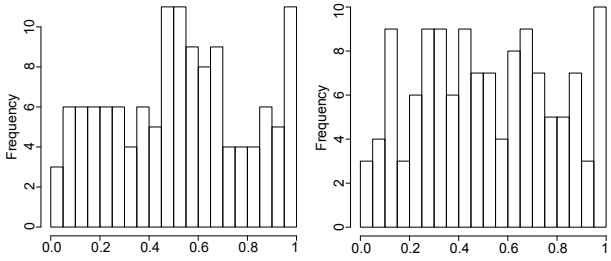


Figure 5.10: Histograms of the LOO probability integral transformation (LOO-PIT) by following *LOO-CV*, for both the model with derivatives (left) and the model without derivatives (right).

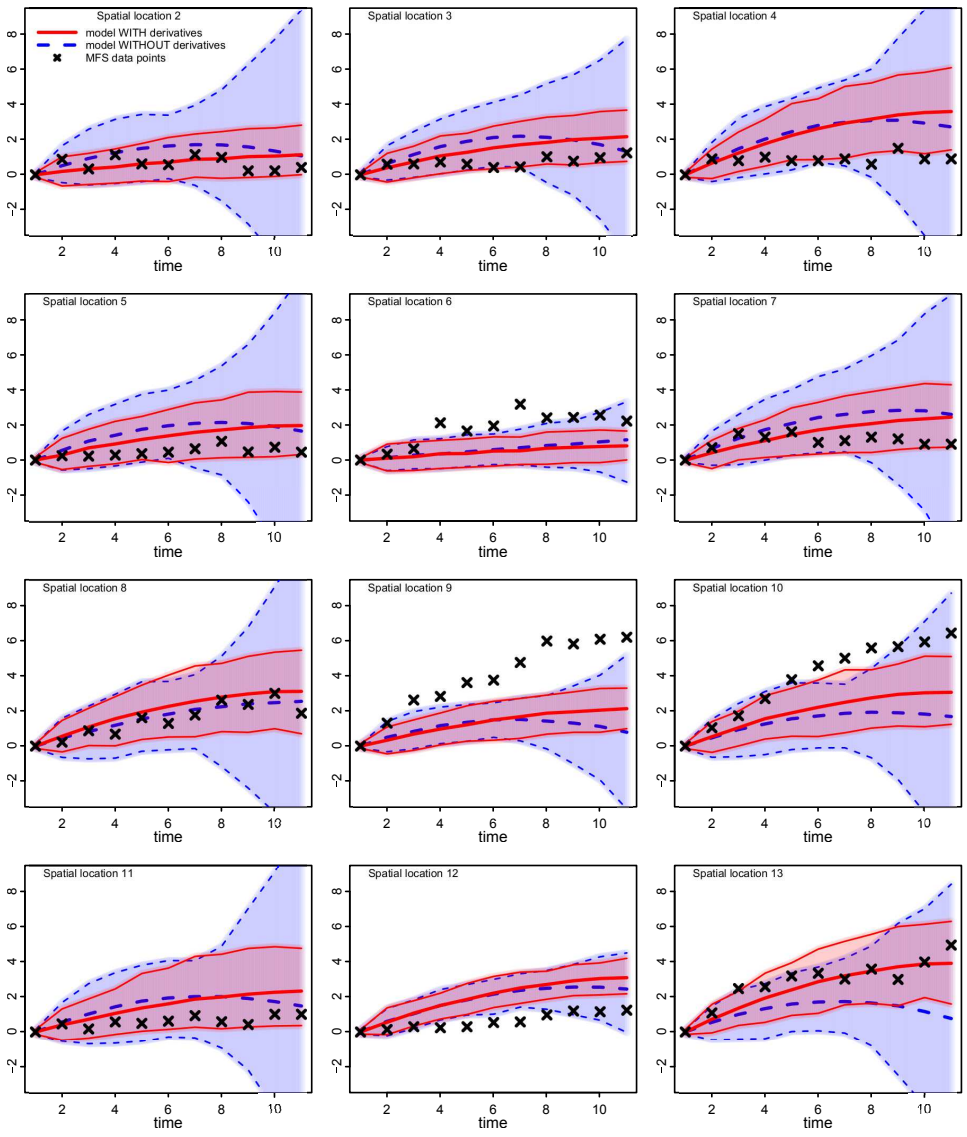


Figure 5.11: Predicted means and 95% pointwise credible intervals of new time-series at predicted locations in the cross-validation scheme *LOLO-CV*, for both splines models, with and without derivatives. The actual data points $\boldsymbol{y}$ are also plotted as crosses.

coordinates. This indicates that the function depends on S and Euclidean spatial distance in a smoother and less non-linear way. Especially, the variable S with a very long lengthscale contributes with a constant effect of one to the correlation and implies a constant function to the Gaussian process, thus being an irrelevant variable to the process.

In the case of the GP modeling approach, separability between the spatial and temporal input dimensions can be clearly appreciated in Figure 5.13 in Appendix 5.A, where the values of the process $\boldsymbol{f}$ versus the input variables (H, S and I) on the spatial dimension are independent on the time points, only differing in their overall scale.

The Frequency histograms of the LOO probability integral transformation (LOO-PIT) for both modeling approaches, the GP model in Figure 5.6 and the splines model in Figure 5.10, with and without derivatives, show similarities to uniform distributions. Which means good model performances and that the models are compatible with the observed data.

The use of an observation model with the absence of noise at the starting time points, $\{y_{it} = 0 : t = 1\}$, forces the predictive distributions $p(\tilde{\boldsymbol{y}}|\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z})$ to be zero at those starting points, in both modeling approaches, as it can be seen in Figures 5.3 and 5.5 for the GP model and Figures 5.8 and 5.11 for the splines model.

Inducing monotonicity by means of virtual observations of the sign of the partial derivatives in the data points $\{\text{sign}(\frac{\partial f_{it}}{\partial t}) = 1 : t \in \{7, 10\}\}$ has been sufficient to achieve monotonicity throughout of the time-series in both modeling approaches, thus preventing an overly smoothing effect on the posterior functions due to using many virtual points for monotonicity, as studied in Chapter 4 of this thesis. In the same way, the consideration of additional observations of the value of the partial derivative equal to zero at the ending time points, $\{\frac{\partial f_{it}}{\partial t} = 0 : t = 11\}$, and an observation model with the absence of noise for these observations, induced a stationary state at the ending of the time-series.

All these constraints have to be considered both in the sampling and predicting data points.

The cross-validation scheme *LOLO-CV* based on leaving a whole time-series out of the training data, has been carried out in order to evaluate the prediction performance of complete new time-series at new locations. Figures 5.5 and 5.11 show the predicted time-series at new locations following the cross-validation scheme *LOLO-CV* for the GP model and the splines model, respectively, with and without derivatives. Predictions are quite similar in both modeling approaches. Closer predictions to the actual data, narrower predictive intervals and good dynamics of the functions for the model with derivatives can be appreciated, because monotonicity and saturation constraints improves and reduces the credible intervals of the predictions. The model without derivatives shows decreasing patterns on the

functions which are not consistent with the prior knowledge. Credible intervals, especially at the last part of the splines-based functions (time-series), are getting much bigger. Here, it can be seen as imposing the monotonicity and saturation constraints on the splines functions improve considerably the credible intervals.

Monotonicity and long term saturation properties of the curves were not ensured using the models without derivative information. Hence, the proposed model with derivative information yields a better fit and predictions for dynamics of the functions, improving their interpretability. In this sense, the analysis of the color fading curves using a model without derivative information could not be done properly because the temporal degradation, especially at the last time points, is unrealistic.

Tables 5.3 and 5.5 show the MSE and ELPD computed following the two cross-validation scenarios, *LOO-CV* and *LOLO-CV*, for the GP and splines models, respectively.

The MSE following the *LOO-CV* of the model with derivatives is slightly lower than the model without derivatives, in both modeling approaches. The MSE following the *LOLO-CV* is lower for the model with derivatives than the model without derivatives, in both modeling approaches. Furthermore, when the uncertainty is taken into account in the evaluation with the ELPD statistic, the improvement of using derivatives is even considerably larger in both CV scenarios. Therefore, the results of these statistics confirm that the model with derivatives is closer to new data, either in terms of the expected log-density or the mean error. Furthermore, the predictive performance for this case study is similar in both modeling approaches, GPs and splines.

Prediction sensitivity due to the short set of data available has been found in both modeling approaches. Figures 5.5 and 5.11 show poor predictive performance in some spatial locations when compared to the observed data. This is due to the high sensitivity of the model to leaving some data out since the dataset is small.

The order of the covariance matrix is of $NT \times NT$ in the spatio-temporal GP model, and $N \times N$ in the spatially correlated splines model, requiring $O\big((NT)^3\big)$ and $O(N^3)$ computation expense, respectively, in the matrix inversion. This operation needs to be repeated at each HMC step with changing hyperparameters. This prevents using sampling methods for Bayesian inference on Gaussian process to fit and predict large data set, since the computational expenses increase rapidly with NT and N in each case respectively. In case of large data set, distributional approximation methods are recommended.

In order to make spatial continuous maps of color fading estimates, predictions of color fading time-series, $p(\tilde{\boldsymbol{y}}_j.|\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z})$, have been computed for all the spatial pixel locations j of the rock art painting image (Figure 5.1). As the predictive performance is similar in both modeling approaches, and the splines model is

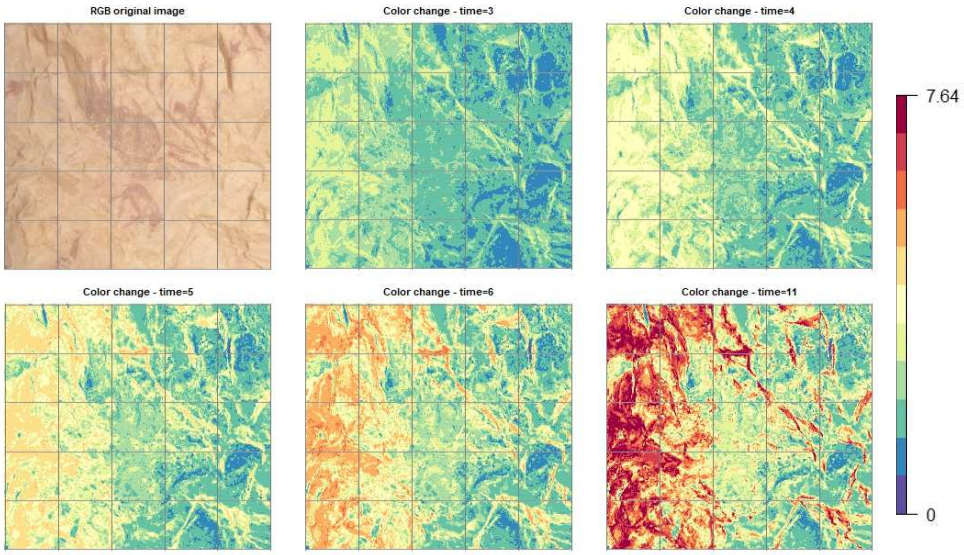


Figure 5.12: Spatial distribution over a continuous area of the rock art paintings image and the temporal evolution at time points $t = 3$, $t = 4$, $t = 5$, $t = 6$ and $t = 11$, of predicted mean color fading estimates from the model.

considerably faster, these maps have been computed using the splines approach. In Figure 5.12, six images representing the spatial distribution and their evolution over time of those color fading estimates are shown. The images correspond to the time points $t = 3$, $t = 4$, $t = 5$, $t = 6$ and $t = 11$, respectively. The spatial distribution over time of color fading values seems to be quite unvarying. This was expected since the time-series adjusted in the different locations have similar patterns, smooth, monotonically increasing and tending to saturate in the long term. Color fading values, especially when they are low values, like in this case (the maximum is of 7.64), are not worth being converted to the RGB color space and plotted as an image, because the color changes will be not visible in a RGB image. The best way is plotting color ΔE^*_{ab} values (Figure 5.12). The science of colorimetry argues that ΔE^*_{ab} values higher that approximately 3.5 would be perceptible for the human eye looking at the real object [Malacara, 2011].

The actual equivalency of the time points ($t = 1, \ldots, 11$) used in MFS measurements in years depends on the hours and intensity of sunlight that affects the paintings on a changing daily basis. Without proper monitoring of light, this equivalency is difficult to obtain. Although this aspect of the research was not considered in the current study, future work will include an evaluation of the location and

geographical orientation of the paintings together with long-term monitoring of light and UV radiation with the aim of estimating the dose acting upon the paintings in years.

5.13 Conclusion

Color is an important aspect in the documentation and conservation of the historical materials, such as rock art paintings, so the knowledge of the potential color degradation on these systems is crucial for eventual safeguarding and conservation. MFS measurements are difficult and lengthy to materialize, especially in these rock art systems, so an interpolation procedure in order to make predictions for other locations on the surface is important. Furthermore, these measurements in these systems are contaminated with large fluctuations, so the consideration of constraints in the modeling in order to overcome possible modeling issues that may arise due to these large fluctuations are highly encouraged.

We have formulated two reliable modeling frameworks: one based on Gaussian processes (GPs) and another on spatially correlated time-series. In both models the regular process is jointly modeled with the derivative process, thus model constraints related to the derivative of the functions could be included in the model in order to fit the desired properties of the MFS functions and minimize the effects of largely fluctuations in the original observations.

A GP model properly exploits the spatio-temporal covariance structure of the data by means of its multi-dimensional (space and time) covariance function. Furthermore, the GP has been extended to jointly model the regular and derivatives observations, and estimate a joint covariance function between regular and derivative, making it a more informative and rich model and, at the same time, guaranteeing that the functions are non-decreasing as a function of time.

However, the spatially correlated splines time-series model, which correlates the time-series by means of correlating their splines coefficients, requires less computation than the GP model, and the predictive performance in this case study is similar to the GP, as we can see in CV predictions and MSE and ELPD statistics. The computation requirements to invert the covariance matrix in the spatially correlated time-series model has been reduced substantially compared to using a GP model with a spatio-temporal covariance function. The spatially correlated time-series model framework, where a complete covariance structure among splines coefficients is considered, requires $O\big((NK)^3\big)$ computational demand in the covariance matrix inversion, whereas spatio-temporal GPs require $O\big((NT)^3\big)$. Notice that the number of knots K in spline-based models is usually much lower than the number of time points T ($K \ll T$). In case of null covariance is considered between splines

coefficients belonging to different spline knots, the computational expenses of the correlated splines model becomes $O(N^3K)$. And furthermore, if the same spatial structure is considered for the splines coefficients belonging to different spline knots, as we implemented in the present work, the computational expenses of the proposed model becomes $O(N^3)$.

Taking into account these first order constraints demonstrated being beneficial, either in terms of predictive performance or application-specific interpretability. Predictive capacity (MSE ans ELDP) are considerably better with the model with derivatives compared to the model without derivatives. However, high sensitivity of the models to leaving some data out has been found due to the data set is small.

A multivariate covariance function in a GP has allowed the usage of many predictors to evaluate the covariance structure of the data. In this sense, we have been able to include the color space variables, the spatial distance in the covariance function, and demonstrated the colorimetric variables being useful to correlate MFS data. The contribution of the spatial positions on this covariance structure has been found to be quite weak, consequently, often and widely used traditional spatial or spatio-temporal models cannot detect a useful correlation structure among the data.

Reliable color fading estimates evolution maps can be elaborated by means of using the proposed model with derivative information in comparison to the model without derivative information.

Finally, multivariate covariance metrics and zero and first order constraints might be very hard to implement outside of a Bayesian framework and Gaussian process models. A Gaussian process is flexible enough and allowed us to properly model this complex covariance structure of the time-series dependent on different input covariates. The Bayesian framework has allowed us to jointly use normally distributed observations with probit distributed observations of the sign of the partial derivatives, allowing to fulfill the determinants on the behavior of the functions.

5.A Predictive distributions versus the predictors in the GP modeling approach

The posterior distributions of the process, $p(\boldsymbol{f}|\boldsymbol{y}, \boldsymbol{m}, \boldsymbol{z})$, versus the input variables H, S and I, and the time points, are plotted in Figure 5.13. The variables H, S and I belong to the spatial dimension. Function $\boldsymbol{f}$ is plotted for the different time points.

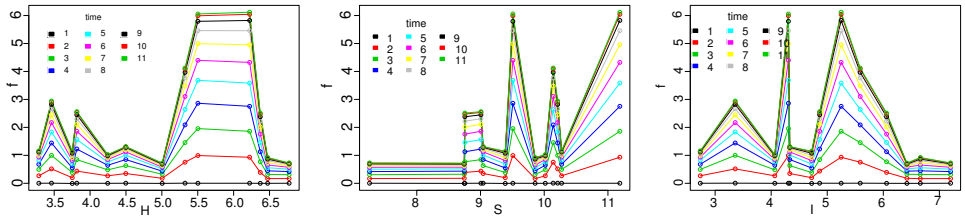


Figure 5.13: Posterior means of the latent Gaussian process $\boldsymbol{f}$ versus the input variables H, S, and I, for specific time points.

5.B Posterior covariance matrix in the GP modeling approach

The joint covariance matrix K of the process is visualized in Figures 5.14 and 5.15. Figure 5.14-(left) shows the part of the covariance matrix that involves the regular process. Figure 5.14-(right) shows a submatrix of the covariance matrix of the regular process, in which the spatio-temporal covariance structure of three spatial locations and their time-series can be appreciated. The black lines in Figure 5.14-(left) divide the covariance matrix according to whether it involves covariances among the 13 training data (K_{ff}) or among the 4 predicting data ($K_{\tilde{f}\tilde{f}}$) or among the interaction between training and predicting data ($K_{f\tilde{f}}$ and $K_{\tilde{f}f}$).

Figure 5.15 shows the parts of the covariance matrix that involve the regular process and its derivatives (derivative process). The block $K_{ff'}$ contains the covariances among regular and derivative observations for the training data. The block $K_{f'f'}$ contains the covariances among derivative observations for the training data. The block $K_{\tilde{f}\tilde{f}'}$ contains the covariances among regular and derivative observations for the predicting data. The block $K_{f'\tilde{f}'}$ contains the covariances among derivative observations for training and predicting data. And, finally, the block $K_{\tilde{f}'\tilde{f}'}$ contains the covariances among derivative observations for predicting data.

