CARDIFF
UNIVERSITY
PRIFYSGOL
CAERDYDD

M.PHIL THESIS

Learning theory & Gaussian Process Regression for surrogate modeling, and a novel framework for Design Optimization under uncertainty. Application to an early-stage aircraft wing design.

José-Luis DORADO-LADERA

A thesis submitted in fulfillment of the requirements for the degree of Master of Philosophy

in the

School of Engineering

May 6, 2021

"If I were again beginning my studies, I would follow the advice of Plato and start with mathematics. "

Galileo Galilei

CARDIFF UNIVERSITY

Abstract

School of Engineering

Master of Philosophy

Learning theory & Gaussian Process Regression for surrogate modeling, and a novel framework for Design Optimization under uncertainty. Application to an early-stage aircraft wing design.

by José-Luis DORADO-LADERA

The aim of this thesis is to study the problem of regression for surrogate modeling, to develop a novel framework for design optimization that makes no assumptions on the model, and to give guidelines to apply the aforementioned theory to an early-stage aircraft wing design. The first part provides a summary of the mathematical foundations of learning theory for regression. The theory of Reproducing Kernel Hilbert Spaces is broadly covered. In addition, Gaussian Process regression is explained in detail. The second part introduces a novel framework for design optimization. Sampling from a probability distribution is at the core of this framework. Therefore, algorithms for simulating different distributions are described in detail. Furthermore, rejection sampling, the theory of Markov chains, and Metropolis-Hasting are explained as methods for simulating arbitrary distributions. The framework aims to optimize the parameters of a Probability Density Function in the input space of a surrogate model in order to satisfy prescribed performance in the output. Stochastic Optimization is suggested as the optimization process and a description of Simulated Annealing is included. This MPhil is part of a project in collaboration with Airbus. They provided a dataset with the goal of optimizing the jig twist of an aircraft wing. The last part analyzes this data and provides future researchers in the project with guidelines to train a Gaussian Process and apply the novel framework mentioned above to tackle the optimization problem.

Acknowledgements

I would like to extend my gratitude to Airbus and EPSRC for making this project available and supplying the funding that has made it possible for me to do this MPhil.

lI would like to express my gratitude to my supervisors Dr. Abhishek Kundu and Prof. David Kennedy for giving me the opportunity to study this MPhil at Cardiff University and for their advice.

I would like to offer my special thanks to Dr. Esther Dorado and Dr. Bertrand Gauthier for their selfless contributions to this thesis. Conversations with Dr. Gauthier improved the quality this work. I owe him the usage of Maximum Mean Discrepancy as a distance between distributions .

I am deeply grateful to the friends I made in Cardiff, especially Dr. Julian Herbert. They made the process of adaptation to a new country and another language much easier for me.

Finally, I would like to thank my family and my closer friends. They are always there to support me, even when there are hundreds of miles between us.

Contents

List of Figures

List of Algorithms

List of Abbreviations

FEM	Finite Element Method.
EI	Bending Stiffness.
GJ	Torsional Stiffness.
SMT	External loads. Shear, Moment and Torque.
DOE	Design Of Experiments.
OLHS	Optimized Latin Hypercube Sampling.
SFD	Space Filling Design.
CFD	Computational Fluid Dynamics.
CSM	Computational Structural Mechanics.
Cl	Coefficient of Lift.
L/D	Lift over Drag ratio.
IQ	Interesting Quantities.
ML	Machine Learning.
GP	Gaussian Process.
PDF	Probability Density Function.
i.i.d.	Independent and identically distributed.
PDS	Positive Definite Symmetric.
RKHS	Reproducing Kernel Hilbert Spaces.
MAP	Maximum a posteriori.
SVD	Singular Value Decomposition.
MCMC	Markov Chain Monte Carlo.
M-H	Metropolis-Hastings.
HMC	Hamiltonian Monte Carlo.
SA	Simulated Annealing.
MMD	Maximum Mean Discrepancy.
NN	Neural Networks.

List of Symbols

$\propto$	Proportional to.
$\mathcal{O}$	big O notation.
$\mathbb{R}^N$	N-dimensional Euclidean space.
$\mathcal{C}(\mathcal{X})$	Banach space of continuous functions on $\mathcal{X}$ with the infinity norm.
Θ	Parameters space.
θ	Parameters variable.
$I_{k\times k}$	$k \times k$ identity matrix.
$\mathcal{X}$	Input space in a regression/surrogate model. Usually a subset of $\mathbb{R}^N$.
$\mathcal{Y}$	Output space in a regression/surrogate model. Usually a subset of $\mathbb{R}^M$.
$\mathcal{Z}$	Joined $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ input-output space.
n	Number of samples.
ν	Borel measure on Euclidean Spaces.
$\mathcal{B}(\mathcal{X})$	Borel σ-algebra on $\mathcal{X}$.
$\mathbb{P}(A)$	Probability that event A occurs. See section 1.4.
$p(x)$	Probability density at the point or element x. See section 1.4.
$\mathbb{E}[X]$	Expected value of X.
$\mathrm{Var}(X)$	Variance of X.
$\mathrm{Cov}(X, X')$	Covariance between X and X'.
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2. This notation will be use in the sense of either a random variable distribution, $\mathcal{X} \sim \mathcal{N}(\mu, \sigma^2)$, or probability density, $p(x) \sim \mathcal{N}(\mu, \sigma^2)$. The terminology Gaussian distribution may be used. The same notation will be used for the multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$.
ρ	Probability measure governing the samples $\mathbf{z} = ((x_1, y_1), \ldots, (x_n, y_n)) \in \mathcal{Z}^n$.
$\rho(y\|x)$	Conditional probability on $\mathcal{Y}$ with respect to $x \in \mathcal{X}$.
$\rho_{\mathcal{X}}(x)$	Marginal probability on $\mathcal{X}$.
$\langle f, g \rangle_{\mathcal{H}}$	Inner product in the Hilbert space $\mathcal{H}$.
$\\|f\\|_{\mathcal{H}}$	Norm in the Banach space $\mathcal{H}$.
f_ρ	Regression function of ρ.
σ^2	Usually the variance of a normal distribution. In GP regression, the variance of the likelihood $p(y\|x, w) = \mathcal{N}(w^T x, \sigma^2)$ or $p(y\|x, w) = \mathcal{N}(w^T \Phi(x), \sigma^2)$.

$\sigma^2(x)$	Variance of $y \in \mathcal{Y}$ according to $\rho(y \mid x)$.
σ_ρ^2	$\mathcal{E}_\rho(f_\rho)$. Expected value of $\sigma^2(x)$ according to $\rho_{\mathcal{X}}$.
Σ_w	Covariance matrix of the weights prior in GP regression.
$\mathcal{E}_\rho(f)$	The least squares error of f w.r.t ρ.
$\boldsymbol{z}$	Independent and identically distributed samples according to ρ.
$\mathcal{E}_{\boldsymbol{z}}(f)$	The empirical error of f with respect to the sample $\boldsymbol{z}$.
$\mathcal{H}$	Hypothesis space. RKHS.
$\mathcal{E}_{\boldsymbol{z},\gamma}(f)$	The regularized empirical error of f w.r.t. the sample $\boldsymbol{z}$.
$f_{\mathcal{H}}$	Target function. A function minimizing the least square error in $\mathcal{H}$.
$f_{\boldsymbol{z}}$	Empirical target function. A function minimizing the empirical error w.r.t. $\boldsymbol{z}$.
$f_{\boldsymbol{z},\gamma}$	Empirical target function. A function minimizing $\mathcal{E}_{\boldsymbol{z},\gamma}$.
$\mathcal{L}^2_\nu(\mathcal{X})$	The Hilbert space of square integrable functions on $\mathcal{X}$.
K	Kernel function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Transition matrix in the MCMC context. Transition kernel function on $\Omega \times \otimes$ in the MCMC context.
$K[\boldsymbol{t}]$	$K[\boldsymbol{t}] = \big(K(t_i, t_j)\big)_{\substack{1 \le i \le s \\ 1 \le j \le s}} \in \mathbb{R}^{s \times s}$ where $t \in \mathcal{X}^s$.
$K[\boldsymbol{t}, \boldsymbol{t'}]$	$K[\boldsymbol{t}, \boldsymbol{t'}] = \Big(K(t_i, t'_j)\Big)_{\substack{1 \le i \le s \\ 1 \le j \le r}} \in \mathbb{R}^{s \times r}$ where $t \in \mathcal{X}^s$ and $t' \in \mathcal{X}^r$.
L_K	Linear operator associated with the kernel K.
λ_k	kth eigenvalue of a Linear operator.
$\Phi(x)$	Feature map.
$\phi(x)$	kth component of a feature map. kth eigenfunction of a Linear operator.
$\boldsymbol{x}$	$\boldsymbol{x} = \{x_1, \ldots, x_n\}$ where $x_i \in \mathcal{X}$ are the inputs in the training set. $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ where $x_i \in \mathcal{X}$, set of samples in the input space of a surrogate model.
$\boldsymbol{y}$	$\boldsymbol{y} = (y_1, \ldots, y_n)^T$ where $y_i \in \mathcal{Y}$ are the outputs in the training set. $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ where $y_i \in \mathcal{Y}$, set of samples in the output space of a surrogate model.
$\boldsymbol{y}_{\boldsymbol{x}}$	Set of samples in the output space of a surrogate model propagated from the set of samples $\boldsymbol{x}$ in the input space.
$\boldsymbol{X}$	Design matrix $\boldsymbol{X} = [x_1 \ldots x_n] \in \mathbb{R}^{N \times n}$ in GP regression.
$\Phi_{\boldsymbol{X}}$	Proyection of the Design matrix into a feature space $\Phi_{\boldsymbol{X}} = [\Phi(x_1) \ldots \Phi(x_n)]$.
w	Vector of weights in GP regression.
$\bar{w}$	Weights MAP estimate.
$w_{\mathcal{E}_{\boldsymbol{z}}}$	Vector of weights minimizing the empirical error $\mathcal{E}_{\boldsymbol{z}}$.
$w_{\mathcal{E}_{\boldsymbol{z},\gamma}}$	Vector of weights minimizing the regularized empirical error $\mathcal{E}_{\boldsymbol{z},\gamma}$.
A	In GP regression, matrix defined in (2.2.2.2).
A_Φ	In GP regression, matrix defined in (2.35).
Ω	Sample space.
$\mathcal{F}$	Parameteric family of density functions in the input space of a surrogate model.

$\mu_{\mathbb{P}}$	Kernel mean embedding of the probability distribution $\mathbb{P}$.
λ	In section 3.2, parameters of a probability distributions.
$S(\lambda)$	Space of parameters λ.
f_T	Target PDF in the output space of a surrogate model.
f_λ	PDF in $\mathcal{F}$ with parameters λ.
$f_{\mu,\Sigma}$	PDF in the space of gaussian densities with mean μ and covariance matrix Σ.
H	Objective function in an optimization problem.

Dedicated to Esther, my sister, Manuela, my mother and Arya, my love.

Chapter 1

Introduction

The motivation of this work is the optimization of the design parameters of a aircraft wing using data collected from simulations. Nevertheless, this thesis has been written considering an arbitrary design problem. It is divided into two theoretical Chapters (Chapters 2 and 3) and a Chapter that provides guidelines to apply that theory to a particular dataset (Chapter 4) related to the aforementioned aircraft wing case.

The goal of Chapter 2 is to study the problem of regression for surrogate modeling. In order to do so, the learning theory approach is adopted.

Chapter 3 introduces a novel framework for design optimization.

It is assumed that the reader has basic mathematical knowledge in algebra, analysis, probability and measure theory, and Bayesian analysis. Notation clarifications are provided in section 1.4.

1.1 Forward problem

Chapter 2 is a summary of the mathematical foundations of learning theory and the theory of Gaussian Process (GP) regression. Sections 2.1.1, 2.1.2, 2.1.3 and 2.1.4 set the basic concepts and notations of learning theory.

GP is the regression algorithm adopted in this thesis. Reproducing Kernel Hilbert Spaces (RKHS) are a family of Hypothesis Spaces broadly used in Machine Learning (ML) and strong connections between GP and RKHS theory are given.

Section 2.1.5 explains the RKHS theory. Sections 2.1.5.1 and 2.1.5.2

define a linear operator L_K given by a kernel function K and its spectral decomposition. Section 2.1.5.3 shows that a kernel K that is symmetric and positive-definite can be decomposed in a summation involving the eigenfunctions of the linear operator L_K (theorem 2.1.2). Section 2.1.5.4 uses this result to state and prove "the kernel trick" (theorem 2.1.3). The kernel trick is a widely used technique in the ML community to "kernelize" algorithms. This is the case of GP regression where the kernel trick allows the linear model to work with infinite-dimensional feature spaces.

Section 2.1.5.5 characterizes RKHS spaces and their inner product. Section 2.1.5.6 proves the celebrated Representer theorem (theorem 2.1.6). Finally, section 2.2 explains GP regression theory.Full justification of the mathematical steps needed to build a GP for regression are given.

1.2 Inverse problem

Chapter 3 introduces a novel framework for design optimization.The aim is to find a probability distribution in the input space of a surrogate model that satisfies a prescribed performance in the output when uncertainties are propagated.

Sampling from a distribution is at the core of this framework. Section 3.1 provides algorithms for sampling from a given Probability Density Function (PDF).

Sections 3.1.1 and 3.1.2 describe how a computer generates uniform random samples from $(0, 1)$, which is the basic tool for any other sampling algorithm.

Section 3.1.3 explains the inverse transform sampling method, which is used in section 3.1.4 to give two algorithms (algorithms 3 and 4) for sampling from a standard normal distribution.

In order to apply the framework introduced in this thesis, it is needed to set a family of probability distributions $\mathcal{F}$ in the input space of the surrogate model. One of the families proposed in this work is the multivariate normal distribution, and efficient methods for sampling from it are required. Section 3.1.4.2 uses algorithms 3 or 4 to design a method for sampling from any multivariate normal distribution.

Section 3.1.5 presents two methods for sampling from an arbitrary distribution with known PDF: Rejection sampling and Metropolis Hasting (M-H), and some notion of Markov chain theory, which is required to understand M-H.

The main novel ideas of this thesis are given in sections 3.1.6 and 3.2.

Section 3.1.6 illustrates a method to use the sampling algorithms explained in the Chapter (or any other sampling algorithm) in constrained spaces. This method would be relevant to satisfy constraints in the input variables of the surrogate model, or if there are constraints in the parameters of the parametric family $\mathcal{F}$. For instance, the weights of mixture distributions described in section 3.2.4.2.

Section 3.2 introduces the aforementioned novel framework for design optimization.A parametric family of distributions $\mathcal{F}$ is set in the input space of the surrogate model and its parameters are optimized to satisfy the prescribed performance in the outputs. Different approaches could be adopted for the optimization process. Section 3.2.2 suggests stochastic optimization. Simulated Annealing (SA), which is based in M-H, is described in section 3.2.2.1.

Different objective functions are proposed according to different requirements in the output space. Section 3.2.3 provides an objective function for the particular case of a given target PDF, which is explained in detail.

Section 3.2.4 suggests multivariate normals and mixture distributions as parametric families $\mathcal{F}$.

An important trait of this framework is that it makes no assumptions in the surrogate model. However, section 3.2.5 makes an interesting suggestion if differentiability is provable.

1.3 Airbus project

The research conducted during this MPhil was part of a project in collaboration with Airbus. They provided a dataset with the goal of optimizing the jig twist of an aircraft wing.

The aim of Chapter 4 is to provide future researchers in the project with a methodology to tackle this problem. This methodology is based on Chapters 2 and 3.

Sections 4.1 and 4.2 analyze the data in detail. Experiments were conducted in the dataset to make recommendations about training a GP. Section 4.3 provides guidelines to apply the design optimization framework introduced in Chapter 3 in this particular problem.

1.4 Probability measures and notation

The Borel σ-algebra on an arbitrary topological space $\mathcal{Z}$ will be denoted by $\mathcal{B}(\mathcal{Z})$. On a sample space $\mathcal{Z}$, if the σ-algebra of the probability space is not specified, then it will be $\mathcal{B}(\mathcal{Z})$.

Let Z be a random variable that takes values on a topological space $\mathcal{Z}$. Let ρ be a probability measure on $\mathcal{B}(\mathcal{Z})$ and consider the probability space $(\mathcal{Z}, \mathcal{B}(\mathcal{Z}), \rho)$. In some cases, the probability density of $z \in \mathcal{Z}$ with respect to the Lebesgue measure on $(\mathcal{Z}, \mathcal{B}(\mathcal{Z}))$ will be denoted by $p(z)$ and defined as the Radon-Nikodym derivative,

$$p(z) = \frac{d\rho(z)}{dz}.$$

Rigorously, a PDF f_Z of the random variable Z should be defined. However, for the sake of simplicity, p will be used to refer to probability density with no distinction between random variables. Therefore, if Z' is another random variable in a measurable space $(\mathcal{Z}', \mathcal{B}(\mathcal{Z}'), \rho')$ with PDF $f_{Z'}$ then,

$$p(z) = f_Z(z),\ \forall z \in \mathcal{Z},$$

and,

$$p(z') = f_{Z'}(z'),\ \forall z' \in \mathcal{Z}'.$$

Analogously, if $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $Z = (X, Y)$, then the conditional probability density of $y \in \mathcal{Y}$ given $x \in \mathcal{X}$ will be,

$$p(y|x) = \frac{d\rho(y|x)}{dy},$$

and the marginal probability density of $x \in \mathcal{X}$ will be,

$$p(x) = \frac{d\rho_{\mathcal{X}}(x)}{dx}.$$

Therefore, if $A \in \mathcal{B}(\mathcal{X})$, $B \in \mathcal{B}(\mathcal{Y})$, $C \in \mathcal{B}(\mathcal{Z})$ and $x \in \mathcal{X}$ then,

$$\begin{array}{rcccl} \mathbb{P}(X \in A) & = & \int_A d\rho_{\mathcal{X}}(x) & = & \int_A p(x)dx, \\ \mathbb{P}(Y \in B \mid x) & = & \int_B d\rho(y|x) & = & \int_B p(y|x)dy, \\ \mathbb{P}(Z \in C) & = & \int_C d\rho(z) & = & \int_C p(z)dz, \end{array}$$

where $\mathbb{P}$ is used for denoting the probability that an event occurs.