The subindexes of the matrix K denote the type of observations that are involved in each covariance block: f - regular observations of training points; $\tilde{f}$ - regular observations of predicting points; f' - derivative observations of training points; $\tilde{f}'$ - derivative observations of predicting points.

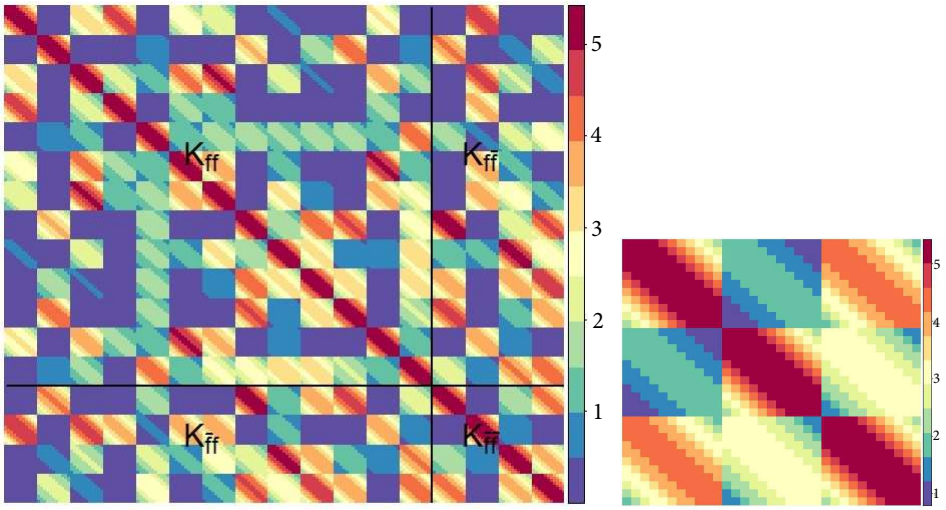


Figure 5.14: Covariance matrix image for the actual and predictive observations of the regular process (left). Extract from the covariance matrix containing the spatio-temporal covariances of the time-series of three locations (right).

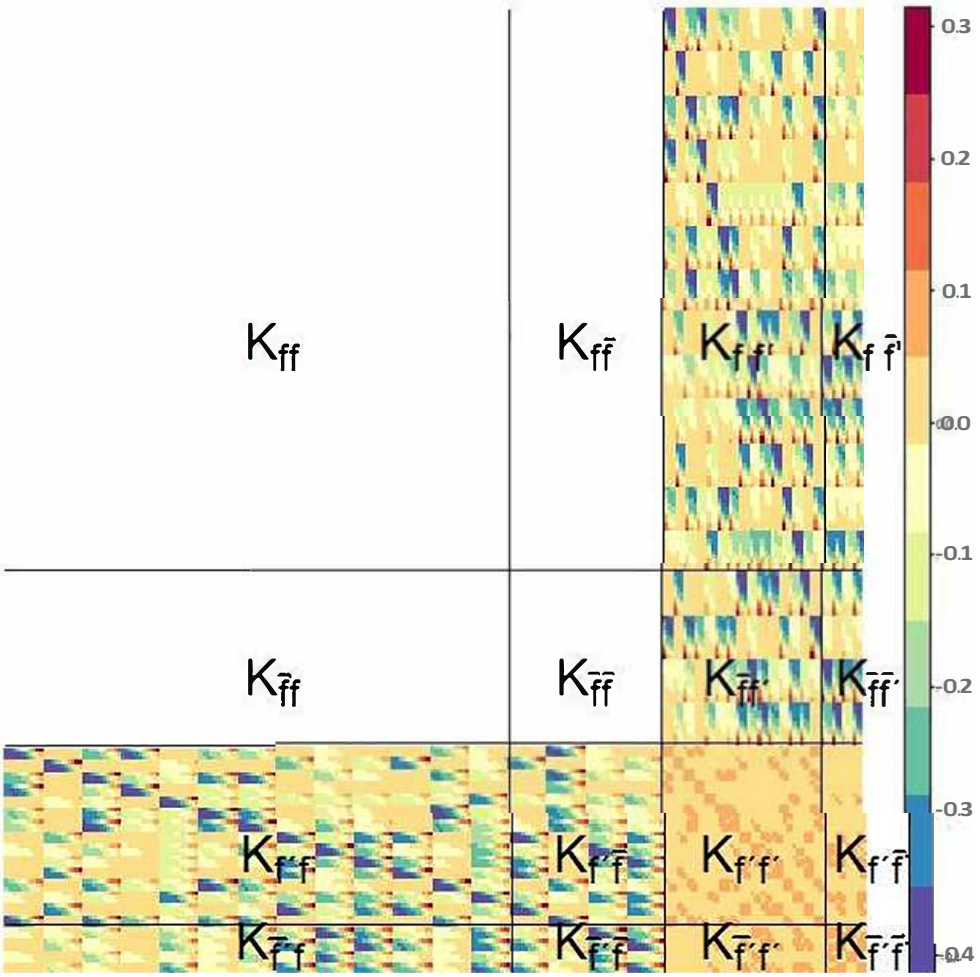


Figure 5.15: Covariance matrix image between the process and its derivatives.

Chapter 6

Application to image sensor noise: Hierarchical modeling for estimating noise in image sensors

In this chapter, an applied study on a specific problem aimed at decomposing and estimating the noise sources in image sensing is carried out. This applied study is a simple and excellent example of the potential and flexibility of the hierarchical Bayesian modeling framework in order to naturally and accurately propagate uncertainty and solve complex applications with multiple factors and multilevel structures.

6.1 Summary

This chapter presents a Bayesian approach to modeling the sensor noise components involved in imaging with digital cameras. Sensor noise sources cause differences in the signal recorded among pixels in a single image and among multiple images. We argue that a Bayesian multilevel random-effects model is suitable for approaching image sensing and its major noise components in the output image. A Bayesian hierarchical model is fitted using Markov chain Monte Carlo (MCMC) sampling methods. Bayesian inference is based on the joint posterior distribution of the model parameters and MCMC gives an estimate of this posterior distribution. The dataset used for fitting the model consists of trichromatic image data collected under different reflectance and shooting conditions, and the model parameters are formulated and estimated in units of digital numbers. The Bayesian approach provides reliable and flexible modeling of the imaging noise parameters, propagating uncertainties accurately. A feasible correspondence of the noise parameters to their

expected theoretical behaviors and accurate noise parameters estimates are found in the present study. The Bayesian approach can be extended to formulate further components aimed at identifying even more specific parameters of the imaging process.

6.2 Introduction and related work

Some industrial applications require calibrated image sensors. An accurate procedure to model the image sensor noise is useful for optimizing sensor design as well as for finding out how much uncertainty, in their different sources, is present in an image [Dierks, 2004, EMVA, 2010, Kuroda, 2014]. These noise sources are reflected on differences in the signal recorded among pixels in a single image and among multiple images coming from different shoots (image captures), and the variances of these noises are dependent on the level of reflectance imaged.

Image sensing and its different noise components are well documented in the literature, where their manifestations and relationships are well defined. Aguerrebere, Delon, Gousseau, and Musé [Aguerrebere et al.], De-Jiang and Tao [2011], Reibel et al. [2003] and Dierks [2004] are excellent introductory references for understanding and extracting a model for image sensing. Based on those references, in Section 6.4 we describe and formulate a conceptual model for all the components involved in image sensing data. This conceptual model will be used as the data-generating model for the proposed hierarchical Bayesian model in Section 6.6.1.

Healey and Kondepudy [1994], Tsin et al. [2001] and Campos [2000] are also excellent references where authors determine a model for the sensing process that additionally include the description of other stages in the practice of imaging applications, such as the optical system of the camera [Campos, 2000], image processing parameters [Tsin et al., 2001] and reflectance and illumination variations [Healey and Kondepudy, 1994]. There are many other references, from different disciplines such as machine vision, photometry, physics, and electronics, dealing with the definition of all the parameters involved in the sensing process [Granados et al., 2010, Grant, 2005, Han et al., 2011, Zhang et al., 2011]. The fact is that their interpretations agree on the process of image sensing.

In order to estimate the effects and contributions of the sensor noise parameters on the output image, the data-generating model of the process is fitted to a set of observed image data. For this purpose, classical iterative optimization algorithms based on maximum likelihood and point estimates for the parameters [Dong et al., 2018, Healey and Kondepudy, 1994, Tsin et al., 2001] and the well-known Photon transfer method [Janesick et al., 1987], are the current methods used for fitting

and inferring those noise components. In particular, the Photon transfer method is the technique used to characterize the noise parameters in the ISO [ISO, 2013] and EMVA [EMVA, 2010] standards for image noise measurements. The Photon transfer method [De-Jiang and Tao, 2011, Dierks, 2004, Reibel et al., 2003] is based onnested independent and point estimate computations, which induces probably high error propagation, apart from the need to do some of other assumptions such as normality and independence between some parameters (i.e. photo response non-uniformity and photon noise).

The present study is focused on Bayesian modeling and inference of those sensor noise components. In the Bayesian context [Bernardo and Smith, 2009, Gelman et al., 2013, Jaynes, 2003], all inference is based on the (multivariate) joint posterior distribution of the model parameters and hyperparameters. Computing the joint posterior distribution is often difficult and, for this reason, different computation approaches can be used. Markov chain Monte Carlo (MCMC) [Brooks et al., 2011] are sampling methods that provide samples of the joint posterior distribution of parameter θ given data y $\big(p(\theta|y)\big)$. These samples can be used to make inference and assess the significance of the different parameters in the model. In addition, interactions between different terms can be easily explored by means of their joint posterior distributions.

Bayes estimates have many advantages compared to point estimates of classical methods, as it was already stated in Section 2.4 of Chapter 2. Furthermore, Bayesian hierarchical modeling permits construct models with complex structures, while propagating all sources of uncertainty in inferences, as it was also commented in Chapter 2 and Section 2.3.4. Bayesian hierarchical models naturally leads to more reliable inferences and better real-world answers [Browne et al., 2006, Gelman et al., 2013]. All these advantages motivates the present work and the use of a Bayesian approach for the study of image sensing noise components.

6.3 Contributions of the study

In the present study, a novel Bayesian approach in the field of sensor noise imaging characterization is presented. A probabilistic model based on the data-generating model is fitted to a set of a time-series of images with different reflectance and wavelengths under uniform illumination conditions. The data-generating model adds independent random components nested inside other fixed components, i.e. a multilevel random-effects model with multiple factors and grouping. The unknown parameters in the model, which are the parameters of the noise components of the process, are learned through sampling methods based on MCMC.

The flexibility, accuracy, and intuitiveness of the Bayesian framework for mod-

eling and calibrating the sensor noise components is worth noticing. The results show a reliable and flexible modeling, able to naturally and accurately propagate uncertainties of noise parameters.

The rest of the chapter is structured as follows. Section 6.4 reviews all the noise components, their manifestations and relationships, formulating the data-generating model (theoretical model) for image sensing. Section 6.5 describes the available experimental data. Section 6.6 focuses on the modeling and inference formulation of the proposed Bayesian multilevel random-effects model. Section 6.7 analyzes the results of fitting the proposed model on the experimental data. Section 6.8 describes the procedures used for model checking and assessment. 6.9 discusses the standards for image noise measurements and make a brief qualitative comparison with the proposed statistical modeling. Finally, Section 6.10 draws some conclusions.

6.4 Image sensing model

A digital image is formed once the electromagnetic energy coming from or reflected by an object is registered into an image sensor at a certain instant after shooting (image capture). Objects mainly send out reflected energy that originally comes from light sources, either natural or artificial. The reflectance of an object represents the capacity to reflect ligth energy and is usually considered as a continuous factor between 0 and 1, where zero represents null reflectance and one total reflectance [Pratt, 2007]. An image sensor is composed of many individual sensing elements (pixels) arranged in a regular matrix that registers incoming light at a certain instant or shoot.

Basically, photons of energy emitted from and reflected by the object are captured by a single pixel. Each one of the photons inside the pixel has a probability, called quantum efficiency, to create a free electron. Then, from the incoming photons, a number of electrons are created inside the pixel. Finally, the electrons, after being converted into a voltage, are amplified and digitized into an output digital number, also known as the gray level or intensity value of an image [Dierks, 2004, Healey and Kondepudy, 1994, Tsin et al., 2001].

Following De-Jiang and Tao [2011], Reibel et al. [2003] and Dierks [2004], a simple model of the output digital numbers y_{it} registered in the i'th pixel and at the t'th image shoot, as a function of the reflectance r of the reflective object and the wavelength w of the light, can be written as follows:

$$y_{it}(r,w) = K_i \cdot e_{it}(r,w) + \mu_K \cdot D_i + \mu_K \cdot C_t(r,w) + \mu_K \cdot R_{it} + A_{it}. \qquad (6.1)$$

The number of electrons $e_{it}(r,w)$ is a function of the number of photons coming into the pixel and of the probability $q(w)$ of creating a free electron from an incoming photon by the pixel sensing element. A model for the electrons is usually

approximated as a Poison model

$$e_{it}(r,w) \sim Po\big(q(w)\cdot\mu_p(r,w)\big) = Po\big(\mu_e(r,w)\big), \tag{6.2}$$

where $\mu_e(r,w)$ is the mean number of the electrons e created from the incoming photons inside the pixel. $\mu_p(r,w)$ is the mean number of the incoming photons p which are dependent on the reflectance r and the wavelength w. The probability $q(w)$ is also depending on the wavelength of the light. The variances $\mu_p(r,w)$ of these Poisson variables are called photon noise and represent the variances of the incoming energy in function of reflectance and wavelength. Note that the variance of $\mu_e(r,w)$ also represents the photon noise, since it is directly proportional to the mean number of photons. Moreover, it should be noted that photon noise is always present in images and is never dependent on the camera sensor.

The gain factor variable K_i governs the process of converting electrons into voltage, its amplification and digitalization [Dierks, 2004, Healey and Kondepudy, 1994, Reibel et al., 2003]. There is evidence in the literature of considering K_i contaminated with Gaussian noise (6.3) which represents one part of the spatial noise of image sensors, commonly named photo response non-uniformity (PRNU). PRNU models the inter-pixel differences when generating electrons from the incoming photons [Dierks, 2004, Gow et al., 2007, Reibel et al., 2003], which are due to pixel pitch and other pixel characteristics [Dierks, 2004, Gow et al., 2007].

$$K_i \sim N(\mu_K, \sigma_K^2) \tag{6.3}$$

In the previous equation (6.3), μ_K and σ_K^2 are the mean and variance of the variable K_i.

In addition to the electrons $e_{it}(r,w)$ generated from the incoming light energy, current noise $C_t(r,w)$ is an effect by which free electrons can be thermally generated during the exposure time [De-Jiang and Tao, 2011, Dierks, 2004, Gow et al., 2007, Reibel et al., 2003] in the t'th image shoot. It is related to the temperature at a certain instant or shoot and is expected to be an effect varying only on the temporal dimension t, being constant across pixels [De-Jiang and Tao, 2011, Gow et al., 2007] . Establishing long intervals between shoots and small exposure times, trying to maintain low temperatures in the sensor, $C_t(r,s)$ could be considered random and modeled as a Poisson stochastic variable [EMVA, 2010, ISO, 2013, Marqués-Mateu et al., 2013]. Temperature inside a pixel depends on the incoming light [De-Jiang and Tao, 2011, Dierks, 2004, Gow et al., 2007], then current noise will be an effect dependent on reflectance r and wavelength w (6.4).

$$C_t(r,w) \sim Po(\mu_C(r,w)) \tag{6.4}$$

In the previous equation (6.4), $\mu_C(r, w)$ is the mean of the variable C as a function of reflectance and wavelength.

Apart from the light induced electrons, dark electrons D_i are generated in the i'th pixel without the presence of incident light. They are generated from dark current variations across pixels, and commonly named fixed pattern noise (FPN). This is an effect affecting the spatial dimension, being the same in all different frames or shoots [De-Jiang and Tao, 2011, Dierks, 2004, El Gamal et al., 1998]. Although some cameras may have some kind of non-random spatial pattern [Campos, 2000], for most camera sensors this spatial pattern is completely random [El Gamal et al., 1998] following a Poisson model

$$D_i \sim Po(\mu_D), \tag{6.5}$$

where μ_D is the mean of the variable D.

Moreover, reset noise R_{it} refers to the remaining electrons in the circuitry capacitors even after being emptied in the previous exposure. It is expected to be an effect defined independently on both dimensions i and t and completely random, so modeled by a Poisson variable

$$R_{it} \sim Po(\mu_R), \tag{6.6}$$

where μ_R is the mean of the variable R.

The parameters D_i, $C_t(r, w)$ and R_{it} are multiplied in eq. (6.1) by the mean gain parameter μ_K, to encapsulate the process of converting electrons into digital numbers.

Finally, after the charge is transferred, and converted into a voltage, amplified and digitized, the noise effects amplifier, flicker noise $(1/f)$ [Han et al., 2011] and quantization add also some noise A_{it} to the final output digital number [De-Jiang and Tao, 2011, Dierks, 2004, Han et al., 2011]. They are expected to be random and normally distributed

$$A_{it} \sim N(\mu_A, \sigma_A^2), \tag{6.7}$$

where μ_A and σ_A^2 are the mean and variance of the variable A.

The variabilities of K_i (PRNU), D_i (FPN), $e_{it}(r, w)$ (photon noise), $C_t(r, w)$ (current noise), R_{it} (reset noise), and A_{it} (amplifier, 1/f and quantization noises) will be the essential parameters of an image sensor and the quantities of interest to be estimated from the model as noise parameters in this work. Photon noise is always present in an image and is never dependent on the camera sensor. The other noise parameters are dependent on the camera sensor, and so will be the parameters to compare the quality of different image sensors. The quantum efficiency is also clearly a very important parameter of quality, although it can only be estimated

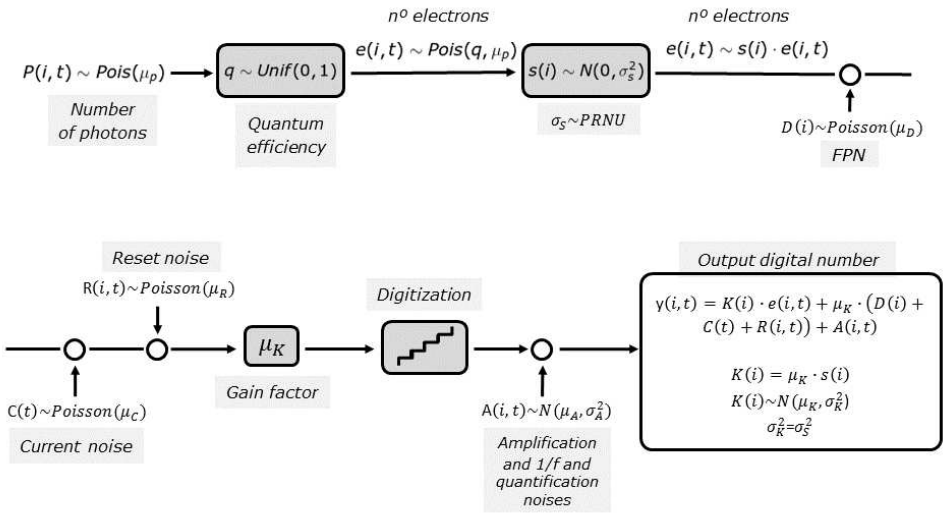


Figure 6.1: Flowchart of the image sensing characterized for this work. for specific values of reflectance r and wavelength w. The index i denotes the pixel dimension and the index t denotes the exposures.

with the measurement, by means of a radiometer device, of the number of incoming photons into any individual pixel.