Not only the definition of the PDF can be avoided but also the random variable itself, and the measurable space will be known by the context. For example, expressions such as,

$$\text{For } x \in \mathcal{X} \text{ and } y \in \mathcal{Y},\ p(y|x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(y-x)^2}{2\sigma^2}\right) = \mathcal{N}(x, \sigma^2),$$

will be common. Hence, there will be abuses of notation,

$$\begin{aligned} x &\sim \mathcal{N}(0,1) \\ p(x) &= \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right), \end{aligned}$$

where there is no distinction between the random variable and the PDF variable.

Following the usual Bayesian inference procedure, conditional probability notation will be used with no rigorous definitions with a variety of objects such as PDF parameters or data points. For instance, expressions such as,

$$p(y|x, \sigma^2) = \mathcal{N}(x, \sigma^2),$$

will be common.

However, in some situations where a more rigorous notation is essential to avoid ambiguity, PDFs or random variables will be specifically defined, especially in Chapter 3.

Symbol $\sim$ will be used to designate that a random variable follows a particular probability distribution, or to indicate that a random variable follows a distribution given by a particular PDF. For example, notations such as,

$$x \sim \mathcal{N}(0,1)$$

and

$$X \sim f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right),$$

or even

$$x \sim f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right),$$

will be used.

These abuses of notation are adopted in order to simplify the exposition and to aid understanding.

Chapter 2

Forward problem: a review on learning theory and Gaussian Process Regression

The motivation of this chapter arises from the need for surrogate models in the cases where optimization algorithms are run over a design space and the outcome of interest is not easily measured. For example, simulations from a Finite Element Method (FEM) might take several hours or even days. This computational cost makes the run of optimization algorithms unfeasible since they require evaluating the model many times (see, e.g., Wang et al. (2005)). See Chapter 4 for a specific example. In these cases, the outcome of interest is measured only in a few points in the design space generating a dataset. When running an optimization algorithm, the outcome of points that are not in that dataset can be estimated by interpolation. The method to interpolate the points in the dataset is called surrogate model[1]. For instance, Forrester, Sóbester, and Keane (2008) explain this issue in engineering design and provide different surrogate modeling techniques to tackle it.

The problem of estimating the unknown points from the dataset is known as regression in statistical modeling and Machine Learning (ML). This chapter provides a summary of the mathematical foundations of learning theory[2] and describes one of the algorithms used to tackle this problem, Gaussian Process (GP) regression. GP regression is also known as kriging in geostatistics and Engineering (see, e.g., Simpson et al. (2001)).

[1] Also known as metamodels (a model of the model), approximation models, response surface models, or emulators.

[2] The general theory behind the regression problem from the ML perspective.

The benefits and advantages of using GP as a surrogate modeling technique are given in the Chapter. The key points are highlighted in remarks 27, 29, 41 and 38, and in theorem 2.1.6.

Therefore, the first part of this chapter (section 2.1) is dedicated to the study of the mathematical foundations of learning theory. The second part (section 2.2) describes Gaussian Process regression in detail.

2.1 Fundamentals of learning

This section intends to illustrate the mathematical tools and objects in probability and measure theory which are used in learning theory to tackle regression. The goal is to approximate a regression function $f_\rho : \mathcal{X} \to \mathcal{Y}$ where ρ is a probability measure defined on $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and it is unknown. However, samples of ρ,

$$\mathbf{z} = \{(x_0, y_0), \dots, (x_n, y_n)\},$$

are given.

$\mathcal{X}$ and $\mathcal{Y}$ are usually finite dimensional vector spaces (of possibly different dimension). Considering $\mathcal{Y} \subset \mathbb{R}^M$, it will be assumed that $M = 1$ in this Chapter. In contrast, considering $\mathcal{X} \subset \mathbb{R}^N$, N can be any strictly positive integer.

2.1.1 The regression function f_ρ

Let $\rho(y|x)$ be the conditional probability on $\mathcal{Y}$ with respect to $x \in \mathcal{X}$. Let $\rho_\mathcal{X}(x)$ be the marginal probability on $\mathcal{X}$. This is $\rho_\mathcal{X}(A) = \rho(\pi^{-1}(A))$ where,

$$\begin{aligned} \pi \colon \mathcal{X} \times \mathcal{Y} &\to \mathcal{X} \\ (x, y) &\mapsto x, \end{aligned}$$

and $A \subset \mathcal{X}$.

Definition 2.1.1 (Regression function)**.** *The regression function is defined by,*

$$\begin{aligned} f_\rho \colon \mathcal{X} &\to \mathcal{Y} \\ x &\mapsto f_\rho(x) = \int_\mathcal{Y} y \, d\rho(y|x). \end{aligned} \tag{2.1}$$

Therefore, $f_\rho(x)$ is the expected value of $y \in \mathcal{Y}$ given $x \in \mathcal{X}$.

Remark 1. f_ρ can not be calculated since $\rho(y|x)$ is unknown.

Similarly, a variance function on $\mathcal{X}$ can be defined as the variance of $\rho(y|x)$.

Definition 2.1.2 (Variance function). *The variance function σ^2 is defined by,*

$$\begin{aligned} \sigma^2 \colon \mathcal{X} &\to \mathbb{R} \\ x \mapsto \sigma^2(x) &= \int_{\mathcal{Y}} (y - f_\rho(x))^2 \, d\rho(y|x). \end{aligned}$$

Definition 2.1.3. *σ_ρ^2 is used to refer to the expected value of $\sigma^2(x)$ according to $\rho_{\mathcal{X}}$,*

$$\sigma_\rho^2 = \int_{\mathcal{X}} \sigma^2(x) \, d\rho_{\mathcal{X}}.$$

Remark 2. σ_ρ^2 can be seen as an analogous to the condition number in linear algebra (Cucker and Smale (2001)). Considering in this case probability measures instead of matrices.

Remark 3. σ_ρ^2 only depends on the probability measure ρ.

2.1.2 The generalization error

Definition 2.1.4 (Generalization error). *Given an arbitrary ρ-integrable function $f : \mathcal{X} \to \mathcal{Y}$, the generalization error (or least squares error) of f is defined as,*

$$\mathcal{E}_\rho(f) = \int_{\mathcal{Z}} (f(x) - y)^2 \, d\rho. \tag{2.2}$$

It is possible to "break" the integral that defines the generalization error. Although it is defined with respect to ρ, it can be decomposed to express it as an integral with respect to the conditional probability $\rho(y|x)$ and the marginal $\rho_{\mathcal{X}}$.

Proposition 2.1.1 (Fubini's theorem). *Let $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a ρ-integrable function. Then,*

$$\int_{\mathcal{Z} = \mathcal{X} \times \mathcal{Y}} g(x,y) \, d\rho = \int_{\mathcal{X}} \left(\int_{\mathcal{Y}} g(x,y) \, d\rho(y|x) \right) d\rho_{\mathcal{X}}.$$

Proposition 2.1.2. *Let $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a ρ-integrable function. Then,*

$$\mathcal{E}_\rho(f) = \int_{\mathcal{X}} (f(x) - f_\rho(x))^2 \, d\rho_{\mathcal{X}} + \sigma_\rho^2. \tag{2.3}$$

Proof. Using proposition 2.1.1,

$$\begin{array}{rcl}\mathcal{E}_\rho(f) & = & \int_{\mathcal{Z}}(f(x)-y)^2\,d\rho \\ & = & \int_{\mathcal{Z}}\left((f(x)-f_\rho(x))-(y-f_\rho(x))\right)^2\,d\rho \\ & = & \int_{\mathcal{X}}(f(x)-f_\rho(x))^2\,d\rho_{\mathcal{X}}+\int_{\mathcal{Z}}(y-f_\rho(x))^2\,d\rho \\ & & +2\int_{\mathcal{X}}(f(x)-f_\rho(x))\left(\int_{\mathcal{Y}}(y-f_\rho(x))\,d\rho(y|x)\right)d\rho_{\mathcal{X}} \\ & = & \int_{\mathcal{X}}(f(x)-f_\rho(x))^2\,d\rho_{\mathcal{X}}+\sigma_\rho^2.\end{array}$$

□

Corollary 2.1.1. *f_ρ (2.1) minimizes $\mathcal{E}_\rho$ (2.2).*

The first term of $\mathcal{E}_\rho$ considering its expression in (2.3),

$$\int_{\mathcal{X}}(f(x)-f_\rho(x))^2\,d\rho_{\mathcal{X}},$$

is an average of the error of using f instead of f_ρ. The second term σ_ρ^2 is a lower bound of $\mathcal{E}_\rho$ which is reached when $f=f_\rho$. It only depends on the probability measure ρ (remark 3).