Figure 6.1 shows a detailed flowchart of the image sensing model that includes all the parameters and relationships that have been defined above. This flowchart has been constructed considering specific values of reflectance r and wavelength w, so they are not included in the noise components depicted in the Figure.

6.5 Data description

The experiment consisted of time-sequential imaging of a ColorChecker by using a trichromatic image sensor camera. A ColorChecker is a reflectance calibration pattern which contains several reflectance patches, each one with constant reflectance (Figure 6.2). The real reflectance values of each one of these patches are unknown, so the reflectance factor will be considered as levels of a categorical factor in the modeling (Section 6.6). A trichromatic colorimeter provides simultaneous measurements of three primary wavelengths (usually Red R, Green G, and Blue B). The result of the experiment is a time series of images with a spatially arranged matrix of pixel-values across the sensor in each image with different reflectance and wavelengths.

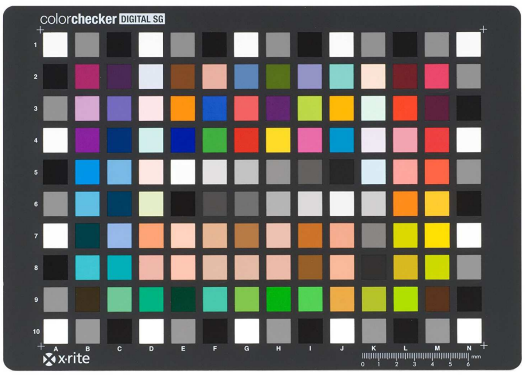


Figure 6.2: Reflectance calibration pattern.

The experimental data was comprised of 60 images from different shoots (t=1,...,60). Samples of 500 random pixels (i=1,...,500) from 11 different reflectance patches (r=1,...,11) were provided for each image, resulting in 5500 pixels through the sensor, 500 grouped pixels for each one of the 11 reflectance patches. Finally, three different wavelength ranges of the light were used (w=1,2,3) for each pixel.

One hundred out of this five hundred pixels within each reflectance patch were used as testing observations for the posterior predictive checks, in order to check and validate the model in equation (6.9). Therefore, only 400 pixels in each one of the reflectance patches were used to fit the model.

In order to get uniform average conditions on the experiment, stable and homogeneous incident light on both dimensions, spatial and temporal, was needed. The experiment was conducted under laboratory conditions using a typical colorimetry setup following the recommendations of the *Commission Internationale de l'Éclairage* [CIE, 2004].

The imaging device used in the experiments was the Foveon X3® Pro 10M CMOS sensor which has a stack of three photosensitive layers and provides true trichromatic imagery. It is considered as a high-class device that provides extremely low-noise readout and removes typically fixed pattern noise associated with other CMOS sensors [Merrill, 1999]. The dynamic range of the sensor is 12 bits (0-4095 digital numbers or grey levels or intensity values), the total number of pixel sensors is 2268 columns x 1512 rows x 3 layers, or 3.4 million pixels per layer, and the pixel pitch of the array is 9.12 μm. This sensor also provides other interesting practical features such as low power consumption, variable pixel size, and blooming immunity.

6.6 Proposed modeling and inference

6.6.1 Multilevel random-effects model

We propose a multilevel random-effects model to approach the theoretical model in equation (6.1) and its components. Previously, if we approximate the Poisson variables $e_{it}(r,w)$ in equation (6.1) as Normal variables,

$$e_{it}(r,w) \sim N\big(\mu_e(r,w), \sigma_e^2(r,w)\big), \tag{6.8}$$

then we can rewrite the model in (6.1) as follows:

$$\begin{aligned} y_{it}(r,w) = \quad & \mu_K \cdot \mu_e(r,w) + dK_i \cdot \mu_e(r,w) + \mu_K \cdot de_{it}(r,w) \\ & + \mu_K \cdot D_i + \mu_K \cdot C_t(r,w) + \mu_K \cdot R_{it} + A_{it}, \end{aligned} \tag{6.9}$$

where dK_i and $de_{it}(r,w)$ are the remaining zero-mean normal variables after removing the means μ_K and $\mu_e(r,w)$, of variables K_i and $e_{it}(r,w)$ in equation (6.1), respectively:

$$dK_i \sim N(0, \sigma_K^2), \tag{6.10}$$

$$de_{it}(r,w) \sim N\big(0, \sigma_e^2(r,w)\big). \tag{6.11}$$

In the previous equation (6.9), the component $dK_i \cdot de_{it}(r,w)$ has not been taken into consideration because it yields a very low component.

Thus, the model in (6.9) will be the model to be approached by means of the proposed multilevel random-effects model. We assume that we have an array of observations $\boldsymbol{y} \in \mathbb{R}^{N \times T \times R \times W}$ of image digital numbers (grey levels/intensity values), with an element $y_{it}(r,w)$ representing an observation of the image digital number registered at the i'th pixel and at the t'th image shoot and as a function of the levels r and w. Similarly to the previous Section 6.4, N denotes the pixels in an image sensor ($i = 1, ..., N$), T denotes the number of image exposures ($t = 1, ..., T$), R denotes the number of levels of reflectance examined ($r = 1, ..., R$) and W denotes the levels of wavelengths examined ($w = 1, ..., W$). r and w are categorical variables.

The collection $\boldsymbol{y}$ of observations is considered to follow a Normal distribution depending on an underlying mean function $\boldsymbol{f}$ and standard noise σ,

$$p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma^2 \boldsymbol{I}), \tag{6.12}$$

where $\boldsymbol{I}$ is the identity matrix. The mean function $\boldsymbol{f}$ is a sum function of independent random effects nested inside the fixed effects of the categorical variables r and w.

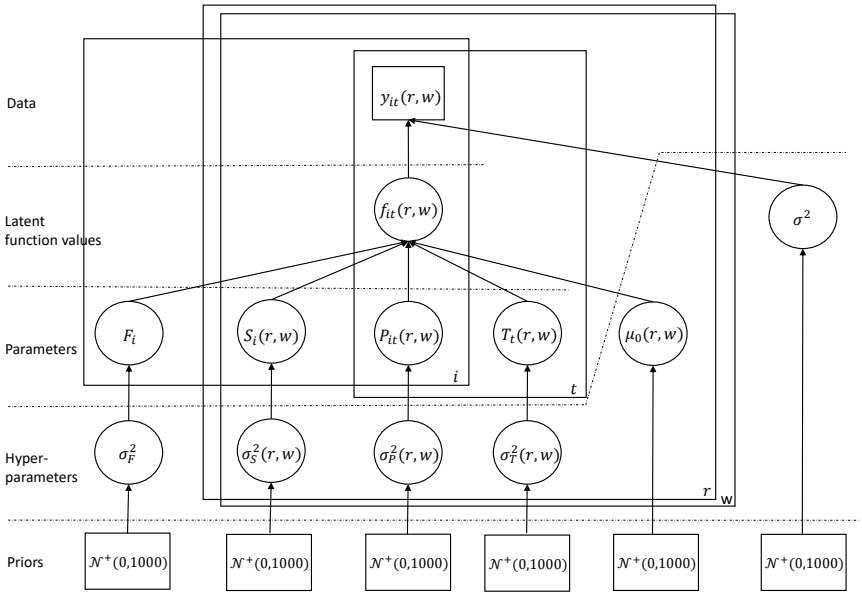


Figure 6.3: Directed acyclic graph of the proposed Bayesian model in equations (6.12) and (6.13).

Thus, for a single observation (i, t), the underlying function takes the form:

$$f_{it}(r, w) = \mu_0(r, w) + S_i(r, w) + F_i + T_t(r, w) + P_{it}(r, w). \tag{6.13}$$

In Figure 6.3 the directed acyclic graph of the proposed Bayesian model in equations (6.12) and (6.13) is depicted. The parameter $\mu_0(r, w)$ is the fixed effect as a function of the categorical variables r and w. It gathers component $\mu_K \cdot \mu_e(r, w)$ in the theoretical model shown in equation (6.9), which represents the mean reflectance as a function of r and w.

The parameter $S_i(r, w)$ in equation (6.13) models component $dK_i \cdot \mu_e(r, w)$ in the theoretical model shown in equation (6.9), where $\mu_e(r, w)$ is the mean electrons as a function of reflectance r and wavelength w, and dK_i is the PRNU zero-mean normal random variable as seen in equation (6.10). Therefore, $S_i(r, w)$ can be modeled as a zero-mean normal prior distribution, defined on the pixel dimension i and as a function of r and w:

$$p\big(S_i(r, w)|\sigma_S(r, w)\big) = \mathcal{N}\big(S_i(r, w)|0, \sigma_S^2(r, w)\big). \tag{6.14}$$

The standard deviation $\sigma_S(r, w)$ of this parameter S models the PRNU for specific

levels of r and w.

The parameter F_i in equation (6.13) models the component $\mu_K \cdot D_i$ in the theoretical model shown in equation (6.9), where μ_K is a constant and D_i is the FPN Poisson distributed as seen in equation (6.5). The assumption of considering Poisson generated electrons after their conversion to digital numbers ($\mu_K \cdot D_i$) to normally distributed variables is truly reasonable in this context. Therefore, the parameter F_i is modeled following a zero-mean Normal prior distribution:

$$p\big(F_i|\sigma_F\big) = \mathcal{N}\big(F_i|0, \sigma_F^2\big), \tag{6.15}$$

whose standard deviation σ_F represents the FPN of the image sensor, which does not depend on reflectance or wavelength.

The parameter $T_t(r, w)$ in equation (6.13) models component $\mu_K \cdot C_t(r, w)$ in the theoretical model shown in equation (6.9), where μ_K is a constant and $C_t(r, w)$ is the current noise Poisson variable as seen in equation (6.4). Like in the previous case of $\mu_K \cdot D_i$, the Poisson variable $\mu_K \cdot C_t(r, w)$ can be approximated as a Normal prior distribution by the parameter $T_t(r, w)$ as seen below:

$$p\big(T_t(r, w)|\sigma_T(r, w)\big) = \mathcal{N}\big(T_t(r, w)|0, \sigma_T^2(r, w)\big). \tag{6.16}$$

The standard deviation $\sigma_T(r, w)$ of this parameter T will represent the current noise for specific levels of r and w.

The parameter $P_{it}(r, w)$ in equation (6.13) models component $\mu_K \cdot de_{it}(r, w)$ in the theoretical model shown in equation (6.9), where μ_K is a constant and $de_{it}(r, w)$ is the photon noise approximated as a zero-mean Normal variable as seen in equation (6.11). Then, $P_{it}(r, w)$ is modeled as a zero-mean Normal variable:

$$p\big(P_{it}(r, w)|\sigma_P(r, w)\big) = \mathcal{N}\big(P_{it}(r, w)|0, \sigma_P^2(r, w)\big), \tag{6.17}$$

where its variance $\sigma_P^2(r, w)$ represents the photon noise for the specific levels of r and w.

Finally, the residual of the model in equation (6.12) will gather component $\mu_K \cdot R_{it}$ and A_{it} in the theoretical model shown in equation (6.9) which is expected to be a random Normal variable defined independently on both dimensions i and t in equations (6.6) and (6.7). These residuals will also contain other possible independent and uncontrolled random noise factors in the experimentation or even in the process.

The likelihood function of the observations $\boldsymbol{y}$ given the parameters $\boldsymbol{\mu}_0 = \{\mu_0(r, w)\}$, $\boldsymbol{S} = \{S_i(r, w)\}$, $\boldsymbol{F} = \{F_i\}$, $\boldsymbol{T} = \{T_t(r, w)\}$, $\boldsymbol{P} = \{P_{it}(r, w)\}$, and

σ, can be seen below:

$$p(\boldsymbol{y}|\boldsymbol{\mu}_0, \boldsymbol{S}, \boldsymbol{F}, \boldsymbol{T}, \boldsymbol{P}, \sigma) = \prod_{\forall i,t,r,w} \mathcal{N}\big(y_{it}(r,w)|\mu_0(r,w), S_i(r,w), F_i, T_t(r,w), P_{it}(r,w), \sigma\big). \quad (6.18)$$

6.6.2 Bayesian inference

Bayesian inference is done over the joint posterior distribution of parameters and hyperparameters given the data, which is proportional to the likelihood and priors, assuming independent priors among the effects:

$$\begin{aligned} p(\boldsymbol{\mu_0}, \boldsymbol{S}, \boldsymbol{F}, \boldsymbol{T}, \boldsymbol{P}, \sigma|\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{\mu_0}, \boldsymbol{S}, \boldsymbol{F}, \boldsymbol{T}, \boldsymbol{P}, \sigma) p(\boldsymbol{\mu_0}) p(\boldsymbol{S}|\boldsymbol{\sigma}_S) p(\boldsymbol{F}|\sigma_F) \\ &\cdot p(\boldsymbol{T}|\boldsymbol{\sigma}_T) p(\boldsymbol{P}|\boldsymbol{\sigma}_P) p(\sigma) p(\boldsymbol{\sigma}_S) p(\sigma_F) p(\boldsymbol{\sigma}_T) p(\boldsymbol{\sigma}_P) = \\ &\Big(\prod_{\forall i,t,r,w} \mathcal{N}\big(y_{it}(r,w)|\mu_0(r,w), S_i(r,w), F_i, T_t(r,w), P_{it}(r,w), \sigma\big) \Big) \\ &\times \Big(\mathcal{N}\big(\mu_0(r,w)|0, 1000\big) \Big) \Big(\prod_{\forall i,r,w} \mathcal{N}\big(S_i(r,w)|0, \sigma_S^2(r,w)\big) \Big) \Big(\prod_{\forall i} \mathcal{N}\big(F_i|0, \sigma_F^2\big) \Big) \\ &\times \Big(\prod_{\forall t,r,w} \mathcal{N}\big(T_t(r,w)|0, \sigma_T^2(r,w)\big) \Big) \Big(\prod_{\forall i,t,r,w} \mathcal{N}\big(P_{it}(r,w)|0, \sigma_P^2(r,w)\big) \Big) \\ &\times \mathcal{N}\big(\sigma|0, 1000\big)\, \mathcal{N}\big(\sigma_S(r,w)|0, 1000\big)\, \mathcal{N}\big(\sigma_F|0, 1000\big) \\ &\times \mathcal{N}\big(\sigma_T(r,w)|0, 1000\big)\, \mathcal{N}\big(\sigma_P(r,w)|0, 1000\big) \end{aligned} \quad (6.19)$$

In equation (6.19), $p(\boldsymbol{y}|\boldsymbol{\mu_0}, \boldsymbol{S}, \boldsymbol{F}, \boldsymbol{T}, \boldsymbol{P}, \sigma)$ is the likelihood of the model, and $p(\boldsymbol{S}|\boldsymbol{\sigma}_S)$, $p(\boldsymbol{F}|\sigma_F)$, $p(\boldsymbol{T}|\boldsymbol{\sigma}_T)$, and $p(\boldsymbol{P}|\boldsymbol{\sigma}_P)$ the priors for the corresponding parameters and $p(\boldsymbol{\mu}_0)$, $p(\sigma)$, $p(\boldsymbol{\sigma_S})$, $p(\sigma_F)$, $p(\boldsymbol{\sigma_T})$ and $p(\boldsymbol{\sigma_P})$ the priors for the hyperparameters, where $\boldsymbol{\sigma}_S$ denotes the collection $\{\sigma_S(r,w)\}$, and similarly $\boldsymbol{\sigma}_T = \{\sigma_T(r,w)\}$ and $\boldsymbol{\sigma}_P = \{\sigma_P(r,w)\}$. If no prior information is available for the hyperparameters, we still need to specify vague prior distributions. For the parameters $\boldsymbol{\mu}_0$, vague Normal distributions with large variances are defined. For the standard deviation parameters $\boldsymbol{\sigma}_S$, σ_F, $\boldsymbol{\sigma}_T$, $\boldsymbol{\sigma}_P$ and σ, we define positive half-Normal distributions with large variances [Kass and Wasserman, 1995, Yang and Berger, 1996].

The joint posterior distribution of the parameters have been estimated with MCMC using Gibbs sampling [Brooks et al., 2011, Geman and Geman, 1993] and WinBUGS software [Lunn et al., 2000, Ntzoufras, 2011]. Samples of the joint and marginal posterior distributions of the model parameters are obtained, and estimates and credible intervals are inferred for the model parameters. Three simulation chains have been launched for every one of the parameters, with 100000 iterations,

of which the first 30000 iterations were rejected as burn-in, and finally, only 1 of every 100 was retained with the aim of reducing the correlation in the samples. The convergence of the simulation chains was evaluated with the split-Rhat convergence diagnosis [Gelman and Rubin, 1992] and the effective sample size of the chains. A value of 1 in the split-Rhat convergence statistic indicates convergence of the simulation chain, although conventionally accepted values of convergence would be between 1 and 1.1. In this study, a split-Rhat value lower than 1.05 has been obtained for all parameter simulation chains.

6.7 Experimental results and analysis

In this study, we are interested in analyzing the standard deviation parameter estimates $\boldsymbol{\sigma}_S$, σ_F, $\boldsymbol{\sigma}_T$, $\boldsymbol{\sigma}_P$ and σ, which are the quantities that allow us to characterize the mean noise caused by parameters $\boldsymbol{S}$, $\boldsymbol{F}$, $\boldsymbol{T}$, $\boldsymbol{P}$ and $residuals$, respectively. Notice that the units of the estimated effects are units of image digital numbers.

As we will see in next Section 6.7.1, some of the noise estimates are reflectance dependent, fact that suggests the computation of their coefficients of variation, in which the linear effect of reflectance (linear-multiplicative effect of the mean number of electrons) on the parameters is removed. In this way, different sensors or different experimentations with different dynamic ranges can be compared.

The coefficient of variation (CV) is the ratio between the standard deviation and the mean of the component considered ($CV = \sigma/\mu$), that is, the inverse of the signal-to-noise ratio. In fact, the coefficient of variation defines the quality of a sensor as a discriminatory power of a signal. The overall means are represented by $\boldsymbol{\mu}_0$.

6.7.1 Standard deviation of the parameters

Figure 6.4 shows the 95% pointwise credible intervals for the parameters $\boldsymbol{\sigma}_S$, σ_F, $\boldsymbol{\sigma}_T$, $\boldsymbol{\sigma}_P$ and σ. They are plotted against the mean effects of the reflectance and wavelength variables which are modeled by the parameter $\mu_0(r, w)$.

As stated in Section 6.6, the parameter $\sigma_S(r, w)$ models the noise effect of component $\mu_e(r, w) \cdot dK_i$ in the theoretical model. The increasing of $\sigma_S(r, w)$with respect to reflectance r (x-axis) (Figure 6.4(a)) is due to the linear-multiplicative effect of the electrons $\mu_e(r, w)$ on dK_i, since dK_i is expected to be a zero-mean normal variable independent on reflectance. However, this linear behaviour that can be appreciated in the figure is broken at the lowest values of reflectance. In fact, it has been pointed out a non-linear interaction between PRNU (variability of dK_i) and the light intensity in low and high illumination levels [Gow et al., 2007].

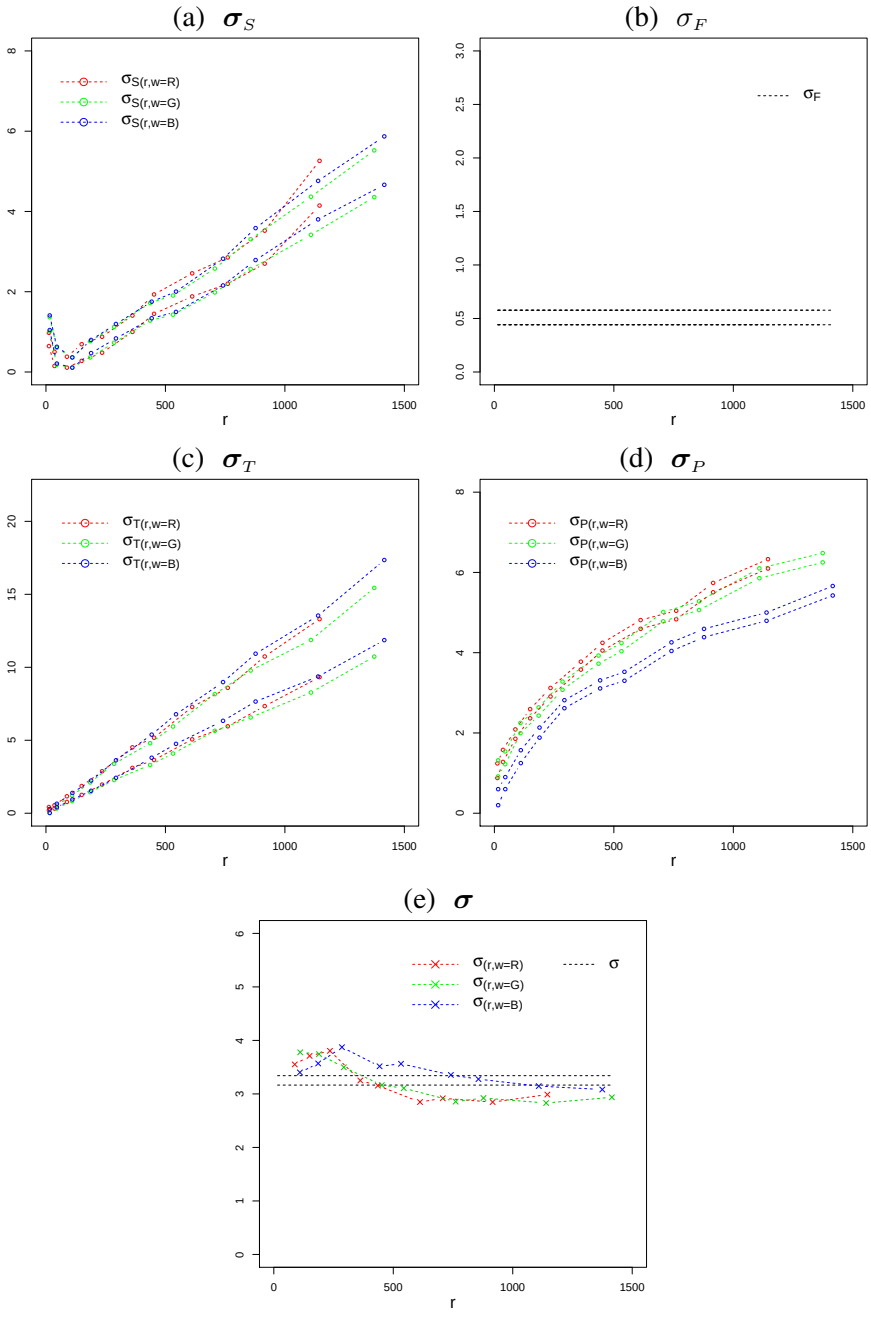


Figure 6.4: 2.5% and 97.5% posterior quantiles for the standard deviation of the parameters S ($\boldsymbol{\sigma}_S$) (a), F (σ_F) (b), T ($\boldsymbol{\sigma}_T$) (c), P ($\boldsymbol{\sigma}_P$) (d), and $residuals$ (σ) (e), versus mean output-reflectance r and wavelengths w. In (e), the residual deviation as a function of reflectance r and wavelength w ($\sigma_{(r,w)}$) is computed and plotted jointly with the mean residual deviation (σ).

The minimum values estimated for this noise parameter was around 0.2 and the maximum ones around 6.

The estimated noise effect σ_F is not dependent on reflectance and was estimated around 0.5 (Figure 6.4(b)). This represents the FPN and can be considered a negligible value for the Foveon X3® image sensor, as specified in the characteristics provided by the manufacturer.

The estimated noise effect $\sigma_T(r, w)$ shows a linear dependency with respect to reflectance r, either in mean or in variance (Figure 6.4(c)). This linear behaviour was expected since it represents the current noise which depends on temperature and therefore on reflectance as well.

The estimated noise effect $\sigma_P(r, w)$ does not increase linearly with respect to reflectance r (Figure 6.4(d)). As stated in Section 6.6, $\sigma_P(r, w)$ models the noise effect of component $\mu_K \cdot de_{it}(r, w)$ in the theoretical model. μ_K is a constant and the number of electrons $de_{it}(r, w)$ is a normal approximations to a Poisson variable, so that its standard deviation increases with the square root of the mean electrons $\left(\sqrt{\mu_e(r, w)}\right)$ (see equation (6.2)). Then, the increasing trend of $\sigma_P(r, w)$ as function of r will be due to the variance of $de_{it}(r, w)$ (photon noise) which increases with the square root of electrons or, equivalently, the squared root of reflectance.

Finally, the specifications of the sensor also indicate low-readout noise effects, for which and jointly with the reset noise, a mean error of 3.3 was estimated in this study by the mean residual deviation parameter σ in Figure 6.4(e). The residuals are not completely independent with respect to the reflectance r, and a slight decreasing trend in the residual deviation at low reflectance can be found. However, this lack of independence on the residuals is clearly very small with trend effects lower than 1 and, hence, can be considered negligible in practice. For reflectance > 700, the residuals are without trend. This fact reflects that some of the noise components (reset noise, amplifier noise, flicker noise, and quantization noise) included in the residual deviation parameter might be slightly dependent on reflectance at low intensities.

The gain factor μ_K is embedded in all the noise parameter estimates, so they represent units of image digital numbers (electrons times gain factor). The differences among wavelengths reflect different behaviours, that is, the wavelengths R, G, and B do not generate exactly the same noise under the same conditions.

6.7.2 Variation coefficients of the parameters

Due to the dependency of the noise estimates, $\boldsymbol{\sigma}_P$ (photon noise) of the parameter $\boldsymbol{P}$, $\boldsymbol{\sigma}_T$ (current noise) of the parameter $\boldsymbol{T}$, and $\boldsymbol{\sigma}_S$ (PRNU) of the parameter $\boldsymbol{S}$, on

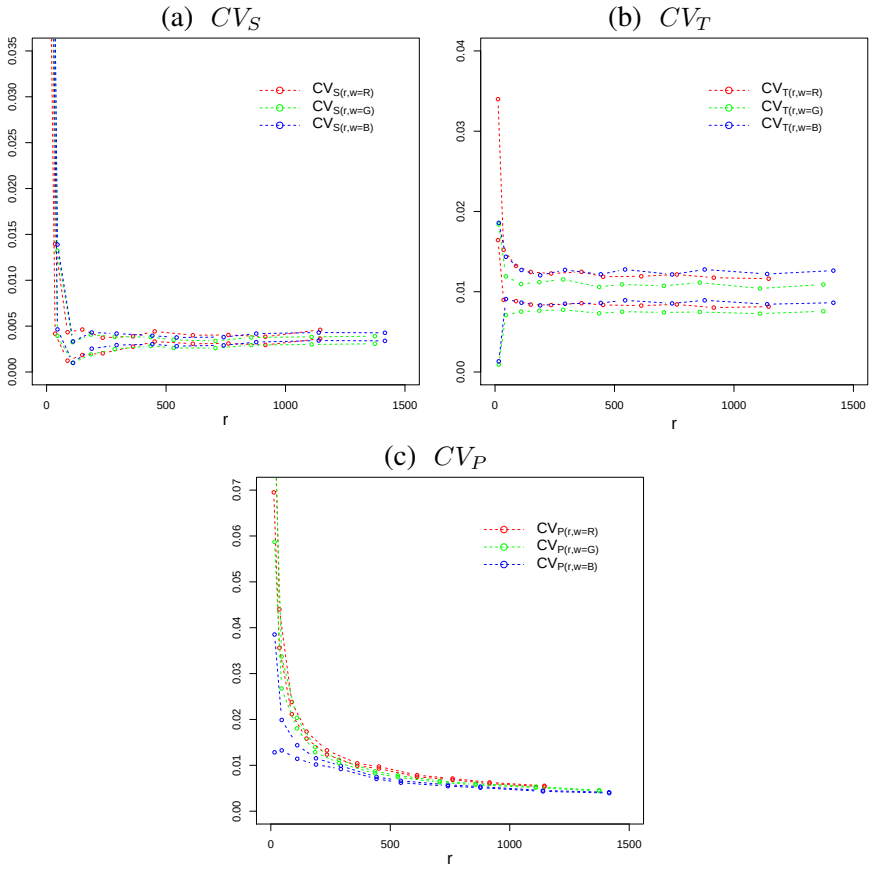


Figure 6.5: 2.5% and 97.5% posterior quantiles for the coefficient of variation of the parameter S (CV_S) (a), parameter T (CV_T) (b), parameter P (CV_P) (c), versus mean output-reflectance μ_0 and wavelengths w.

the level of reflectance imaged (Figures 6.4(a), 6.4(c), and 6.4(d)), their coefficients of variation are computed. For the parameters $\boldsymbol{F}$ and $residuals$ the computation of their variation coefficients make no sense since both the parameter $\boldsymbol{F}$ is an independent variable on reflectance (see equation (6.5)) and the residuals can be considered in practice variance-constant with respect to reflectance (see Section 6.7.1). Their absolute mean noise effects, σ_F and σ, were estimated around 0.5 and 3.3 digital numbers, respectively (Figures 6.4(b) and 6.4(e)).

Figure 6.5 shows the coefficients of variation of the parameters $\boldsymbol{S}$ (CV_S), $\boldsymbol{T}$ (CV_T) and $\boldsymbol{P}$ (CV_P).

The coefficient of variation CV_S in Figure 6.5(a) is mainly constant with respect

to reflectance, since the linear-multiplicative effect of the electrons $\boldsymbol{\mu}_e$ was removed, except for the lowest values of reflectance where $\boldsymbol{\sigma}_S$ has a non-linear behaviour as can be seen in Figure 6.4(a). The mean noise effect of PRNU (CV_S) was estimated at around 0.4% of the input signal, except for the lowest values of reflectance that reached up to 5%.

The coefficient of variation CV_T in Figure 6.5(b) is constant due to the fact that the linear effect of the light intensity on the current noise was removed. The mean noise effect of current noise (CV_T) was estimated at around 1% of the input signal.

As commented above, the slope of the noise effects $\boldsymbol{\sigma}_P$ in Figure 6.4(d) is due to the photon noise which increases with the square root of the mean number of electrons. In fact, when computing the coefficient of variation CV_P in Figure 6.5(c), it can be observed that the resulting slope is very similar to $1/\sqrt{\mu_K \cdot \mu_e(r,w)}$ which is the coefficient of variation of the photon noise in image digital numbers. This is estimated to be between 1.5% and 0.5% of the registered signal.

It can be stated, therefore, that the linear effect of the reflectance does not imply a lost of quality in the signal, since the coefficient of variation remains equal. However, it is an exception for the photon noise which does imply a lost of quality in the low values of the reflectance, as shown in its coefficient of variation in Figure 6.5(c). It is due to the inherent dependency of the variance of the electrons on the reflectance.

6.8 Model checking and validation

For model checking, common procedures of checking normality and tendencies on the predicted residuals for the set of test data can be used. The *probability integral transformation* (PIT) is a statistic that guarantees a good model performance and ensures that the model is compatible with the data. It is based on computing the probability of predictions to be lower or equal to their corresponding actual observed values [Gelfand et al., 1992, Gelman et al., 2013] for the set of test data,

$$\text{PIT}_{it} = P(y_{it}^{predicted} \leq y_{it}^{observed}).$$

Then, the similarity or provenance of these probabilities from a standard uniform distribution endorses these probabilities with the desirable property of having the same interpretation across models, which implies a good fit to the data and good prediction [Bayarri and Berger, 2000]. Using sampling methods computing the probability of a predicted value being minor the observed one is straightforward through the collection of simulated values.

Furthermore, the root mean square error (RMSE) of the predictions,

$$\text{RMSE} = \sqrt{\frac{1}{N \cdot T} \sum_{i,t}^{N,T} (y_{it}^{predicted} - y_{it}^{observed})^2},$$

is a global measure of the closeness of the model to the data that can be useful for model selection.

Figures 6.6(a) and 6.6(b) show histograms for all the predicted residuals and the predicted residuals inside the group ($r =$ "1", $w = R$), respectively, which have the shape of a Gaussian distribution with zero mean. Figure 6.6(c) contains an interaction plot of the predicted residuals in order to check the independence between pixel (i) and exposure (t) dimensions. It is neither noticed any kind of residual pixel pattern over time nor any kind of residual temporal pattern over the pixel dimension. Despite a slight trending effect of the residuals with respect to reflectance is foreseen (Figure 6.4(e)), the residuals can be considered stationary in practice, as stated in previous Section 6.7.1. Thus, it can be concluded that the residuals can be considered random and normally distributed around zero, showing a good fitting-to-data scenario.

The root mean square error of predictions results

$$\text{RMSE} = 3.55$$

units of image digital numbers, which compared to the dynamic range of the experimentation (between 0 and 1500 image digital numbers) can be concluded that the model is accurate and close to the data.

Finally, in Figure 6.7 the frequency histogram of the PIT statistics is plotted, showing clear similarities to a uniform distribution, which ensure good performance properties of the model, pointing to good and reliable approximation to the real process observed by the data.

6.9 Discussion

The present study aims to argue and show the reliability and accuracy of Bayesian modeling and inference, by its ability to define proper prior probability distributions and to infer full posterior probability distributions for the parameters of interest [Gelman et al., 2013]. This differs from and contrasts with the fixed parameter definitions and point estimates of the classical methods [Bishop, 2006, Browne et al., 2006, Raiko et al., 2006].

Furthermore, we emphasized the inherent capability of propagating uncertainty among quantities of the Bayesian approach [Brown and Prescott, 2014, Gelman et al., 2013, Gelman and Hill, 2006], in contrast to classical methods and, in particular, in

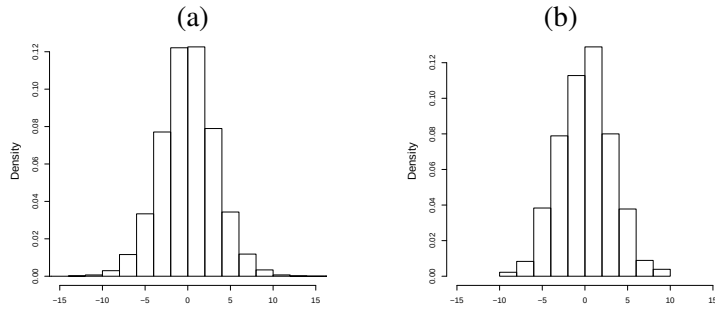


Figure 6.6: (a) Histogram of all residuals. (b) Histogram of residuals inside the group (r="1" and w=R).

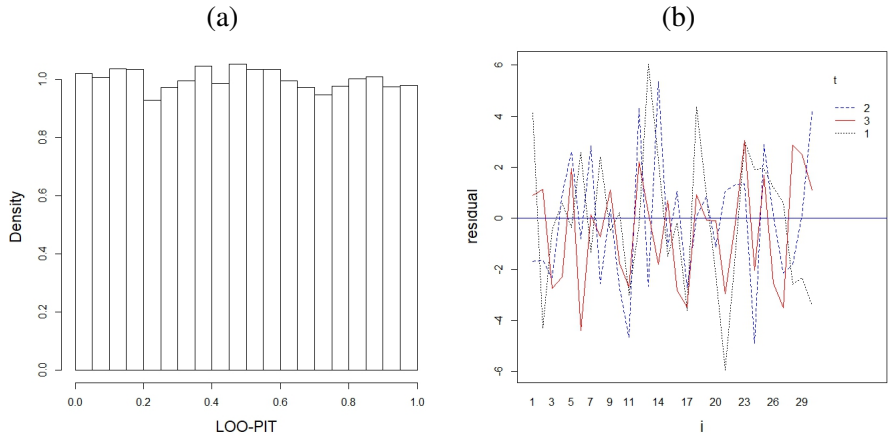


Figure 6.7: (a) Histogram of the predictive posterior checks (LOO-PIT). (b) Interaction plot between pixel (i) and exposure (t) dimensions inside a group (r="1" and w=R).

contrast to the rigidness and error propagation of nested independent point estimate computations of the commonly used Photon-transfer method for estimating sensor noise.