Therefore, as mentioned at the beginning of this section, the goal is to approximate f_ρ from samples of ρ on $\mathcal{Z}$.

2.1.2.1 Empirical error

Let

$$\mathbf{z}=\{(x_1,y_1),\dots,(x_n,y_n)\}\in\mathcal{Z}^n$$

be independent and identically distributed samples according to the probability measure ρ.

Definition 2.1.5 (Empirical error). *If $f:\mathcal{X}\to\mathcal{Y}$ is a ρ-integrable function, then the empirical error of f with respect to $\mathbf{z}$ is,*

$$\mathcal{E}_{\mathbf{z}}(f)=\frac{1}{n}\sum_{i=1}^{n}(f(x_i)-y_i)^2.$$

Remember that $\mathcal{E}_{\mathbf{z}}(f)$ converges in probability towards $\mathcal{E}_\rho(f)$ because of the law of large numbers. See, e.g., Dekking (2005) or Yao and Gao (2016) for more details about the law of large numbers. See,e.g., Cucker and Smale (2001) for more details about the convergence of $\mathcal{E}_{\mathbf{z}}(f)$ towards $\mathcal{E}_\rho(f)$.

2.1.3 Hypothesis spaces and the existence of $f_{\mathcal{H}}$ and f_z

Definition 2.1.6 ($\mathcal{C}(\mathcal{X})$ as a Banach space)**.** *Let $\mathcal{C}(\mathcal{X})$ be the space of continuous functions on $\mathcal{X}$ with the infinity norm,*

$$\|f\|_\infty = \sup_{x\in\mathcal{X}} |f(x)|, \;\; f \in \mathcal{C}(\mathcal{X}). \tag{2.4}$$

Definition 2.1.7 (Hypothesis space)**.** *The hypothesis space $\mathcal{H}$ is a subset of $\mathcal{C}(\mathcal{X})$ such that,*

1. *$\mathcal{H}$ is compact,*

2. *for all $f \in \mathcal{H}$, there exists M such that $|f(x) - y| \leq M$ almost everywhere (w.r.t ρ).*

Remark 4. Hypothesis spaces do not necessarily have to be Hilbert spaces. Notice that the infinity norm (2.4) is not derived from an inner product and defining an inner product is not a necessary condition for fulfilling the requirements of definition 2.1.7. However, a Hypothesis space can be a Hilbert space. Indeed, section 2.1.5 focuses on special cases of Hypothesis spaces that are also Hilbert spaces.

Remember that the main goal is to find an approximation of f_ρ. The problem can be reduced to finding an approximation of f_ρ in $\mathcal{H}$.

Therefore, the aim is to formulate algorithms for finding good approximations of f_ρ in $\mathcal{H}$. The selection of $\mathcal{H}$ is closely related to the formulation of those algorithms. Section 2.1.5 will cover one family of hypothesis spaces called Reproducing Kernel Hilbert Spaces (RKHS). Section 2.2 will explain an specific algorithm which uses RKHS spaces to tackle regression.

Expressions 2.5 and 2.6 give a rigorous definition of an approximation in $\mathcal{H}$ with respect to the generalization error (definition 2.1.4) and the empirical error (definition 2.1.5) respectively. Proposition 2.1.3 proves its existence.

Definition 2.1.8 (Target function)**.** *Define $f_{\mathcal{H}}$ to be a function minimizing the generalization error (definition 2.1.4) in $\mathcal{H}$, i.e.,*

$$f_{\mathcal{H}} = \underset{f\in\mathcal{H}}{\operatorname{argmin}} \;\; \mathcal{E}_\rho(f). \tag{2.5}$$

Remark 5. A minimizer of

$$\int_{\mathcal{X}} (f(x) - f_\rho(x))^2 \, d\rho_{\mathcal{X}},$$

is also a minimizer of (2.5). See (2.3).

Definition 2.1.9 (Empirical target function). *Let* $\mathbf{z} \in \mathcal{Z}^n$ *be i.i.d samples according to* ρ*. A empirical target function* $f_{\mathcal{H},\mathbf{z}}$ *(or* $f_{\mathbf{z}}$*) is a function minimizing the empirical error (definition 2.1.5) in* $\mathcal{H}$*, i.e.,*

$$f_{\mathbf{z}} = f_{\mathcal{H},\mathbf{z}} = \underset{f \in \mathcal{H}}{\operatorname{argmin}} \ \mathcal{E}_{\mathbf{z}}(f). \tag{2.6}$$

The existence of $f_{\mathbf{z}}$ and $f_{\mathcal{H}}$ can be proved from the hypotheses on $\mathcal{H}$ (definition 2.1.7).

Proposition 2.1.3. $f_{\mathbf{z}}$ *(2.6) and* $f_{\mathcal{H}}$ *(2.5) exists.*

Proof. The compactness of $\mathcal{H}$ and the continuity of the functions $\mathcal{E}_\rho : \mathcal{C}(\mathcal{X}) \to \mathbb{R}$ and $\mathcal{E}_{\mathbf{z}} : \mathcal{C}(\mathcal{X}) \to \mathbb{R}$ lead to the existence of $f_{\mathcal{H}}$ and $f_{\mathbf{z}}$. A broader discussion can be found in Cucker and Zhou (2007). □

Remark 6. $f_{\mathcal{H}}$ may not be unique. However, it is unique when $\mathcal{H}$ is convex. Proof can be found in, e.g., Cucker and Smale (2001).

2.1.4 Sample error and approximation error

Definition 2.1.10 (Normalized error). *Let* $\mathcal{H}$ *be a hypothesis space (definition 2.1.7). The normalized error in* $\mathcal{H}$ *of a function* $f \in \mathcal{H}$ *is,*

$$\mathcal{E}_{\mathcal{H}}(f) = \mathcal{E}_\rho(f) - \mathcal{E}_\rho(f_{\mathcal{H}}). \tag{2.7}$$

Remark 7. $\mathcal{E}_{\mathcal{H}}(f) \geq 0$ for all $f \in \mathcal{H}$.

Remark 8. $\mathcal{E}_{\mathcal{H}}(f_{\mathcal{H}}) = 0$.

Remember that $f_{\mathcal{H}}$ is a function minimizing the generalization error (definition 2.1.4) in $\mathcal{H}$. Thus, the normalized error (2.7) is the error of using f instead of $f_{\mathcal{H}}$ in $\mathcal{H}$.

Definition 2.1.11 (Sample error). *Let* $\mathcal{H}$ *be a hypothesis space. Let* $f_{\mathbf{z}} \in \mathcal{H}$ *be a function minimizing the empirical error (definition 2.1.5). The sample error is the normalized error (2.7) of* $f_{\mathbf{z}}$*, i.e,*

$$\mathcal{E}_{\mathcal{H}}(f_{\mathbf{z}}) = \mathcal{E}_\rho(f_{\mathbf{z}}) - \mathcal{E}_\rho(f_{\mathcal{H}}). \tag{2.8}$$

Definition 2.1.12 (Approximation error). *Let $\mathcal{H}$ be a hypothesis space. Let*

$$\mathcal{A}(\mathcal{H}) = \int_{\mathcal{X}} (f_{\mathcal{H}}(x) - f_{\rho}(x))^2 \, d\rho_{\mathcal{X}}.$$

The approximation error is the generalization error of $f_{\mathcal{H}}$ (definition 2.1.8), i.e,

$$\mathcal{E}_{\rho}(f_{\mathcal{H}}) = \int_{\mathcal{X}} (f_{\mathcal{H}}(x) - f_{\rho}(x))^2 \, d\rho_{\mathcal{X}} + \sigma_{\rho}^2 = \mathcal{A}(\mathcal{H}) + \sigma_{\rho}^2. \tag{2.9}$$

Remark 9. The approximation error (2.9) is independent of sampling. It can be seen as the error of using $\mathcal{H}$.