The Photon transfer method (Dierks [2004] and ISO [2013]) is considered as the standards for electronic noise characterization. However standard is subject to some assumptions, such as:

- Linear sensitivity (photo-response) of the sensor, i.e., the radiometric response (gray level values) increases linearly with the number of photons received.

- All noise components are stationary and white with respect to time (exposures)

and space (matrix of pixels). The parameters describing the noise component are invariant with respect to time and space.

- Only the total quantum efficiency is wavelength dependent, i.e., the effects caused by light of different wavelengths can be linearly superimposed.

If these conditions are not fulfilled, the computed parameters by the Photon-transfer method are meaningless [Tsin et al., 2001]. The Photon-transfer method is based on the photo-response noise with and without light to determine all the parameters characterizing completely the sensor radiometry. The Photon-transfer method uses the property of spatial non-uniformities of a sensor array being the same for every exposure, to remove the effect of the spatial non-uniformity by differentiating two images. If temporal non-uniformity is present in the behavior of a sensor, that is, the mean response is not stationary with respect to time, then the estimate does not represent the different photo-response among pixels. Therefore, the computed parameters by the Photon-transfer method are meaningless.

However, using statistical modeling, it is not needed to make nested independent computations but estimating all the components at once in a model. Non-linear effects for the photo-response of a sensor can be easily considered in a statistical modeling approach, by using non-linear functions in classical methods and non-parametric models in a Bayesian approach. In our work, due to the fact that we have only a few reflectances available, we have defined the parameters $\boldsymbol{\mu}_0$ as categorical factors, which also allow for modeling non-linear effects. If more reflectances were available, we could define, for example, a non-parametric prior distribution or a splines model for the parameters $\boldsymbol{\mu}_0$.

The present study is a novel attempt to model sensor noise parameters. For this reason, the data-generating model with the defining effects has been formulated as found in the general state of the art literature, where noise parameters $\boldsymbol{S}$ and $\boldsymbol{T}$ are considered completely random-structured effects and independent with respect to time (exposures) and space (matrix of pixels), respectively. However, correlated effects in space and time for the parameters $\boldsymbol{S}$ and $\boldsymbol{T}$, respectively, can be naturally considered by using Bayesian hierarchical models. Furthermore, non-stationary noise parameters, such as spatial effects varying in time or time effects varying in space, might also be feasibly considered in a Bayesian framework for the parameters $\boldsymbol{S}$ and $\boldsymbol{T}$, respectively, or for both simultaneously under some constraints.

These powerful and flexible modeling features of Bayesian hierarchical models [Coley et al., 2017, Dai et al., 2017] are promising in image sensing, opening the door to formulate new data-generating models where new effects could be investigated.

6.10 Conclusion

The formulation of a Bayesian hierarchical model based on an additive multilevel random effects model allowed us to identify the major noise components that take place in image sensing. The approach presented in this chapter provided a useful interpretation and an accurate estimation of the image sensing noise parameters. Bayesian modeling permitted a reliable definition of parameters as random effects, and by means of the MCMC method, the model is fitted in a way that all the components share information and uncertainties.

We have focused on the analysis and interpretations of the parameters $\boldsymbol{\sigma}_S$, σ_F, $\boldsymbol{\sigma}_T$, $\boldsymbol{\sigma}_P$ and σ which represent the mean noise of the parameters PRNU ($\boldsymbol{\sigma}_S$), FPN (σ_F), current noise ($\boldsymbol{\sigma}_T$), the interaction between the photon noise and PRNU ($\boldsymbol{\sigma}_P$), and amplifier, flicker and quantification noises (σ).

On the other hand, the dependency of the estimated noise parameters $\boldsymbol{\sigma}_P$, $\boldsymbol{\sigma}_T$ and $\boldsymbol{\sigma}_S$ on reflectance suggested the computation of the coefficients of variation (error and mean intensity ratio) in order to remove the linear effect of the mean level of reflectance imaged. Thus, they can be considered useful quantities to be compared among sensors as a discriminatory power of the signal.

The coefficients of variation of the noise are larger at lower reflectances than higher reflectances due to the effect of the photon noise, as can be seen in Figure 6.5(c). The photon noise effects decline, inversely proportional to the square root of the reflectance, from approximately 3% of the registered signal at very low reflectances to 0.4% at high reflectances. On the other hand, the effects of the current noise (Figure 6.5(a)) and PRNU (Figure 6.5(b)) are practically constant of approximately 1% and 0.4% of the registered signal, respectively. However, the noise effects are significantly much higher at reflectances close to zero for all noise parameters, as can be seen in Figures 6.5(a), 6.5(b) and 6.5(c). Which reveal that high image intensity values are preferred to lower image intensity values for applications such as, for example, image pattern recognition tasks.

The advantages of using Bayesian hierarchical models have been stated. Primarily, the accurate way of propagating uncertainty among quantities following probability theory rules. Secondly, the high modeling flexibility and ability to define parameters with non-linear and correlated effects and multiple factors. For example, there may be some imaging sensors that show systematic FPN patterns, instead of being completely random as the one considered here. Nevertheless, handling this issue is straightforward in the Bayesian approach provided that appropriate prior distributions with correlated effects are defined.

A brief comparison of our approach with the existing standards has been made. The assumptions of linearity and stationary of the parameters, considered by the standards, can be easily overcome by statistical modeling and, especially, using a

Bayesian approach. Furthermore, the Photon-transfer method is based on nested independent computations, with highly error propagation, in contrast to the Bayesian multilevel random-effects modeling approach presented herein.

The Bayesian multilevel random-effects modeling approach presented in this paper is a general methodology that can be applied to any other imaging sensor or camera, under different experimental, independently of the dynamic ranges. Comparison among cameras can be carried out with the proposed coefficients. In the near future, we would like to asses the assumptions of complete randomness of the current noise and the FPN. For this purpose, we will have to include appropriate prior distributions with correlated effects in the Bayesian model. Furthermore, we would like to model all the noise parameters with their exactly defining probability distributions, instead of approximating them by Normal distributions, which is an usual assumption in image sensing.

Chapter 7

Conclusion

The motivation of this work has been two-fold. On the one hand, we aimed to make use of advanced statistical models in Bayesian framework to solve three real-world applications and ultimately gain insights into these applied fields. On the other hand, we aimed to make a contribution to two methodological aspects in Bayesian GPs.

In order to solve the real-world applications, the need for some advanced modeling features arose. For example, flexible modeling to properly control or constrain the dynamics of the predicted functions were required. Models that exploit to the full the correlation structure of the data in order to make useful and accurate generalization of data, also in scenarios with a short and/or very noisy set of sampling observations, were sought. Good scaling properties in the computation of the models in the cases of large data sets were also of interest. Furthermore, hierarchical and flexible modeling to accurately decompose a stochastic signal in its different components was also sought.

Regarding the methodological contributions, we dealt with the analysis of the performance and practical implementation of a novel and recently developed low-rank approximate GP model, with the aim of recognizing the relationships among the key factors of the method, as well as making recommendations and diagnosis of the approximation. We focused on its implementation in a probabilistic programming framework and on the use of computational sampling methods. Furthermore, we dealt with the analysis of the overly smoothing effect that the use of many virtual derivative observations to induce monotonicity in functions causes on the posterior functions, especially in GP functions.

Application to rock art paintings considered in this work, aimed to predict spectrometry measurements on the surface of rock art paintings. The main objective in this application was to construct a model that fully exploits the correlation structure of the data. In the land use classification task, which aimed to perform a

spatio-temporal classification of the land-uses of parcels dedicated to growing citrus fruits, also sought a model that would exploit properly the correlation structure of the data in an scenario of a short set of sampling observations. We argued that a GP prior model with a multidimensional covariance function is one of the most natural ways to accomplish this objective for this type of data, which consists of spatio-temporal stochastic observations with covariance structure assumptions of continuity, stationarity and monotonicity. In addition, we showed the utility of using GP prior models as latent functions in non-Gaussian likelihood models such as the multinomial model for the classification of land-uses carried out in this work.

However, we argued that the spatio-temporal GP model with a square exponential covariance function was not the most appropriate model for solving the land use classification task since the class of a parcel is expected to switch arbitrarily in time. So the assumption of a smooth and monotonic covariance structure over time might not be the most appropriate or at least too simple to model the temporal structure of the process. As a line of future research, we propose modeling this application by specifying a Markov chain model for the transition probabilities of classes in time and a multivariate GP prior, with the spatial predictors, to relate the transition probabilities among parcels (space).

In application to rock art paintings, the motivation of including prior knowledge in the modeling in the form of derivative information, with the aim of inducing monotonicity and long-term stabilization to the predicted functions, also arose. We showed that models with additional derivative information have a stronger inductive bias, yielding better predictive performance and confidence intervals. However, inducing monotonicity through additional (virtual) observations arises with some practical issues. Overly smoothed posterior functions are obtained if many inducing points for monotonicity are used. This is due to the monotonicity information is included in the likelihood of the model through additional observations instead of into the prior of the function, which makes the posterior distribution of the function dependent on the number and location of the inducing points. This overly smoothing effect is even more severe using GP functions because monotonic functions do not have a characteristic lengthscale, and the value of the estimated lengthscale tends to be larger as more inducing points are used. We demonstrated that if the function is quite smooth this problem can be avoided in practice by choosing fewer inducing points and placing them appropriately.

The limitation that inference on GPs is computationally very demanding, motivated us to make a contribution to the recently developed Hilbert space approximate GP model. We analyzed in detail the performance and accuracy of the method, which ultimately lie on the relationship among the key factors: the number of basis functions, the boundary factor and the lengthscale of the function. We made recommendations for the values of these key factors based on the recognized relationships

among them, that will help users in diagnosing and improving performance. We demonstrated the applicability and the implementation of the methodology, the reduction of the computation and the improvement in sampling efficiency. The main drawback of this approach is that its computational complexity scales exponentially with the number of input dimensions. Hence, in practice, input dimensionalities larger than 3 become too computationally demanding. In these cases, we proposed using the approximate GP model as individual components in an additive modeling scheme.

Finally, application to image sensor noise consisted in decomposing the signal recorded by an image sensor into its different noise sources. Our main motivation was to propose a more flexible, hierarchical and accurate model for characterizing the noise components in image sensors, in comparison with the existing standards of image noise measurements. We argued that the Bayesian framework, by its property of defining conditional dependencies among parameters in a fully probabilistic model, allows for fully propagation of uncertainty among noise parameters, obtaining accurate and reliable estimates in flexible models. This differs from and contrasts with the fixed parameter definitions and point estimates of the classical methods used in the standards for image noise measurement.

This work leaves some lines of future research. The first one was already stated above when we proposed a mixed model, with Markov chains and GPs, to address the spatio-temporal land-use classification task. The second line of future research is to construct analytical models for the relationships among the key factors of the Hilbert space approximate GP models aiming at automatizing the diagnosis of the performance of the approximation. The third line of future research is to analyze these relationships in multidimensional cases, building useful graphs or analytical models that encode these relationships in multidimensional approximate GPs. Finally, we propose as a future research line to conduct simulation experiments to study the possible benefits of using additional information, such as function value constraints or gradient constraints, to alleviate the overly smoothing effect on the posterior functions when using many inducing points for monotonicity.

Appendix A

More case studies for the Hilbert space approximate Gaussian process method

A.1 Case study V: Same-sex marriage data

This data set relates the proportion of support for same-sex marriage to the age. The data consists of 74 observations of the amount of people y_i supporting same-sex marriage from a population n_i per age group i ($i = 1, \ldots, 74$). The observational model is a binomial model with parameters population n_i and probability of supporting same-sex marriage p_i per age group i,

$$y_i \sim \text{Binomial}(p_i, n_i).$$

The population per age group n_i is a known quantity and the goal is to estimate the same-sex support probability p_i or mean number of support people per age group. Probabilities $\boldsymbol{p} = (p_1, \ldots, p_{74})$ are modeled by a GP function $f : \mathbb{R} \to \mathbb{R}$ with a squared exponential covariance function k, as a function of age input values $\boldsymbol{x} = (x_1, \ldots, x_{74})$, and through the *logit* function as a link function,

$$\begin{aligned} p_i &= \text{logit}(f(x_i)) \\ f(x) &\sim \mathcal{GP}(0, k(x, x', \theta)). \end{aligned}$$

Saying that the function $f(\cdot)$ follows a GP model is equivalent to say that $\boldsymbol{f}$ is multivariate Gaussian distributed with covariance matrix K, where $K_{ij} = k(x_i, x_j, \theta)$, with $i, j = 1, \ldots, 74$. In the HSGP model, the function $f(x)$ is approximated as in

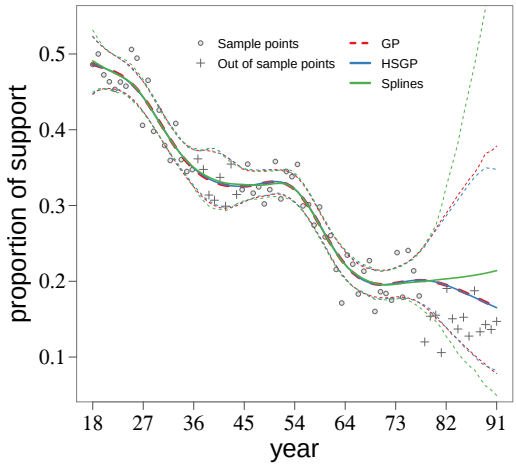


Figure A.1: Posterior mean predictive distributions of the proposed HSGP model, the regular GP model, and the Splines model. 95% credible intervals are plotted as dashed lines.

equation (3.9), with the squared exponential spectral density as in equation (3.3), and eigenvalues λ_j and eigenfunctions ϕ_j as in equations (3.7) and (3.8).

In order to do model comparison, in addition to the regular GP model and HSGP model, an splines-based model is also fitted using the Thin Plate Regression Splines approach in Wood [2003] and implemented in the R-package *mgcv* [Wood, 2015]. A Bayesian approach is used to fit this splines model using the R-package *brms* [Bürkner et al., 2017].

Figure A.1 shows the posterior mean predictive distributions of the three models, the regular GP, the HSGP model with $m = 20$ basis functions and boundary factor $c = 1.5$, and the splines model with 20 knots. Sample observations are plotted as circles in the figure, and the out-of-sample observations, which have been used for testing, are plotted as crosses.

For the HSGP model, different models with different number of basis functions and boundary factor have been fitted. The root mean square errors (RMSE) for every one of these models have been computed against the regular GP model, and plotted as a function of the number of basis functions m and boundary factor c in Figure A.2, for sample (left) and test (right) data. The expected patterns of the approximation as a function of the number of basis functions and boundary factor are recognized: as the boundary factor increases, more basis functions are needed.

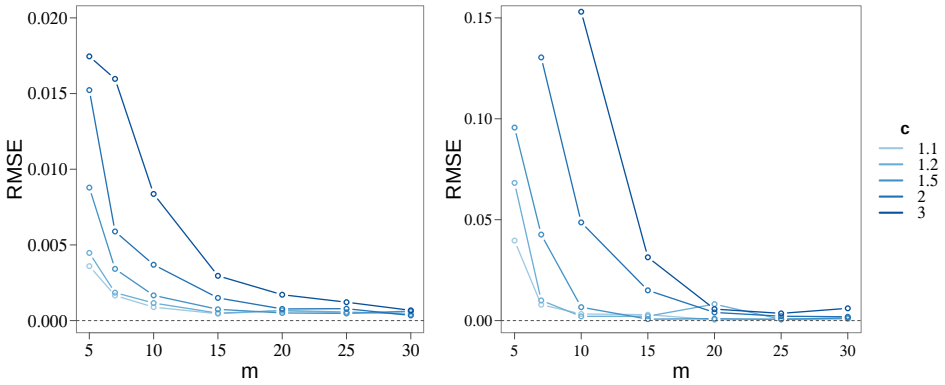


Figure A.2: Root mean square error (RMSE) of the HSGP model, computed against the regular GP model, as a function of the number of basis functions m and boundary factor c. RMSE for sample data (left) and RMSE for out-of-sample data (right).

Figure A.3 shows the RMSE of the regular GP, HSGP and splines models, computed against the actual data, for training and test data, as a function of the number of basis functions m and boundary factor c for the HSGP model, and knots for the splines model. We can see how the splines models do not extrapolate data properly.

Figure A.4 shows computational times, in seconds per iteration (iteration of the HMC sampling method), as a function of the number of basis functions m, for the HSGP model, and knots, for the splines model. The computational times is represented in the y-axis which is on a logarithmic scale. The HSGP model is on average roughly 10 times faster than the regular GP, in this particular case.

The Stan model codes for the exact GP, the approximate GP and the splines models of this case study can be found in:

`https://github.com/gabriuma/Doctoral_thesis/tree/master/Case-study-V_Same-sex-marriage`

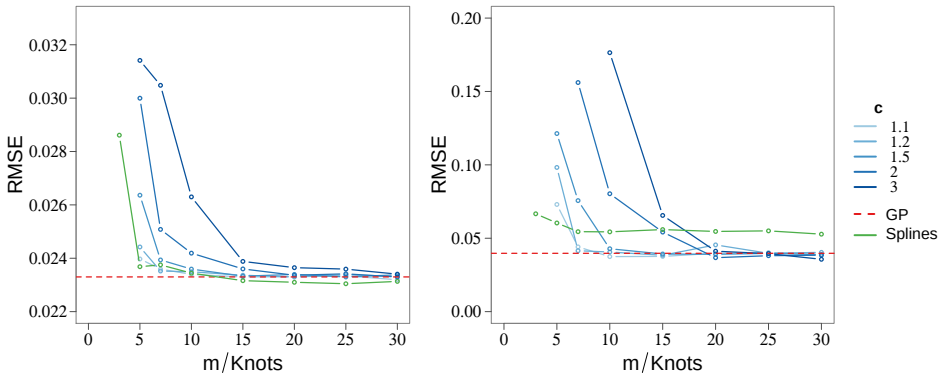


Figure A.3: Root mean square error (RMSE) of the different methods, regular GP, HSGP and splines models, computed against the actual data, as a function of the number of basis functions m and boundary factor c for the HSGP model, and knots for the splines model. RMSE for sample data (left) and RMSE for out-of-sample data (right).