Remark 10. From (2.8) and (2.9), the generalization error of a empirical target function $f_{\mathbf{z}}$ can be expressed as follows,

$$\begin{aligned}\underbrace{\int_{\mathcal{X}} (f_{\mathbf{z}}(x) - f_{\rho}(x))^2 \, d\rho_{\mathcal{X}} + \sigma_{\rho}^2}_{\text{Proposition 2.1.2}} &= \underbrace{\mathcal{E}_{\rho}(f_{\mathbf{z}})}_{\substack{\text{Generaliza-}\\\text{tion error}}} = \underbrace{\mathcal{E}_{\mathcal{H}}(f_{\mathbf{z}})}_{\substack{\text{Sample}\\\text{error}}} + \underbrace{\mathcal{E}_{\rho}(f_{\mathcal{H}})}_{\substack{\text{Approxima-}\\\text{tion error}}} \\ &= \mathcal{E}_{\mathcal{H}}(f_{\mathbf{z}}) + \mathcal{A}(\mathcal{H}) + \sigma_{\rho}^2 \,. \end{aligned} \tag{2.10}$$

Remember from remark 3 that σ_{ρ}^2 solely depends on the probability measure on ρ. Hence, the quantity $\int_{\mathcal{X}} (f_{\mathbf{z}}(x) - f_{\rho}(x))^2 \, d\rho_{\mathcal{X}}$ is the excess generalization error of using $f_{\mathbf{z}}$. Estimating this quantity is a primary aim of learning theory.

Observing the right hand side of (2.10), it can be seen that estimating the excess generalization error can be divided into two problems:

1. estimating the sample error $\mathcal{E}_{\mathcal{H}}(f_{\mathbf{z}})$,

2. estimating the excess approximation error $\mathcal{A}(\mathcal{H})$.

Remark 11. Regarding the first point above, the sample error $\mathcal{E}_{\mathcal{H}}(f_{\mathbf{z}})$ depends on the sample $\mathbf{z}$ and, therefore, on the probability measure ρ. Notice that it loses dependence on the behaviour of f_{ρ}. It can be seen as a distance between $f_{\mathbf{z}}$ and $f_{\mathcal{H}}$ in $\mathcal{H}$. Therefore, the complexity or behaviour of f_{ρ} will not affect $\mathcal{E}_{\mathcal{H}}(f_{\mathbf{z}})$.

Regarding the second point, in contrast to the sample error, $\mathcal{A}(\mathcal{H})$ will be affected by the behaviour of f_{ρ}. However, it is independent of sampling.

Selecting a hypothesis space $\mathcal{H}$ and designing algorithms to compute or approximate $f_{\mathbf{z}}$ in $\mathcal{H}$ will be the procedure for tackling the main goal of this Chapter: approximating the regression function f_ρ. A fundamental issue is to select $\mathcal{H}$. The selection of $\mathcal{H}$ plays a key role in the design of those algorithms. It has a major impact on the sample error and the approximation error. The selection of $\mathcal{H}$ to optimize these errors is known as the bias-variance problem.

2.1.4.1 Bias-variance problem

Determine and fix a hypothesis space $\mathcal{H}$. It is clear that increasing the number of samples n makes the sample error lower since there is more information about ρ in order to approximate f_ρ. In addition, that makes no difference to the approximation error which is independent of sampling (remark 9). Therefore, increasing n will decrease the generalization error of $f_{\mathbf{z}}$. See (2.10) that relates generalization, sample and approximation errors. Undoubtedly, the more samples are used, the better $f_{\mathbf{z}}$ will approximate f_ρ.

However, the size of $\mathcal{H}$ does not make the same impact on $\mathcal{E}_\rho(f_{\mathbf{z}})$. Now fix n. If $\mathcal{H}$ grows with the addition of more elements, then the approximation error will decrease (or remain stable at worst) because it will be possible to find a better approximation $f_{\mathcal{H}}$ of f_ρ in $\mathcal{H}$. Nevertheless, the sample error will increase. The bias-variance problem aim is to choose the size of $\mathcal{H}$ when n is fixed in order to minimize the generalization error $\mathcal{E}_\rho(f_{\mathbf{z}})$.

In the literature, the terminology bias-variance refers to the optimization of the bias (associated with the approximation error) and the variance (associated with the sample error).

In summary, a very small hypothesis space $\mathcal{H}$ leads to a large approximation error $\mathcal{E}_\rho(f_{\mathcal{H}})$. This is called underfitting. Yet, a large hypothesis space $\mathcal{H}$ leads to a large sample error $\mathcal{E}_{\mathcal{H}}(f_{\mathbf{z}})$. This is called overfitting.

Example 1. Figure 2.1 illustrates examples of underfitting and overfitting. A function built from a second degree polynomial and two exponential functions was used as a regression function f_ρ. Training points $\mathbf{z}$ were generated from evaluations of f_ρ plus Gaussian noise to represent a probability measure ρ. With this construction, $\rho(y|x)$ is a Gaussian distribution (in a real scenario, however, the only information about ρ would be $\mathbf{z}$).

The underfitting example was generated minimizing the empirical error $\mathcal{E}_{\mathbf{z}}(f)$ (definition 2.1.5) over the following hypothesis space

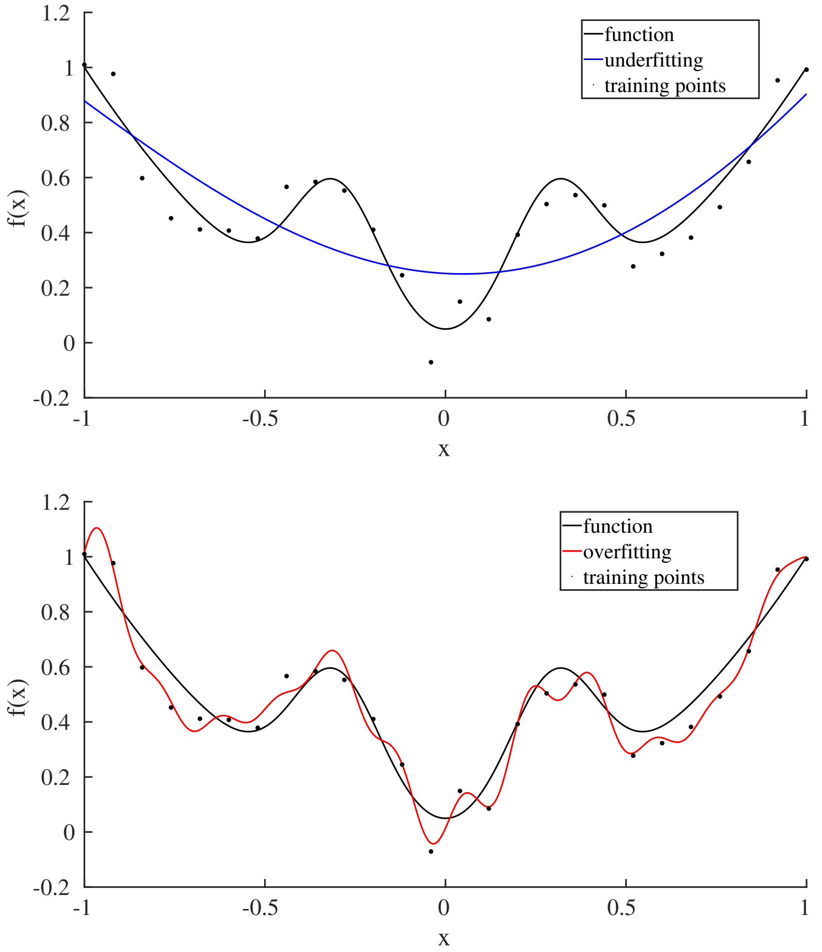


FIGURE 2.1: Example of underfitting and overfitting. *Black line*: a $\mathcal{C}^\infty$ function f_ρ. *Black points*: training points. Evaluations of f_ρ with Gaussian noise. *Blue line*: example of underfitting. Function $f_{\mathcal{H}_u}$ minimizing the empirical error $\mathcal{E}_{\boldsymbol{z}}(f)$ in $\mathcal{H}_u$ (2.11) where $\boldsymbol{z}$ are the training points. *Red line*: example of overfitting. It is analogous to the blue line but using $\mathcal{H}_o \supset \mathcal{H}_u$ (2.12) instead of $\mathcal{H}_u$.

of functions,

$$\mathcal{H}_u = \{f \in \mathcal{C}(\mathbb{R}) \mid f(x) = \sum_{i=1}^{2} a_i \sin(b_i x + c_i),\ a_i, b_i, c_i \in \mathbb{R}\}. \quad (2.11)$$

The problem in this case is that there are too many changes in the behaviour of f_ρ to be captured by the functions in $\mathcal{H}_u$.