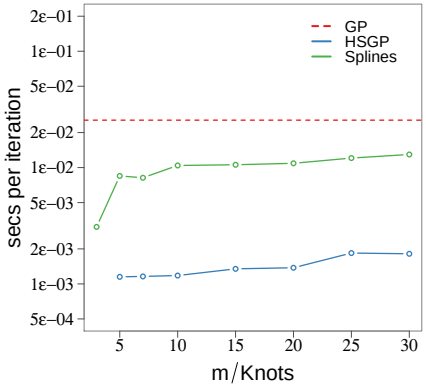


Figure A.4: Computational time (y-axis), in seconds per iteration (iteration of the HMC sampling method), as a function of the number of basis functions m and knots. The y-axis is in a logarithmic scale.

A.2 Case study VI: 2D Simulated data

This example consists of a simulated dataset with $n = 120$ ($i = 1, \ldots, n$) single draws from a Gaussian process prior with two input dimensions ($D = 2$). A squared exponential covariance function, with hyperparameters marginal variance $\alpha = 1$ and lengthscales $\ell_1 = 0.10$, for the first dimension, and $\ell_2 = 0.35$, for the second dimension, is used. The corresponding matrix of input values is $X = [\boldsymbol{x}_1 \ \boldsymbol{x}_2 \ \cdots \ \boldsymbol{x}_n]^\top \in \mathbb{R}^{n \times 2}$ with $\boldsymbol{x}_i \in \{[-1, 1], [-1, 1]\} \subset \mathbb{R}^2$. Gaussian noise $\sigma = 0.2$ was added to the GP draws to form the final noisy set of observations $\boldsymbol{y} \in \mathbb{R}^n$.

The regular GP model over the outcome variable $\boldsymbol{y}$ and matrix of inputs $X \in \mathbb{R}^{n \times 2}$, can be written as follows,

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{f} + \boldsymbol{\epsilon} \\ \boldsymbol{\epsilon} &\sim \mathcal{N}(0, \sigma^2 I) \\ f(\boldsymbol{x}) &\sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}', \theta)), \end{aligned}$$

where I represents the identity matrix, $\boldsymbol{e}$ the noise term, and $\boldsymbol{f} = \{f(\boldsymbol{x}_i)\}_{i=1}^n$ represents the underlying function values to the noisy observations $\boldsymbol{y}$. The function $f : \mathbb{R}^2 \to \mathbb{R}$ is a GP prior with a multivariate squared exponential covariance function k. Saying that the function $f(\cdot)$ follows a GP model is equivalent to say that $\boldsymbol{f}$ is multivariate Gaussian distributed with covariance matrix K, where $K_{rs} = k(\boldsymbol{x}_r, \boldsymbol{x}_s, \theta)$, with $r, s = 1, \ldots, n$.

The marginalized form, by integrating out the latent values $\boldsymbol{f}$, of the previous latent GP model results:

$$\boldsymbol{y} \sim \mathcal{N}(0, K + \sigma^2 I).$$

In the HSGP model with D input dimensions, the latent function f, evaluated at input vector $\boldsymbol{x} \in \mathbb{R}^D$, is approximated as in equation (3.13),

$$f(\boldsymbol{x}) \approx \sum_j^m \Big(S(\sqrt{\boldsymbol{\lambda}_j}) \Big)^{1/2} \phi_j(\boldsymbol{x}) \beta_j,$$

where S is the spectral density, as a function of $\sqrt{\boldsymbol{\lambda}_j}$, of the D-dimensional squared exponential covariance function,

$$S(\sqrt{\boldsymbol{\lambda}_j}) = \sigma^2 2\pi \prod_{d=1}^D \ell_d \exp\Big(-\frac{1}{2} \sum_{d=1}^D \ell_d^2 \sqrt{\lambda_{\mathbb{S}_{jd}}}^2 \Big),$$

and $\boldsymbol{\lambda}_j$ is the D-vector, which elements are the univariate eigenvalues whose indices

correspond to the elements of the D-tupple $\mathbb{S}_{j\cdot}$,

$$\boldsymbol{\lambda}_j = \left\{\lambda_{\mathbb{S}_{jd}}\right\}_{d=1}^{D} = \left\{\left(\frac{\pi \mathbb{S}_{jd}}{2L_d}\right)^2\right\}_{d=1}^{D},$$

and ϕ_j is the multivariate eigenfunction as the product of univariate eigenfunctions whose indices correspond to the elements of the D-tupple $\mathbb{S}_{j\cdot}$,

$$\phi_j(\boldsymbol{x}) = \prod_{d=1}^{D} \phi_{\mathbb{S}_{jd}} = \prod_{d=1}^{D} \sqrt{\frac{1}{L_d}} \sin\left(\sqrt{\lambda_{\mathbb{S}_{jd}}}(x_d + L_d)\right),$$

where x_d is the input value corresponding to dimension d. In the previous equations, j denotes the index for the $m = \prod_{d=1}^{D} m_d$ multivariate basis functions , where m_d is the number of basis functions considered for dimension d. $\mathbb{S}$ is the matrix of D-tuples, with rows being the indices of every possible combinations of univariate eigenvalues over the D dimensions. L_d is the boundary for the dimension d. The parameters β_j are $\mathcal{N}(0, 1)$ distributed, and α and ℓ_d are the marginal variance and lengthscale of dimension d, respectively, of the approximate multivariate covariance function.

In order to do model comparison, in addition to the regular GP and HSGP models, a two-dimensional splines-based model is also fitted using a cubic spline basis, penalized by the conventional integrated square second derivative cubic spline penalty [Wood, 2017], and implemented in the R-package *mgcv* [Wood, 2015]. A Bayesian approach is used to fit this spline model using the R-package *brms* [Bürkner et al., 2017].

Figure A.5 shows the data-generating GP function, from where the dataset was drawn, and the mean posterior predictive functions of the three models, the regular GP, the HSGP, and the splines, fitted over the dataset. Sample observations are also plotted in the plots as circles. For the HSGP model, $m_1 = 40$ and $m_2 = 40$ basis functions for each dimension respectively, were used, which lead to a total of 1600 multivariate basis functions. A boundary factor for each dimension $c_1 = 1.5$ and $c_2 = 1.5$ were used. For the splines model, 40 knots in each dimension were used.

Figure A.6 shows the difference functions between the data-generating function and the GP, HSGP and splines models, respectively.

In order to assess performance of the models as a function of the number of basis functions and knots, different models with different number of basis functions, for the HSGP model, and different number of knots, for the splines model, have been fitted. In all models, the same number of basis functions and knots per dimension were used. Figure A.7-(left) shows the root mean squared error (RMSE), computed against the data-generating function, as a function of the boundary factor c, and the

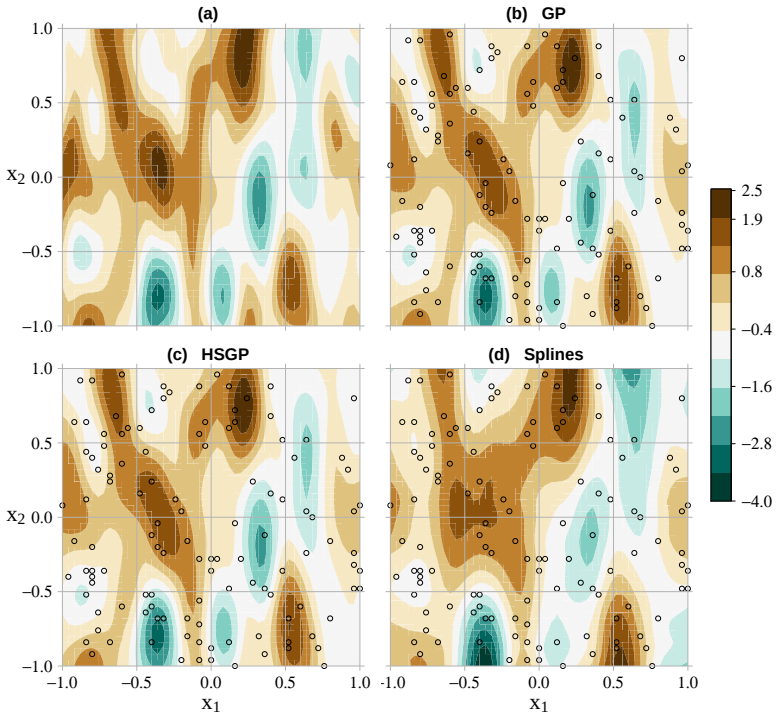


Figure A.5: (a) Data-generating function. (b) Mean posterior predictive function of the GP model. (c) Mean posterior predictive function of the HSGP model. (d) Mean posterior predictive function of the splines model. Sample points are plotted as circles

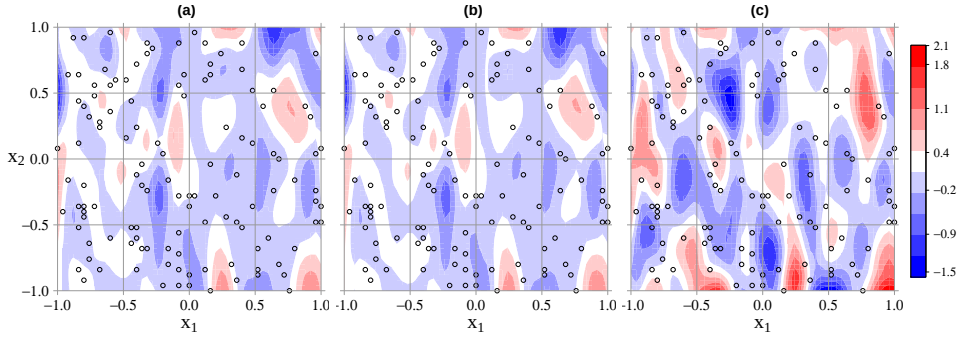


Figure A.6: Mean error between the data-generating function and the GP (a), HSGP (b) and splines (c) models. Sample points are plotted as circles.

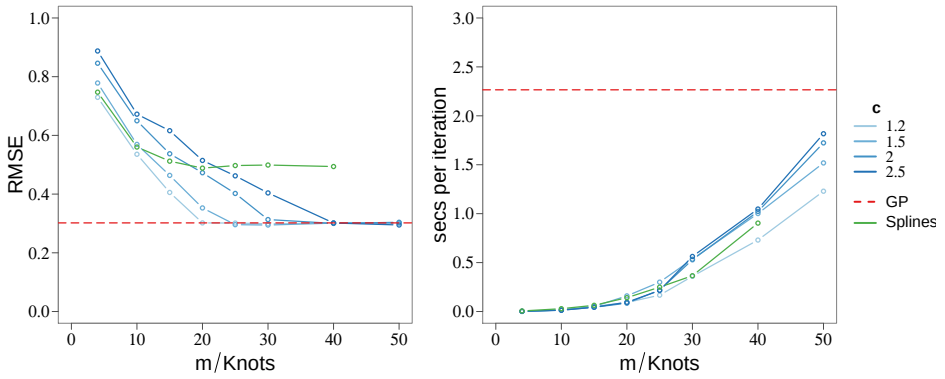


Figure A.7: Root mean square error (RMSE) (left) and computational time (right) in seconds per iteration (iteration of the HMC sampling method) of the different methods computed against the data-generating function, as a function of the boundary factor c, number of basis functions m and knots.

number of univariate basis functions m, for the HSGP model, and knots, for the splines model. From Figures A.6 and A.7-(left), it can be seen a close approximation of the HSGP model to the regular GP model. However, the performance of the splines model is significantly worse. Figure A.7-(right) shows the computational times of the different models as a function of the boundary factor, number of basis functions and knots. Figure A.7 reveals that choosing the optimal boundary factor allows for less number of basis functions and less computational time.

The Stan model codes for the exact GP, the approximate GP and the splines models of this case study can be found in:

`https://github.com/gabriuma/Doctoral_thesis/tree/master/Case-study-VI_Simulated-data_2D`

A.3 Case study VII: Leukemia data

The next example presents a survival analysis in acute myeloid leukemia (AML) in adults, with data recorded between 1982 and 1998 in the North West Leukemia Register in the United Kingdom. The data set consist in survival times t_i and censoring indicator z_i (0 for observed and 1 for censored) for $n = 1043$ cases ($i = 1, \ldots, n$). Some 16% of cases were censored. Predictors are *age* ($\boldsymbol{x}_1$), *sex* ($\boldsymbol{x}_2$), *white blood cell* (WBC) ($\boldsymbol{x}_3$) count at diagnosis with 1 unit = $50 \times 109/L$, and

the *Townsend deprivation index* (TDI) ($\boldsymbol{x}_4$) which is a measure of deprivation for district of residence. We denote the matrix $X = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ \boldsymbol{x}_3\ \boldsymbol{x}_4]^\top \in \mathbb{R}^{n\times 4}$ which contains the predictors.

As the WBC measurements were strictly positive and highly skewed, we fit the model to its logarithm. Continuous predictors were normalized to have zero mean and unit standard deviation. Survival time was normalized to have zero mean for the logarithm of time. We assume a log Gaussian observation model for the observed survival time, t_i, with a function of the predictors, $f(X_i)$, as the location parameter, and σ as the Gaussian noise:

$$p(t_i) = \text{LogNormal}(t_i|f(X_i), \sigma^2)$$

with $X_i = \{x_{i1}, x_{i2}, x_{i3}, x_{i4}\} \in \mathbb{R}^4$ and $f : \mathbb{R} \to \mathbb{R}$.

As we do not have a model for the censoring process, we do not have a full observation model, and the observational model for the censored data t_i is assumed to be the complementary cumulative normal probability distribution:

$$p(y_i > t_i) = \int_{t_i}^{\infty} \text{LogNormal}(y_i|f(X_i), \sigma^2)dy_i = 1 - \Phi\left(\frac{\log(y_i) - f(X_i)}{\sigma}\right),$$

where y_i denotes the uncensored time.

The latent function $f(\cdot)$ is modeled as a Gaussian process, centered in a linear model of the predictors X, and with a squared exponential covariance function k depending on predictors X and hyperparameters $\theta = (\alpha, \ell)$,

$$f(\boldsymbol{x}) \sim \mathcal{GP}(c + \boldsymbol{\beta}\boldsymbol{x}, k(\boldsymbol{x}, \boldsymbol{x}', \theta)),$$

where c and $\boldsymbol{\beta}$ are the intercept and vector of coefficients, respectively, of the linear model. Saying that the function $f(\cdot)$ follows a GP model is equivalent to say that $\boldsymbol{f}$ are multivariate Gaussian distributed with mean function $\mu(\cdot)$ and covariance matrix K, where $\mu(\boldsymbol{x}_i) = c + \boldsymbol{\beta}\boldsymbol{x}_i$ and $K_{rs} = k(\boldsymbol{x}_r, \boldsymbol{x}_s, \theta)$, with $r, s = 1, \ldots, n$. The hyperparameters α and ℓ represent the marginal variance and lengthscale, respectively, of the GP process. Notice that a scalar lengthscale is considered in the multivariate covariance function.

Due to the predictor *sex* ($\boldsymbol{x}_2$) is a categorical variable (1 for female and 2 for male), we can outline a multilevel model for the GP function, in a similar way like categorical effects are treated in linear models. The relative contribution of a GP function given one of the levels of the predictor (equation A.2) to a general mean GP function (equation A.1) is defined. For the other level of the predictor, the GP function effects are set to zero. This multilevel construction is depicted as

following:

$$f(\boldsymbol{x}) \sim \mathcal{GP}(c + \boldsymbol{\beta}\boldsymbol{x}, k(\boldsymbol{x}, \boldsymbol{x}', \theta_1)) \tag{A.1}$$

$$g(\boldsymbol{x}) \sim \mathcal{GP}(0, k(\boldsymbol{x}, \boldsymbol{x}', \theta_2 | \boldsymbol{x}_2 = 2)) \tag{A.2}$$

$$f(\boldsymbol{x}|\boldsymbol{x}_2 = 2) = f(X|\boldsymbol{x}_2 = 2) + g(\boldsymbol{x})$$

In the previous equations, θ_1 contains the hyperparameters α_1 and ℓ_1 which are the marginal variance and lengthscale, respectively, of the general mean GP function, and θ_2 contains the hyperparameters α_2 and ℓ_2 which are the marginal variance and lengthscale, respectively, of the GP function restricted to the male sex ($\boldsymbol{x}_2 = 2$).

Using the HSGP approximation, the functions $f(\boldsymbol{x})$ and $g(\boldsymbol{x}|x_2 = 2)$ are approximated as in equation (3.13), with the D-dimensional (with a scalar lengthscale) squared exponential spectral density S as in equation (3.3) on Chapter 3.4, and the multivariate eigenfunctions ϕ_j and the D-vector of eigenvalues $\boldsymbol{\lambda}_j$ as in equations (3.11) and (3.12), respectively.

Figure A.8 shows estimated conditional comparison of each predictor with all others fixed to their mean values. These posterior estimates correspond to the HSGP model with $m = 10$ basis functions and $c = 3$ boundary factor. The model has found smooth non-linear patterns and the right bottom subplot also shows that the conditional comparison associated with WBC has an interaction with TDI.

Figure A.9 shows the root mean square error (RMSE) computed against the regular GP, and the time of computation as a function of the number of univariate basis functions m and boundary factor c. As the functions are smooth, a few number of basis functions and a large boundary factor are required to obtain a good approximation (Figure A.9-right); Small boundary factors are not allowed when large lengthscales as can be seen in Figure 3.6. Increasing the boundary factor also significantly increases the time of computation (Figure A.9-left).

The Stan model codes for the exact GP and the approximate GP models of this case study can be found in:

`https://github.com/gabriuma/Doctoral_thesis/tree/master/Case-study-VII_Leukemia-data`

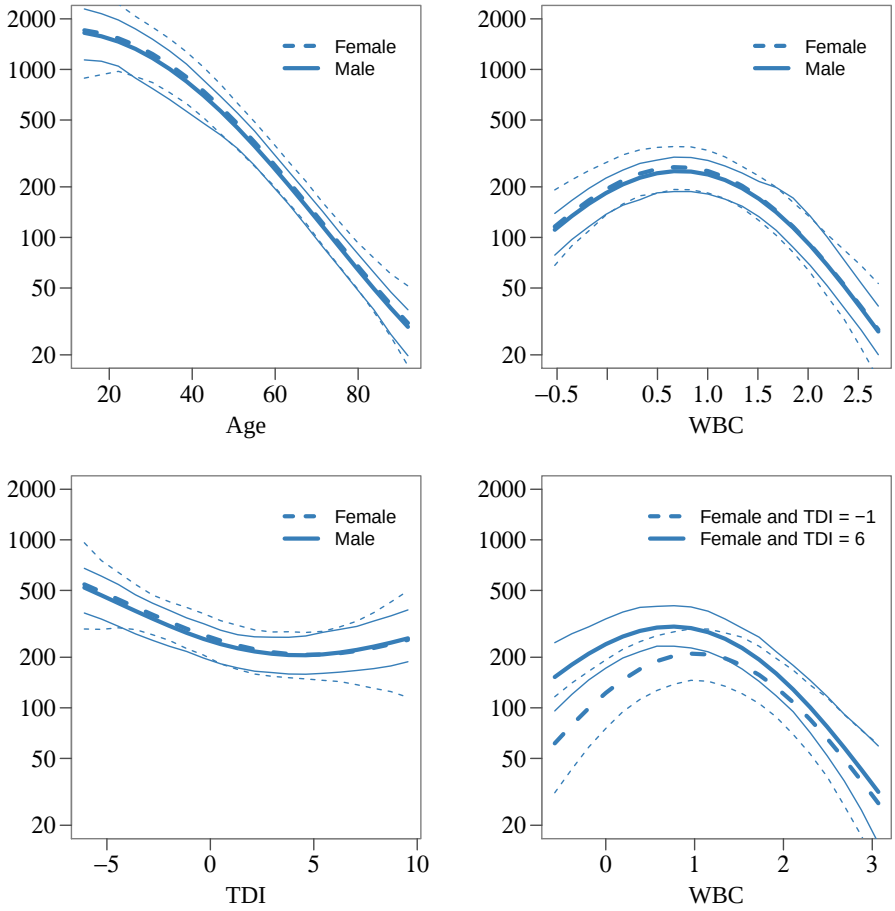


Figure A.8: Expected lifetime conditional comparison for each predictor with other predictors fixed to their mean values. The thick line in each graph is the posterior mean estimated using a HSGP model, and the thin lines represent pointwise 95% credible intervals.

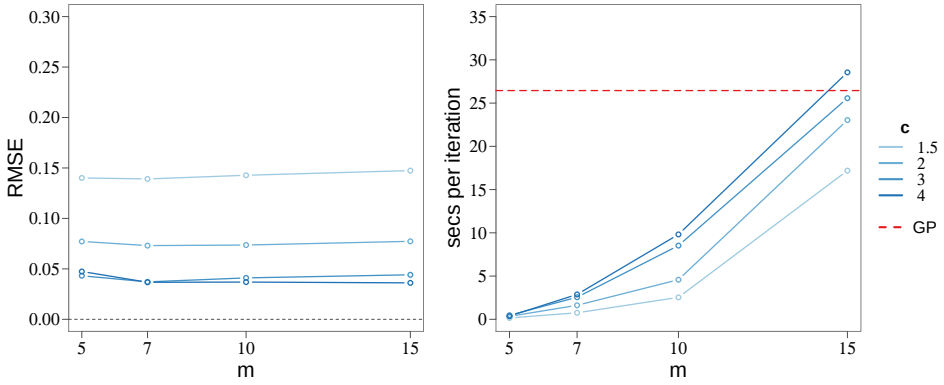


Figure A.9: Root mean square error (RMSE) computed against the GP and time of computation in seconds per iteration (iteration of the HMC sampling method) as a function of the number of basis functions m and boundary factor c.

Appendix B

Approximation of the covariance function using Hilbert space methods

In this section, we briefly present a summary of the mathematical details of the approximation of a stationary covariance function as a series expansion of eigenvalues and eigenfunctions of the Laplacian operator. This statement is basically an extract of the work Solin and Särkkä [2018], where the authors fully develop the mathematical theory behind the Hilbert Space approximation for stationary covariance functions.

Associated to each covariance function $k(\boldsymbol{x}, \boldsymbol{x}')$ we can also define a covariance operator $\mathcal{K}$ over a function $f(\boldsymbol{x})$ as follows:

$$\mathcal{K}f(\boldsymbol{x}) = \int k(\boldsymbol{x}, \boldsymbol{x}')f(\boldsymbol{x}')d\boldsymbol{x}'.$$

From the Bochner's and Wiener-Khintchine theorem, the spectral density of a stationary covariance function $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{\tau})$, $\boldsymbol{\tau} = (\boldsymbol{x} - \boldsymbol{x}')$, is the Fourier transform of the covariance function,

$$S(\boldsymbol{w}) = \int k(\boldsymbol{\tau})e^{-2\pi i \boldsymbol{w}\cdot\boldsymbol{\tau}},$$

where $\boldsymbol{w}$ is in the frequency domain. The operator $\mathcal{K}$ will be translation invariant if the covariance function is stationary. This allows for a Fourier representation of the operator $\mathcal{K}$ as a transfer function which is the spectral density of the Gaussian process. Thus, the spectral density $S(\boldsymbol{w})$ also gives the approximate eigenvalues of

the operator $\mathcal{K}$.

In the isotropic case $S(\boldsymbol{w}) = S(||\boldsymbol{w}||)$ and assuming that the spectral density function $S(\cdot)$ is regular enough, then it can be represented as a polynomial expansion:

$$S(||\boldsymbol{w}||) = a_0 + a_1||\boldsymbol{w}||^2 + a_2(||\boldsymbol{w}||^2)^2 + a_3(||\boldsymbol{w}||^2)^3 + \cdots . \tag{B.1}$$

The Fourier transform of the Laplace operator ∇^2 is $-||\boldsymbol{w}||$, thus the Fourier transform of $S(||\boldsymbol{w}||)$ is

$$\mathcal{K} = a_0 + a_1(-\nabla^2) + a_2(-\nabla^2)^2 + a_3(-\nabla^2)^3 + \cdots , \tag{B.2}$$

defining a pseudo-differential operator as a series of Laplace operators.

If the negative Laplace operator $-\nabla^2$ is defined as the covariance operator of the formal kernel l,

$$-\nabla^2 f(\boldsymbol{x}) = \int l(\boldsymbol{x}, \boldsymbol{x}')f(\boldsymbol{x}')d\boldsymbol{x}',$$

then the formal kernel can be represented as

$$l(\boldsymbol{x}, \boldsymbol{x}') = \sum_j \lambda_j \phi_j(\boldsymbol{x})\phi_j(\boldsymbol{x}'),$$

where $\{\lambda_j\}_{j=1}^{\infty}$ and $\{\phi_j(\boldsymbol{x})\}_{j=1}^{\infty}$ are the set of eigenvalues and eigenvectors, respectively, of the Laplacian operator. Namely, they satisfy the following eigenvalue problem in the compact subset $\boldsymbol{x} \in \Omega \subset \mathbb{R}^D$ and with the Dirichlet boundary condition (another boundary condition could be used as well):

$$\begin{aligned} -\nabla^2\phi_j(\boldsymbol{x}) &= \lambda\phi_j(\boldsymbol{x}), & x \in \Omega \\ \phi_j(\boldsymbol{x}) &= 0, & x \notin \Omega. \end{aligned}$$

Because $-\nabla^2$ is a positive definite Hermitian operator, the set of eigenfunctions $\phi_j(\cdot)$ are orthonormal with respect to the inner product

$$< f, g >= \int_\Omega f(\boldsymbol{x})g(\boldsymbol{x})d(\boldsymbol{x})$$

that is,

$$\int_\Omega \phi_i(\boldsymbol{x})\phi_j(\boldsymbol{x})d(\boldsymbol{x}) = \delta_{ij},$$

and all the eigenvalues λ_j are real and positive.

Due to normality of the basis of the representation of the formal kernel $l(\boldsymbol{x}, \boldsymbol{x}')$, its formal powers $s = 1, 2, \dots$ can be write as

$$l(\boldsymbol{x}, \boldsymbol{x}')^s = \sum_j \lambda_j^s \phi_j(\boldsymbol{x}) \phi_j(\boldsymbol{x}'), \tag{B.3}$$

which are again to be interpreted to mean that

$$(-\nabla^2)^s f(\boldsymbol{x}) = \int l^s(\boldsymbol{x}, \boldsymbol{x}') f(\boldsymbol{x}') d\boldsymbol{x}'.$$

This implies that we also have

$$[a_0 + a_1(-\nabla^2) + a_2(-\nabla^2)^2 + \cdots] f(\boldsymbol{x}) = \int [a_0 + a_1 l^1(\boldsymbol{x}, \boldsymbol{x}') + a_2 l^2(\boldsymbol{x}, \boldsymbol{x}') + \cdots] f(\boldsymbol{x}') d\boldsymbol{x}'.$$

Then, looking at Equations (B.2) and (B.3), it can be concluded

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_j [a_0 + a_1 \lambda_j^1 + a_2 \lambda_j^2 + \cdots] \phi_j(\boldsymbol{x}) \phi_j(\boldsymbol{x}'). \tag{B.4}$$

By letting $||\boldsymbol{w}||^2 = \lambda_j$ the spectral density in Equation (B.1) becomes

$$S(\sqrt{\lambda_j}) = a_0 + a_1 \lambda_j + a_2 \lambda_j^2 + a_3 \lambda_j^3 + \cdots,$$

and substituting in Equation (B.4) then leads to the final searched approximation

$$k(\boldsymbol{x}, \boldsymbol{x}') = \sum_j S(\sqrt{\lambda_j}) \phi_j(\boldsymbol{x}) \phi_j(\boldsymbol{x}'), \tag{B.5}$$

where $S(\cdot)$ is the spectral density of the covariance function, λ_j is the jth eigenvalue and $\phi_j(\cdot)$ the eigenfunction of the Laplace operator in a given domain.

References

Abramowitz, M. and Stegun, I. (1970). *Handbook of Mathematical Functions*. Dover Publications, Inc., New York.

Adler, R. J. (1981). *The geometry of random fields*, volume 62. SIAM.

Aguerrebere, C., Delon, J., Gousseau, Y., and Musé, P. Study of the digital camera acquisition process and statistical modeling of the sensor raw data. *https://hal.archives-ouvertes.fr/hal-00733538* .

Aguilera-Morillo, M. C., Durbán, M., and Aguilera, A. M. (2017). Prediction of functional data with spatial dependence: a penalized approach. *Stochastic Environmental Research and Risk Assessment* **31,** 7–22.

Akhiezer, N. and Glazman, I. (1993). Theory of linear operators in Hilbert space (Ungar, New York, 1963). *Vol. II,* pages 121–126.

Aldrich, J. H., Nelson, F. D., and Adler, E. S. (1984). *Linear probability, logit, and probit models*. Number 45. Sage.

Andersen, M. R., Magnusson, M., Jonasson, J., and Vehtari, A. (2019). Bayesian leave-one-out cross-validation for large data. In *Thirty-sixth International Conference on Machine Learning*.

Andersen, M. R., Siivola, E., Riutort-Mayol, G., and Vehtari, A. (2018). A non-parametric probabilistic model for monotonic functions. *https://sites.google.com/view/nipsbnp2018/accepted-papers* .

Andersen, M. R., Siivola, E., and Vehtari, A. (2017). Bayesian optimization of unimodal functions. *31st Conference on Neural Information Processing Systems (NIPS 2017)* Long Beach, CA, USA.

Arthur, D. (1979). C.T.H. Baker, The numerical treatment of integral equations (Clarendon Press; Oxford University Press, 1978). *https://doi.org/10.1017/S0013091500027863* .

Baladandayuthapani, V., Mallick, B. K., Young Hong, M., Lupton, J. R., Turner, N. D., and Carroll, R. J. (2008). Bayesian hierarchical spatially correlated functional data analysis with application to colon carcinogenesis. *Biometrics* **64,** 64–73.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2014). *Hierarchical modeling and analysis for spatial data*. Chapman and Hall/CRC.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008). Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70,** 825–848.

Barlow, R. E., Bartholomew, D. J., Bremner, J. M., and Brunk, H. D. (1972). *Statistical inference under order restrictions: The theory and application of isotonic regression*. Wiley New York.

Bayarri, M. J. and Berger, J. O. (2000). P values for composite null models. *Journal of the American Statistical Association* **95,** 1127–1142.

Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical transactions of the Royal Society of London* pages 370–418.

Beal, M. J. et al. (2003). *Variational algorithms for approximate Bayesian inference*. PhD thesis, University of London.

Bernardo, J. M. and Smith, A. F. (2009). *Bayesian theory*, volume 405. John Wiley & Sons.

Betancourt, M. (2016). Identifying the optimal integration time in hamiltonian monte carlo. *arXiv preprint arXiv:1601.00225* .

Bickel, P. and Lehmann, E. (2012). Frequentist interpretation of probability. In *Selected Works of EL Lehmann*, pages 1083–1085. Springer.

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

Brezger, A. and Steiner, W. J. (2008). Monotonic regression based on bayesian p–splines: An application to estimating price response functions from store-level scanner data. *Journal of business & economic statistics* **26,** 90–104.

Briol, F. X., Oates, C., Girolami, M., Osborne, M. A., Sejdinovic, D., et al. (2015). Probabilistic integration: A role in statistical computation? *arXiv preprint arXiv:1512.00933* .

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov chain Monte Carlo*. CRC press.

Brown, H. and Prescott, R. (2014). *Applied mixed models in medicine*. John Wiley & Sons.

Browne, W. J., Draper, D., et al. (2006). A comparison of bayesian and likelihood-based methods for fitting multilevel models. *Bayesian Analysis* **1,** 473–514.

Brumback, B. A., Ruppert, D., and Wand, M. P. (1999). Variable selection and function estimation in additive nonparametric regression using a data-based prior: Comment. *Journal of the American Statistical Association* **94,** 794–797.

Bui, T. D. and Turner, R. E. (2014). Tree-structured gaussian process approximations. In *Advances in Neural Information Processing Systems*, pages 2213–2221.

Bui, T. D., Yan, J., and Turner, R. E. (2017). A unifying framework for gaussian process pseudo-point approximations using power expectation propagation. *The Journal of Machine Learning Research* **18,** 3649–3720.

Bürkner, P.-C. et al. (2017). brms: An r package for bayesian multilevel models using stan. *Journal of Statistical Software* **80,** 1–28.

Burt, D., Rasmussen, C. E., and van der Wilk, M. (2019). Explicit rates of convergence for sparse variational inference in Gaussian process regression. *arXiv preprint arXiv:1903.03571* .

Campos, J. (2000). Radiometric calibration of charge-coupled-device video cameras. *Metrologia* **37,** 459–464.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software* **76,** 1649–1660.

Cassar, M., Brimblecombe, P., and Nixon, T. (2001). Technological requirements for solutions in the conservation and protection of historic monuments and archaeological remains. *Working paper for the STOA Unit* .

CIE (2004). *CIE 015:2004, Colorimetry*. Commission Internationale de l'Éclairage, 3rd edition.

Coley, R. Y., Fisher, A. J., Mamawala, M., Carter, H. B., Pienta, K. J., and Zeger, S. L. (2017). A bayesian hierarchical model for prediction of latent health states from multiple data sources with application to active surveillance of prostate cancer. *Biometrics* **73,** 625–634.

Columbia, M. R., D., S. G., C., H., and Messier, P. (2013). The application of microfadeometric testing to mounted photographs at the indianapolis museum of art. *Microscopy and Microanalysis* **19,** 1412–1413.

Cox, R. T. (1946). Probability, frequency and reasonable expectation. *American journal of physics* **14,** 1–13.

Crainiceanu, C. M., Ruppert, D., and Wand, M. P. (2005). Bayesian analysis for penalized spline regression using winbugs. *Journal of Statistical Software* **14,** 1–24.

Cramér, H. and Leadbetter, M. R. (2013). *Stationary and related stochastic processes: Sample function properties and their applications*. Courier Corporation.

Cressie, N. and Huang, H.-C. (1999). Classes of nonseparable, spatio-temporal stationary covariance functions. *Journal of the American Statistical Association* **94,** 1330–1339.

Csató, L., Fokoué, E., Opper, M., Schottky, B., and Winther, O. (2000). Efficient approaches to gaussian process classification. In *Advances in Neural Information Processing Systems*, pages 251–257.

Cseke, B. and Heskes, T. (2011). Approximate marginals in latent gaussian models. *Journal of Machine Learning Research* **12,** 417–454.

Cunningham, J. P., Shenoy, K. V., and Sahani, M. (2008). Fast gaussian process methods for point process intensity estimation. In *Proceedings of the 25th International Conference on Machine Learning*, pages 192–199. ACM.

Currie, I. D., Durban, M., and Eilers, P. H. (2006). Generalized linear array models with applications to multidimensional smoothing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68,** 259–280.

Dai, T., Guo, Y., Initiative, A. D. N., et al. (2017). Predicting individual brain functional connectivity using a bayesian hierarchical model. *NeuroImage* **147,** 772–787.

De Finetti, B. (2017). *Theory of probability: A critical introductory treatment*, volume 6. John Wiley & Sons.

De Iaco, S., Myers, D. E., and Posa, D. (2002). Nonseparable space-time covariance models: some parametric families. *Mathematical Geology* **34,** 23–42.

De-Jiang, W. and Tao, Z. (2011). Noise analysis and measurement of time delay and integration charge coupled device. *Chinese Physics B* **20,** 087202.

Deisenroth, M. P., Fox, D., and Rasmussen, C. E. (2015). Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **37,** 408–423.

del Hoyo-Meléndez, J. M., Lerma, J. L., López-Montalvo, E., and Villaverde, V. (2015). Documenting the light sensitivity of spanish levantine rock art paintings. *ISPRS Ann Photogramm Remote Sens Spat Inf Sci* **II,** 53–59.

del Hoyo-Meléndez, J. M. and Mecklenburg, M. F. (2010). A survey on the light-fastness properties of organic-based alaska native artifacts. *J Cult Herit.* **11,** 493–499.

Delicado, P., Giraldo, R., Comas, C., and Mateu, J. (2010). Statistics for spatial functional data: some recent contributions. *Environmetrics: The official journal of the International Environmetrics Society* **21,** 224–239.