The overfitting example is analogous to the underfitting one. However, the hypothesis space used was

$$\mathcal{H}_o = \{f \in \mathcal{C}(\mathbb{R}) \mid f(x) = \sum_{i=1}^{8} a_i \sin(b_i x + c_i),\ a_i, b_i, c_i \in \mathbb{R}\} \supset \mathcal{H}_u. \quad (2.12)$$

In contrast with the underfitting example, the problem here is the possibility of finding functions in $\mathcal{H}_o$ which are able to fit very well all training points $\mathbf{z}$. Although it leads to a low empirical error, it fails in approximating smoothly f_ρ and overfits the particular training points that are being used. Notice that if the Gaussian noise is removed, i.e., without considering the stochastic approach of having a probability measure ρ governing the samples, then there would not be overfitting problem.

The bias-variance is a central problem in Machine Learning theory for regression and classification. Therefore, it is widely covered in the literature of this subject. Notice that it was illustrated from the regression perspective in this work. See, e.g., von Luxburg and Schoelkopf (2008), Geman, Bienenstock, and Doursat (1992) or Cucker and Smale (2001) for more details about it.

2.1.4.2 Regularized error

In the last section, the bias-variance problem was presented. The overfitting and underfitting problems were associated with the selection of the hypothesis space $\mathcal{H}$. Consider that $\mathcal{H}$ is fixed in this section. The underfitting problem can only be tackle adding new elements to $\mathcal{H}$ but $\mathcal{H}$ is fixed. However, the overfitting problem can be tackle adding a regularized term to the generalization error, which depends on the functions $f \in \mathcal{H}$ but not on the sample z.

Consider $\mathcal{H}$ to be a Hilbert space of functions. The approach is to define that regularized term according to the norm induced by the inner product in $\mathcal{H}$.

Definition 2.1.13 (Regularized error). *Let $\mathcal{H}$ be a hypothesis space (definition 2.1.7) and a Hilbert space of functions. Given an arbitrary function $f \in \mathcal{H}$, the regularized error of f is defined as,*

$$\mathcal{E}_\gamma(f) = \int_{\mathcal{Z}} (f(x) - y)^2 \, d\rho + \gamma \|f\|^2_{\mathcal{H}}.$$

Definition 2.1.14 (Regularized empirical error). *Let $\mathcal{H}$ be a hypothesis space (definition 2.1.7) and a Hilbert space of functions. The empirical error of $f \in \mathcal{H}$ with respect to the sample $\mathbf{z} = \{(x_i, y_i)\}_{1 \leq i \leq n}$ is,*

$$\mathcal{E}_{\mathbf{z},\gamma}(f) = \frac{1}{n} \sum_{i=1}^{n} (f(x_i) - y_i)^2 + \gamma \|f\|^2_{\mathcal{H}}.$$

Definition 2.1.15 (Empirical target function w.r.t. $\mathcal{E}_{\mathbf{z},\gamma}$). *Let $\mathbf{z} \in \mathcal{Z}^n$ be i.i.d samples according to ρ. A empirical target function $f_{\mathcal{H},\mathbf{z},\gamma}$ (or $f_{\mathbf{z},\gamma}$) is a function minimizing the regularized empirical error (definition 2.1.14) in $\mathcal{H}$, i.e.,*

$$f_{\mathbf{z},\gamma} = f_{\mathcal{H},\mathbf{z},\gamma} = \underset{f \in \mathcal{H}}{\text{argmin}} \ \mathcal{E}_{\mathbf{z},\gamma}(f). \tag{2.13}$$

Given a sample $\mathbf{z} = \{(x_i, y_i)\}_{1 \leq i \leq n}$, instead of considering the empirical error $f_{\mathbf{z}}$, the idea is to tackle regression considering the problem of finding a minimizer $f_{\mathbf{z},\gamma}$ of the regularized empirical error in $\mathcal{H}$. The Representer theorem (theorem 2.1.6) will cover this problem in a special family of hypothesis spaces called Reproducing Kernel Hilbert Spaces (RKHS).

2.1.5 Reproducing Kernel Hilbert Spaces (RKHS)

In the previous sections of this Chapter, the regression problem (i.e., function fitting) was presented from the point of view of a measure ρ governing the pairs (x, y) where the components x are the inputs and the components y are the outputs.

The aim is to find a function $f_{\mathbf{z}}$ that approximates the regression function f_ρ (2.1). The key issue is that the measure ρ is unknown. However, samples $\mathbf{z} = \{(x_0, y_0), \ldots, (x_n, y_n)\}$ of ρ are given. Thus, algorithms will try to build $f_{\mathbf{z}}$ or $f_{\mathbf{z},\gamma}$ from the samples $\mathbf{z}$. The set $\mathbf{z}$ is usally called the training set.

Instead of considering any continuous function to be a $f_{\mathbf{z},\gamma}$ candidate, the problem was reduced to considering only functions in a compact space of functions $\mathcal{H} \subset \mathcal{C}(\mathcal{X})$ called hypothesis space. Section 2.1.4 explains the importance of the choice of $\mathcal{H}$. This section will introduce a family of infinite dimensional hypothesis spaces broadly used in Machine Learning (ML) algorithms called Reproducing Kernel Hilbert Spaces. The Gaussian Process regression is one of those algorithms. Indeed, it is the algorithm adopted in this work to tackle regression (section 2.2).

Although this work is not focusing on the proofs of the different theorems and propositions presented and its aim is an understanding of the mathematical objects and fundations of learning theory, the proofs of the four main results (theorems 2.1.3, 2.1.4, 2.1.5 and 2.1.6) are provided. In the author's opinion, they are interesting for fully understanding the power of using kernels (definition 2.1.16) and their corresponding RKHS spaces.

2.1.5.1 Operators defined by a kernel

Remember that $\mathcal{X}$ is the space of inputs in the regression problem addressed in this Chapter. In this section, it will be assumed that $\mathcal{X}$ is a compact domain or manifold in an Euclidean space. ν will be used to refer to a Borel measure on Euclidean spaces (e.g., the Lebesgue measure).

Remember that $\mathcal{C}(\mathcal{X})$ is the space of continuous functions on $\mathcal{X}$ with the infinity norm (definition 2.1.6).

Definition 2.1.16 (Kernel). *A continuous function,*

$$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R},$$

is called kernel.

Definition 2.1.17. *Let $\mathcal{L}^2_\nu(\mathcal{X})$ be the Hilbert space of square integrable functions on $\mathcal{X}$ with the inner product,*

$$\langle f, g \rangle = \int_{\mathcal{X}} f(x)g(x)d\nu.$$

Definition 2.1.18 (Linear operator L_K). *Let K be a kernel. The linear map,*

$$\begin{gathered} L_K : \mathcal{L}^2_\nu(\mathcal{X}) \to \mathcal{C}(\mathcal{X}) \subset \mathcal{L}^2_\nu(\mathcal{X}) \\ f \mapsto L_K f, \end{gathered}$$

where

$$(L_K f)(x) = \int_{\mathcal{X}} K(x,t) f(t) d\nu(t),$$

is the linear operator associated with the kernel K.

Remark 12. L_K is well-defined. In addition, it is a compact operator. Proof of the continuity of $L_K f$ and its compactness can be found in, e.g., Cucker and Smale (2001). The linearity is trivial.

2.1.5.2 Spectral theorem

Above in this section, the linear operators L_K associated with their respective kernels K were presented. The following theorem goal is to expose that there exists a orthonormal basis $\{\phi_1, \phi_2, ...\}$ of $\mathcal{L}^2_\nu(\mathcal{X})$ consisting in eigenfunctions of L_K. The eigenfunctions of L_K are analogous to the eigenvectors of a matrix as a operator in an Euclidean vector space. Remember that, in the case of $\mathcal{L}^2_\nu(\mathcal{X})$, its elements are square-integrable functions on $\mathcal{X}$ and it is an infinite dimensional vector space.

Theorem 2.1.1 (Spectral theorem). *Let $L : H \to H$ be a compact linear operator on an infinite-dimsional Hilbert space H. Then there exists a orthonormal basis of H, $\{\phi_1, \phi_2, ...\}$, consisting of eigenvectors of L (or eigenfunctions in the case of H being a space of functions). In addition, if the set $\{\lambda_k\}_{k\geq 1}$ are the eigenvalues of L (assuming, without lost of generality, $\lambda_k \geq \lambda_{k+1}$, $\forall k$) and it is not a finite set, then $\lambda_k \to 0$ when $k \to \infty$.*

A *self-adjoint* operator $L : H \to H$ is an operator such that,

$$\langle Lf, g \rangle = \langle f, Lg \rangle, \ \forall f, g \in H.$$

In addition a self-adjoint operator L is *positive* if $\langle Lf, f \rangle \geq 0$ for all non-trivial $f \in H$ (or strictly positive if $\langle Lf, f \rangle > 0$).