Delicado, P., Giraldo, R., and Mateu, J. (2008). Point-wise kriging for spatial prediction of functional data. In *Functional and Operatorial Statistics*, pages 135–141. Springer.

Dierks, F. (2004). Sensitivity and image quality of digital cameras. Technical report, BASLER Vison technologies.

Díez-Herrero, A., Gutiérrez-Pérez, I., and Lario, J. (2009). Analysis of potential direct insolation as a degradation factor of cave paintings in Villar del Humo, Cuenca, Central Spain. *Geoarchaeology* **24,** 450–465.

Diggle, P. J. (2013). *Statistical analysis of spatial and spatio-temporal point patterns*. Chapman and Hall/CRC.

Dong, L., Zhou, J., and Tang, Y. Y. (2018). Effective and fast estimation for image sensor noise via constrained weighted least squares. *IEEE Transactions on Image Processing* **27,** 2715–2730.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid monte carlo. *Physics letters B* **195,** 216–222.

Durrande, N., Ginsbourger, D., and Roustant, O. (2012). Additive covariance kernels for high-dimensional gaussian process modeling. In *Annales de la Faculté des sciences de Toulouse: Mathématiques*, volume 21, pages 481–499.

Duvenaud, D., Lloyd, J. R., Grosse, R., Tenenbaum, J. B., and Ghahramani, Z. (2013). Structure discovery in nonparametric regression through compositional kernel search. *arXiv preprint arXiv:1302.4922* .

Duvenaud, D. K., Nickisch, H., and Rasmussen, C. E. (2011). Additive Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 226–234.

El Gamal, A., Fowler, B. A., Min, H., and Liu, X. (1998). Modeling and estimation of FPN components in CMOS image sensors. In *Proceedings SPIE 3301, Solid State Sensor Arrays: Development and Applications II*, volume 3301, pages 168–177. Society of Photographic Instrumentation Engineers.

EMVA (2010). Standard 1288, Standard for characterization of image sensors and cameras. Technical report, European Machine Vision Association.

Feller, R. L., Johnston-Feller, R. M., and Bailie, C. (1986). Determination of the specific rate constant for the loss of a yellow intermediate during the fading of alizarin lake. *Journal of the American Institute for Conservation* **25,** 65–72.

Feller, W. (2008). *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons.

Ford, B. (2011). Non-destructive microfade testing at the national museum of Australia. *AICCM Bull.* **32,** 54–64.

Ford, B. and Druzik, J. (2013). Microfading: the state of the art for natural history collections. *Collect. forum* **27,** 54–71.

Furrer, E. M. and Nychka, D. W. (2007). A framework to understand the asymptotic properties of kriging and splines. *Journal of the Korean Statistical Society* **36,** 57–76.

Gal, Y. and Turner, R. (2015). Improving the gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs. In *International Conference on Machine Learning*, pages 655–664.

Gelfand, A. E., Dey, D. K., and Chang, H. (1992). Model determination using predictive distributions with implementation via sampling-based methods. Technical report, Stanford University Deptartment of Statistics, California.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013). *Bayesian data analysis*. FL: CRC press, 3 edition.

Gelman, A. et al. (2006). Prior distributions for variance parameters in hierarchical models (comment on article by browne and draper). *Bayesian Analysis* **1,** 515–534.

Gelman, A. and Hill, J. (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press.

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science* pages 457–472.

Geman, S. and Geman, D. (1993). Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *Journal of Applied Statistics* **20,** 25–62.

Gershman, S. J., Hoffman, M. D., and Blei, D. M. (2012). Nonparametric variational inference. In *Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland*.

Geweke, J. (1989). Bayesian inference in econometric models using monte carlo integration. *Econometrica: Journal of the Econometric Society,* pages 1317–1339.

Gibbs, M. N. and MacKay, D. J. (2000). Variational gaussian process classifiers. *IEEE Transactions on Neural Networks* **11,** 1458–1464.

Giles, C. H. (1965). The fading of colouring matters. *J. Appl. Chem.* **15,** 541–550.

Giles, C. H., Johari, D. P., and Shah, C. D. (1968). Some observations on the kinetics of dye fading. *Textile Research Journal* **38,** 1048–1056.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. (1995). *Markov chain Monte Carlo in practice*. Chapman and Hall/CRC.

Giraldo, R., Delicado, P., and Mateu, J. (2010). Continuous time-varying kriging for spatial prediction of functional data: an environmental application. *Journal of Agricultural, Biological, and Environmental Statistics* **15,** 66–82.

Gow, R. D., Renshaw, D., Findlater, K., Grant, L., McLeod, S. J., Hart, J., and Nicol, R. L. (2007). A comprehensive tool for modeling CMOS image-sensor-noise performance. *IEEE Transactions on Electron Devices* **54,** 1321–1329.

GPy (2012). GPy: A Gaussian process framework in python. http://github.com/SheffieldML/GPy.

Granados, M., Ajdin, B., Wand, M., Theobalt, C., Seidel, H.-P., and Lensch, H. P. (2010). Optimal hdr reconstruction with linear digital cameras. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 215–222. IEEE.

Grant, L. (2005). Characterization of noise sources in CMOS image sensors. In *2005 IEEE International solid-state circutis conference*. Institute of Electrical and Electronics Engineers.

Han, B. Y., Shang, Y. Y., Zhao, X. X., and Liu, H. (2011). Research on noise sources in CMOS image sensors. In *Advanced Materials Research*, volume 159, pages 527–531. Trans Tech Publ.

Hastie, T. J. (2017). Generalized additive models. In Routledge, editor, *Statistical models in S*, pages 249–307. Routledge.

Hastings, W. K. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika* **57,** 97–109.

Healey, G. E. and Kondepudy, R. (1994). Radiometric CCD camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **16,** 267–276.

Hennig, P., Osborne, M. A., and Girolami, M. (2015). Probabilistic numerics and uncertainty in computations. *Proc. R. Soc. A* **471,** 20150142.

Hensman, J., Durrande, N., and Solin, A. (2017). Variational Fourier features for Gaussian processes. *The Journal of Machine Learning Research* **18,** 5537–5588.

Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research* **15,** 1593–1623.

ISO (2013). International standard 14739:2003. photography-electronic still-picture imaging - noise measurements. Technical report, International Organization for Standardization.

Janesick, J., Elliott, T., Collins, S., Blouke, M., and Freeman, J. (1987). Scientific charge-coupled devices. *Opticl Engineering* **26,** 692–714.

Jaynes, E. T. (1985). Bayesian methods: general background. *Maximum Entropy and Bayesian Methods in Applied Statistics* pages 1–25.

Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge University Press.

Jeffreys, H. (1961). Theory of probability, 3rd edn Oxford: Oxford university press.

Johnston-Feller, R., Feller, R. L., Bailie, C. W., and Curran, M. (1984). The kinetics of fading: opaque paint films pigmented with alizarin lake and titanium dioxide. *Journal of the American Institute for Conservation* **23,** 114–129.

Kass, R. E. and Wasserman, L. (1995). A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion. *Journal of the American Statistical Association* **90,** 928–934.

Kimeldorf, G. S. and Wahba, G. (1970). A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics* **41,** 495–502.

Kneib, T. and Fahrmeir, L. (2006). Structured additive regression for categorical space–time data: A mixed model approach. *Biometrics* **62,** 109–118.

Kuroda, T. (2014). *Essential principles of image sensors*. CRC press.

Larsen, K. (2015). GAM: the predictive modeling silver bullet. *Multithreaded. Stitch Fix* **30,**.

Lázaro Gredilla, M. (2010). *Sparse Gaussian processes for large-scale machine learning*. PhD thesis.

Lee, D.-J. and Durbán, M. (2011). P-spline anova-type interaction models for spatio-temporal smoothing. *Statistical Modelling* **11,** 49–69.

Li, C., Rana, S., Gupta, S., Nguyen, V., and Venkatesh, S. (2017). Bayesian optimization with monotonicity information. In *Workshop on Bayesian Optimization at Neural Information Processing Systems (NIPSW)*.

Little, R. J. and Rubin, D. B. (2002). *DB. Statistical analysis with missing data*. Wiley, London.

Liu, C., Rubin, D. B., and Wu, Y. N. (1998). Parameter expansion to accelerate em: the px-em algorithm. *Biometrika* **85,** 755–770.

Loève, M. (1977). *Probability theory. 1977*. Springer-Verlag, New York.

Lorenzi, M. and Filippone, M. (2018). Constraining the dynamics of deep probabilistic models. *arXiv preprint arXiv:1802.05680* .

Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000). Winbugs-a Bayesian modelling framework: concepts, structure, and extensibility. *Statistics and computing* **10,** 325–337.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge University Press.

Malacara, D. (2011). *Color vision and colorimetry: theory and applications*. Washington: SPIE.

Marqués-Mateu, Á., Lerma, J. L., and Riutort-Mayol, G. (2013). Statistical grey level and noise evaluation of foveon x3 and cfa image sensors. *Optics & Laser Technology* **48,** 1–15.

Matthews, A. G. d. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. (2017). GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research* **18,** 1–6.

Merrill, R. B. (1999). Color separation in an active pixel cell imaging array using a triple-well structure. US Patent 5,965,875.

Minka, T. (2000). Bayesian linear regression. Technical report, Citeseer.

Minka, T. P. (2001). Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc.

Moore, D. and Russell, S. J. (2015). Gaussian process random fields. In *Advances in Neural Information Processing Systems*, pages 3357–3365.

Neal, R. M. (1993). Probabilistic inference using markov chain monte carlo methods. Technical report, Department of Computer Science, University of Toronto Toronto, Ontario, Canada. CRGT-TG-93-1.

Neal, R. M. (1997). Monte carlo implementation of gaussian process models for bayesian regression and classification. *arXiv preprint physics:9701026* .

Neal, R. M. (1999). Regression and classification using gaussian process priors (with discussion). *In Bernardo, J.M., Berger, J.O., Dawid, A.P., and Smith, A.F.M., editors, Bayesian Statistics. Oxford University Press* **6,** 475–501.

Neal, R. M. et al. (2011). MCMC using Hamiltonian dynamics. *arXiv preprint arXiv:1206.1901* .

Neelon, B. and Dunson, D. B. (2004). Bayesian isotonic regression and trend analysis. *Biometrics* **60,** 398–406.

Nievergelt, Y. (1993). Splines in single and multivariable calculus. *UMAP: Module* **718,**.

Ntzoufras, I. (2011). *Bayesian modeling using WinBUGS*, volume 698. John Wiley & Sons.

O'Hagan, A. and Forster, J. J. (2004). *Kendall's Advanced Theory of Statistics, volume 2B: Bayesian Inference, second edition*, volume 2B. Arnold.

O'Hagan, T. (2004). Dicing with the unknown. *Significance* **1,** 132–133.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: Networks of plausible inference*. Morgan Kaufmann Publishers Inc.

Piironen, J., Paasiniemi, M., and Vehtari, A. (2018). Projective inference in high-dimensional problems: prediction and feature selection. *arXiv preprint arXiv:1810.02406* .

Piironen, J. and Vehtari, A. (2016). On the hyperprior choice for the global shrinkage parameter in the horseshoe prior. *arXiv preprint arXiv:1610.05559* .

Piironen, J., Vehtari, A., et al. (2017). Sparsity information and regularization in the horseshoe and other shrinkage priors. *Electronic Journal of Statistics* **11,** 5018–5051.

Pratt, W. K. (2007). *Digital image processing: PIKS Scientific inside*. Wiley-interscience.

Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research* **6,** 1939–1959.

Quiñonero-Candela, J., Rasmussen, C. E., Figueiras-Vidal, A. R., et al. (2010). Sparse spectrum gaussian process regression. *Journal of Machine Learning Research* **11,** 1865–1881.

Rahimi, A. and Recht, B. (2008). Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184.

Rahimi, A. and Recht, B. (2009). Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pages 1313–1320.

Raiko, T. et al. (2006). *Bayesian inference in nonlinear and relational latent variable models*. PhD thesis, Helsinki University of Technology.

Ramsay, J. O. (1998). Estimating smooth monotone functions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **60,** 365–375.

Ramsay, J. O. and Silverman, B. W. (2007). *Applied functional data analysis: methods and case studies*. Springer.

Rasmussen, C. E. (2003). Gaussian processes to speed up hybrid monte carlo for expensive bayesian integrals. *Bayesian Statistics* **7,** 651–659. Oxford University Press.

Rasmussen, C. E. and Nickisch, H. (2010). Gaussian processes for machine learning (gpml) toolbox. *The Journal of Machine Learning Research* **11,** 3011–3015.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian processes for machine learning*. The MIT Press.

Reibel, Y., Jung, M., Bouhifd, M., Cunin, B., and Draman, C. (2003). CCD or CMOS camera noise characterisation. *The European Physical Journal-Applied Physics* **21,** 75–80.

Reich, B. J., Fuentes, M., and Dunson, D. B. (2011). Bayesian spatial quantile regression. *Journal of the American Statistical Association* **106,** 6–20.

Riihimäki, J. and Vehtari, A. (2010). Gaussian processes with monotonicity information. In JMLR, editor, *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9.

Robert, C. and Casella, G. (2013). *Monte Carlo statistical methods*. Springer Science & Business Media.

Roberts, G. O. and Rosenthal, J. S. (2007). Coupling and ergodicity of adaptive markov chain monte carlo algorithms. *Journal of applied probability* **44,** 458–475.

Roberts, S. J. (2010). *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, University of Oxford.

Ruiz, L., Recio, J., Fernández-Sarría, A., and Hermosilla, T. (2011). A feature extraction software tool for agricultural object-based image analysis. *Computers and Electronics in Agriculture* **76,** 284–296.

Ruppert, D., Wand, M. P., and Carroll, R. J. (2003). *Semiparametric regression*. Number 12. Cambridge University Press.

Saatçi, Y. (2012). *Scalable inference for structured Gaussian process models*. PhD thesis, University of Cambrige.

Shawe-Taylor, J., Cristianini, N., et al. (2004). *Kernel methods for pattern analysis*. Cambridge University Press.

Shively, T. S., Sager, T. W., and Walker, S. G. (2009). A Bayesian approach to non-parametric monotone function estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **71,** 159–175.

Sill, J. (1998). Monotonic networks. In *Advances in Neural Information Processing Systems*, pages 661–667.

Snelson, E. and Ghahramani, Z. (2007). Local and global sparse Gaussian process approximations. In *Artificial Intelligence and Statistics*, pages 524–531.

Solak, E., Murray-Smith, R., Leitheah, W. E., Leith, D. J., and Rasmussen, C. E. (2003). Derivative observations in Gaussian process models of dynamic systems. *Advances in Neural Information Processing Systems* pages 1033–1040.

Solin, A. and Särkkä, S. (2014). Explicit link between periodic covariance functions and state space models. In *Artificial Intelligence and Statistics*, pages 904–912.

Solin, A. and Särkkä, S. (2018). Hilbert space methods for reduced-rank Gaussian process regression. *arXiv preprint arXiv:1401.5508* .

Särkkä, S., Solin, A., and Hartikainen, J. (2013). Spatiotemporal learning via infinite-dimensional bayesian filtering and smoothing: A look at Gaussian process regression through kalman filtering. *IEEE Signal Processing Magazine* **30,** 51–61.

Team, S. D. (2017). *Stan Modeling Language Users Guide and Reference Manual, Version 2.17.0.* http://mc-stan.org.

Tse, S., Guild, S., Orlandini, V., and Trojan-Bedynski, M. (2010). Microfade testing of 19th century iron gall inks. *American Institute for Conservation Textile Speciality Group Postprint* **20,** 167–180.

Tsin, Y., Ramesh, V., and Kanade, T. (2001). Statistical calibration of CCD imaging process. In *Proceedings of the IEEE 2001 International Conference on Computer Vision*, volume 1, pages 480–487. IEEE.

Urtasun, R. and Darrell, T. (2008). Sparse probabilistic regression for activity-independent human pose inference. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, pages 1–8. IEEE.

Vanhatalo, J., Pietiläinen, V., and Vehtari, A. (2010). Approximate inference for disease mapping with sparse Gaussian processes. *Statistics in Medicine* **29,** 1580–1607.

Vanhatalo, J., Riihimäki, J., Hartikainen, J., Jylänki, P., Tolvanen, V., and Vehtari, A. (2013). Gpstuff: Bayesian modeling with Gaussian processes. *The Journal of Machine Learning Research* **14,** 1175–1179.

Vehtari, A., Ojanen, J., et al. (2012). A survey of bayesian predictive methods for model assessment, selection and comparison. *Statistics Surveys* pages 142–228.

Wahba, G. (1978). Improper priors, spline smoothing and the problem of guarding against model errors in regression. *Journal of the Royal Statistical Society. Series B (Methodological)* pages 364–372.

Wahba, G. (1990). *Spline models for observational data*, volume 59. SIAM.

Wang, K. A., Pleiss, G., Gardner, J. R., Tyree, S., Weinberger, K. Q., and Wilson, A. G. (2019). Exact gaussian processes on a million data points. *arXiv preprint arXiv:1903.08114* .

Whitmore, P. M., Pan, X., and Bailie, C. (1999). Predicting the fading of objects: Identification of fugitive colorants through direct nondestructive lightfastness measurements. *J Am Inst Conserv* **38,** 395–409.

Williams, C. K. and Barber, D. (1998). Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20,** 1342–1351.

Williams, C. K. and Seeger, M. (2001). Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, pages 682–688.

Wilson, A. and Nickisch, H. (2015). Kernel interpolation for scalable structured Gaussian processes (kiss-gp). In *International Conference on Machine Learning*, pages 1775–1784.

Wood, S. (2015). Package 'mgcv'. *R package version* **1,** 29.

Wood, S. N. (2003). Thin plate regression splines. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **65,** 95–114.

Wood, S. N. (2017). *Generalized additive models: an introduction with R*. Chapman and Hall/CRC.

Woodworth, G. G. (2004). *Biostatistics: a Bayesian introduction*, volume 499. Wiley-Interscience.

Yang, R. and Berger, J. O. (1996). *A catalog of noninformative priors*. Institute of Statistics and Decision Sciences.

Zhang, C., Yao, S., and Xu, J. (2011). Noise in a CMOS digital pixel sensor. *Journal of Semiconductors* **32,** 115005. doi: 10.1088/1674-4926/32/11/115005.