Proposition 2.1.4. *The eigenvalues $\{\lambda_k\}_{k\geq 1}$ of a compact linear operator L are real if L is self-adjoint. In addition, if L is positive then $\lambda_k \geq 0$, $\forall k \geq 1$. If it is strictly positve then $\lambda_k > 0$, $\forall k \geq 1$.*

Theorem 2.1.1 and proposition 2.1.4 are fundamental results of spectral theory. Proof of them can be found in Debnath and Mikusinski (1990).

2.1.5.3 Mercer kernels and Mercer's theorem

Definition 2.1.19 (Symmetric kernel). *A kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is symmetric if $K(x, x') = K(x', x)$ for all $x, x' \in \mathcal{X}$.*

Definition 2.1.20 (Positive-definite kernel). *A symmetric kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is positive-definite if*

$$\sum_{i}^{s} \sum_{j}^{s} c_i c_j K(x_i, x_j) > 0 \tag{2.14}$$

for any $s \in \mathbb{N}$, $x_1, \ldots, x_s \in \mathcal{X}$ and $(c_1, \ldots, c_s) \in \mathbb{R}^s \setminus \{0\}$.

If $\boldsymbol{x} = (x_1, \ldots, x_s)$, $x_i \in \mathcal{X}$, let $K[\boldsymbol{x}] \in \mathbb{R}^{s \times s}$ be the matrix whose entries are $K(x_i, x_j)$,

$$K[\boldsymbol{x}] = \begin{pmatrix} K(x_1, x_1) & \ldots & K(x_1, x_s) \\ \vdots & \ddots & \vdots \\ K(x_s, x_1) & \ldots & K(x_s, x_s) \end{pmatrix}, \tag{2.15}$$

and

$$f_{\boldsymbol{x}} = (f(x_1), \ldots, f(x_s)),$$

the vector of evaluations of f at $x_1, \ldots, x_s$.

Remark 13. Notice that the expression (2.14) is equivalent to $K[\boldsymbol{x}]$ being a positive-definite matrix for any $\boldsymbol{x} \in \mathcal{X}^s$ and $s \in \mathbb{N}$, i.e.,

$$v^T K[\boldsymbol{x}] v > 0$$

for any $v \in \mathbb{R}^s \setminus \{0\}$, $\boldsymbol{x} \in \mathcal{X}^s$ and $s \in \mathbb{N}$.

Proposition 2.1.5. *If a kernel K is symmetric, then the associated linear operator L_K is self-adjoint.*

Proof. The result is a direct consequence of the Fubini-Tonelli's theorem (which allows to change the order of integration) and the symmetry of K. □

Proposition 2.1.6. *If a kernel K is positive definite then the associated linear operator L_K is positive.*

Proof. Notice that

$$\begin{aligned}\langle L_K, f\rangle &= \int_{\mathcal{X}}\int_{\mathcal{X}} K(x,t)f(x)f(t)d\nu(x)d\nu(t)\\ &= \lim_{s\to\infty}\frac{\nu(\mathcal{X})}{s^2}\sum_{i=1}^{s}\sum_{j=1}^{s}K(x_i,x_j)f(x_i)f(x_j)\\ &= \lim_{s\to\infty}\frac{\nu(\mathcal{X})}{s^2}f_{\boldsymbol{x}}^T K[\boldsymbol{x}]f_{\boldsymbol{x}},\end{aligned}$$

and $f_{\boldsymbol{x}}^T K[\boldsymbol{x}]f_{\boldsymbol{x}} > 0$ since K is positive definite. □

Remark 14. If a function $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$ is continuous and satisfy the conditions of symmetry and positive-definiteness of a kernel (definitions 2.1.19 and 2.1.20), then for the remark 12 and the propositions 2.1.5 and 2.1.6, the associated linear operator L_K is compact, self-adjoint and positive. Notice that those are the hypotheses of the spectral theorem 2.1.1 and the proposition 2.1.4. Therefore, there exists a orthonormal basis of $\mathcal{L}^2_\nu(\mathcal{X})$ consisting of eigenfunctions of L_K, $\{\phi_k\}_{k\geq 1}$. In addition, the respective eigenvalues $\{\lambda_k\}_{k\geq 1}$ are real, positive and $\lambda_k \to 0$ when $k\to\infty$ (assuming, without lost of generality, $\lambda_k \geq \lambda_{k+1}, \forall k$).

Remark 15. If $\{\phi_k\}_{k\geq 1}$ is a orthonormal basis of $\mathcal{L}^2_\nu(\mathcal{X})$, then any function $f\in\mathcal{L}^2_\nu(\mathcal{X})$ can be uniquely written as,

$$f = \sum_{k=1}^{\infty} a_k\phi_k,\ a_k \in \mathbb{R},$$

where,

$$a_k = \langle \phi_k, f\rangle,\ \forall k\geq 1,$$

and the partial sums $\sum_{k=1}^{N} a_k\phi_k$ converge to f in $\mathcal{L}^2_\nu(\mathcal{X})$, i.e.,

$$\sum_{k=1}^{N} a_k\phi_k \xrightarrow{N\to\infty} f,\ \text{with}\ \sum_{k=1}^{N} a_k\phi_k \in \mathcal{L}^2_\nu(\mathcal{X}),\ \forall N\in\mathbb{N}. \tag{2.16}$$

Definition 2.1.21 (Mercer kernel). *A kernel K which is symmetric and positive definite is called Mercer kernel.*

Remark 16. Mercer kernels are usually called Positive Definite Symmetric (PDS) in the literature (see, e.g., Mohri, Rostamizadeh, and Talwalkar (2018)). The terminology of “Mercer kernel” often refers to the kernels that satisfy the conclusion of the Mercer’s theorem

(theorem 2.1.2). Indeed, it can be proved that a kernel is a Mercer kernels (according to this last definition), if and only if, it is a PDS kernel. Hence, they are the same mathematical objects. Theorem 2.1.2 proves the implication to the left. More details can be found in Mohri, Rostamizadeh, and Talwalkar (2018).

Mercer kernels are important mathematical objects because of the implications highlighted in the remark 14 and explained during this section. They are necessary to state the Mercer's theorem (theorem 2.1.2) which plays a key role in the main goal of this section, presenting the Reproducing Kernel Hilbert Spaces.

Remark 17. A pertinent question about Mercer kernels which the reader may be thinking is the existence of such a kernels. They do not only exist, but the next preposition 2.1.7 gives a family of them.

Proposition 2.1.7. *Let $f : (0, \infty) \to \mathbb{R}$ be a completely monotonic function. The kernel,*

$$\begin{aligned} K : \mathcal{X} \times \mathcal{X} &\to \mathbb{R} \\ (x, t) &\mapsto K(x, t) = f(\|x - t\|^2), \end{aligned} \tag{2.17}$$

is a Mercer kernel.

Generally, it is easy to prove the continuity and symmetry of a kernel. However, the positive-definiteness is more challenging. In the case of the kernels defined by (2.17), the continuity and symmetry follows from the completely monotonicity of f and the continuity and symmetry of $\|x - t\|^2$. First proof of the positive-definiteness can be found in Schoenberg (1988).

Examples of kernels according to the proposition 2.1.7 are,

- *Gaussian or squared exponential*:

$$K_{SE}(x, t) = \exp -\frac{\|x - t\|^2}{2l^2}, \ \ l \neq 0. \tag{2.18}$$

- *Rational Quadratic*:

$$K_{RQ}(x, t) = \left(1 - \frac{\|x - t\|^2}{2\alpha l^2}\right)^{-\alpha}, \ \ \alpha, l > 0.$$

Remember that if the convergence in the expression (2.16) satisfy,

$$\sum_{k=1}^{N} a_k \phi_k \in \mathcal{C}(\mathcal{X}),\ \forall N \in \mathbb{N},$$

it is said that the series uniformly converges to f.

In addition, it is said that a serie $\sum_{k=1}^{N} a_k$ converges absolutely if $\sum_{k=1}^{N} |a_k|$ is convergent.

Theorem 2.1.2 (Mercer's theorem)**.** *Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a Mercer kernel (definition 2.1.21). Let $\{\phi_k\}_{k\geq 1}$ be the eigenfunctions of L_K (definition 2.1.18) and $\{\lambda_k\}_{k\geq 1}$ the corresponding eigenvalues. For all $x, t \in \mathcal{X}$,*

$$K(x,t) = \sum_{k=1}^{\infty} \lambda_k \phi_k(x) \phi_k(t), \tag{2.19}$$

where the convergence is uniform and absolute.

Proof. A proof covering all details can be found in Cucker and Zhou (2007). The key point is as follows. Let $K_x(t) = K(x,t)$, $K_x \in \mathcal{L}^2_\nu(\mathcal{X})$. Note that,

$$\begin{aligned}
K(x,t) &= K_x(t) \\
&= \sum_{k=1}^{\infty} \langle K_x, \phi_k \rangle \phi_k(t) && \text{(}\{\phi_k\}_{k\geq 1}\text{ are a orthonormal basis of }\mathcal{L}^2_\nu\text{ (theorem 2.1.1))} \\
&= \sum_{k=1}^{\infty} L_K(\phi_k)(x) \phi_k(t) && \text{(definition 2.1.18)} \\
&= \sum_{k=1}^{\infty} \lambda_k \phi_k(x) \phi_k(t) && \text{(}\phi_k\text{ eigenfunction }L_K\text{ with eigenvalue }\lambda_k\text{)}
\end{aligned}$$

□

First proof of theorem 2.1.2 can be found in Mercer (1909). The first version of it, however, consider ν to be the Lebesgue measure and $\mathcal{X}$ to be an interval $[a, b] \subset \mathbb{R}$. Hochstadt (1989), and Cucker and Zhou (2007) are more recent references for the proof of this theorem. König (1986) gives more details about the generalization for finite measures which is considered in this work.

2.1.5.4 The feature space and the kernel trick

The Mercer′s theorem (theorem 2.1.2) is a necessary requirement for proving the next theorem (theorem 2.1.3), which is a significant result in learning theory. It is an essential tool for designing regression algorithms in Machine Learning, e.g., Gaussian Processes (see section 2.2). It is usually called the kernel trick in the literature of this subject.

Remember that l^2 is the space of square-summable sequences, which is a Hilbert space.

Definition 2.1.22 (Feature map)**.** *Let $\{\phi_k\}_{k\geq 1}$ be the eigenfunctions and $\{\lambda_k\}_{k\geq 1}$ be the corresponding eigenvalues of the linear operator L_K with kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. The map,*

$$\begin{aligned} \Phi : \mathcal{X} &\to l^2 \\ x \mapsto \Phi(x) &= (\sqrt{\lambda_k}\phi_k(x))_{k\geq 1} \end{aligned} \tag{2.20}$$

is called the feature map Φ associated with the kernel K.

Definition 2.1.23 (Feature space)**.** *The image of the feature map $\Phi(\mathcal{X})$ (2.20) is called feature space.*

Remark 18. Notice that the definition 2.1.22 assumes that the set of eigenfunctions $\{\phi_k\}_{k\geq 1}$ is infinite. This is usually the most interesting case. Nevertheless, the definition can easily be changed for the case of $\{\phi_k\}_k$ being finite and then $\Phi(x)$ would also be finite,

$$\Phi(x) = \left(\sqrt{\lambda_1}\phi_1(x), \ldots, \sqrt{\lambda_{k_{max}}}\phi_{k_{max}}(x)\right).$$

Definition 2.1.24. *A kernel which leads to a finite number of non-zero eigenvalues and eigenfunctions is called degenerate kernel.*

Remark 19. A kernel that satisfies the definition 2.1.20 if the equality is considered in (2.14) is called positive-semidefinite kernel. Degenerate kernels are positive-semidefinite kernels that are not positive-definite.

Theorem 2.1.3 (The kernel trick)**.** *The feature map Φ (definition 2.1.22) is well-defined and continuous. In addition,*

$$K(x,t) = \langle \Phi(x), \Phi(t) \rangle.$$

Proof. The following proof can be found in Cucker and Smale (2001). This theorem is an immediate consequence of the Mercer's theorem (theorem 2.1.2),

$$K(x,t) = \sum_{k=1}^{\infty} \lambda_k \phi_k(x)\phi_k(t) = \langle \Phi(x), \Phi(t) \rangle.$$

Also,

$$\sum_{k=1}^{\infty} \lambda_k \phi_k(x)^2 = K(x,x) < \infty, \;\; \forall x \in \mathcal{X}.$$

Thus,

$$(\sqrt{\lambda_k}\phi_k(x))_{k \geq 1} \in l^2.$$

Finally, the continuity of $\Phi : \mathcal{X} \to l^2$ follows from the continuity of K,

$$\begin{aligned} \|\Phi(x) - \Phi(t)\| &= \langle \Phi(x), \Phi(x) \rangle + \langle \Phi(t), \Phi(t) \rangle - 2\langle \Phi(x), \Phi(t) \rangle \\ &= K(x,x) + K(t,t) - 2K(x,t). \end{aligned}$$

Note that $\|\Phi(x) - \Phi(t)\|$ tends to zero when x tends to t. □

Feature spaces are a major breakthrough in function regression because they allow the use of linear regression models even when the data show a non-linear behaviour. Obviously, linear regression applied to the inputs do not capture non-linear behaviour of the target regression function. Therefore, it seems to be useful only for a very few cases, in the linear scenario. However, it is analytically tractable. The idea behind feature spaces is to project the inputs into a space (feature space), whose dimension is higher than the inputs dimension (it is usually infinite-dimsional) and where algorithms will perform linear regression. Although the model of those algorithms will be linear, they will capture non-linear behaviour. The reason for this is that the input projections may show a linear behaviour in the feature space even when the data show a very non-linear scenario.

Example 2. Figure 2.2 illustrates an example of this idea. A two degree polynomial was used as a regression function to be approximated,

$$f(x) = 3x^2 + 2x + 1.$$

Ten random points of this function were generated as a training data. The graph above shows the regression function (blue), the training points (black) and the best linear approximation according to these training points (red). The adjective 'best' was used in the least-squares sense (definition 2.1.5). It can be seen that the linear regression on the inputs fails.

The graph below shows the input projections into the feature space,

$$\Phi(x) = (\phi_1(x), \phi_2(x)) = (x^2, x).$$

Since the regression function is

$$f(x) = 3(x^2) + 2(x) + 1 = 3\phi_1(x) + 2\phi_2(x) + 1,$$

any input point projection belongs to the plane $F(\phi_1, \phi_2) = 3\phi_1 + 2\phi_2 + 1$, and therefore linear regression on the feature space would succeed with a perfect fitting. Indeed,

$$f(x) = 3x^2 + 2x + 1 \in \operatorname{span}\{x^2, x\} \subset \mathcal{C}(\mathcal{X}).$$

Example 3. Projecting into a feature space is interesting even if the regression function does not belong to the selected feature space. Although there is not a perfect fitting, functions belonging to that hypothesis space can be good approximations of the regression function. Figure 2.3 shows an example trying to fit the same function $f(x) = 3x^2 + 2x + 1$ in figure 2.2. However, in this case, the feature space is

$$\Phi(x) = (\phi_1(x), \phi_2(x)) = (\cos(x), \sin(x)).$$

Although $f(x)$ does not belong to $\operatorname{span}\{\cos(x), \sin(x)\}$, it was possible to find the approximation,

$$7.603 - 6.612\cos(x) + 2.267\sin(x),$$

which fits $f(x)$ with a little error. The fact that $f(x)$ is not in the span of $\{\cos(x), \sin(x)\}$ can be seen in the bottom graph. The blue line, which represents the regression function [3], is not belonging to any plane [4]. Nevertheless, the plane

$$\{(\phi_1, \phi_2, f) \mid f = 7.603 - 6.612\phi_1 + 2.267\phi_2\} \quad \text{(shown in red)}$$

[3] The points $(\cos(x), \sin(x), f(x))$ in the graph.

[4] $\notin \{(\phi_1, \phi_2, f) \mid f = a_0 + a_1\phi_1 + a_2\phi_2\}$, $\forall a_0, a_1, a_2 \in \mathbb{R}$.

is close to it.

Example 4. Figure 2.4 is the last example of projection into a feature space. It is an interesting example for a better visualization of the linearity in the feature space. It is easier to see the linearity in one dimension (i.e., points in a line) than in two dimensions (i.e., points in a plane). In contrast to the examples in figures 2.2 and 2.3 where the inputs are 1-dimensional values and the feature space is 2-dimensional, in this case the feature space is also 1-dimensional: $\Phi(x) = \cos(x - 2.8)$. The top graph shows the regression function (blue), 10 training points (black), the best linear approximation in the inputs (red) and the best linear approximation in the feature space $\Phi(x)$ (magenta). The bottom graph shows the training points projections into $\Phi(x)$, i.e., $(\cos(x_i^t - 2.8), f(x_i^t))$ for all training points x_i^t, $i = \{1, \ldots, 10\}$. In contrast with the top graph, it can be seen how the projections into $\Phi(x)$ almost lie in a straight line, and therefore the linear model applied to that feature espace suceeds. It is also an interesting example because it is showing that projecting into a feature space can improve the linear model even if the dimension of the feature space is not higher that the dimension of the inputs.

Figures 2.2 and 2.3 show examples of projections from a 1-dimensional input space to a 2-dimensional feature space. Although a 2-dimensional feature space was chosen due to visualization restrictions, the dimension could be far higher. Even infinite, which is the most interesting case (the non-degenerate case, see definition 2.1.24). However, operations with vectors in the feature space will become more expensive computationally when the dimension is increased. Here is when the kernel trick (2.1.3) comes into the picture. Inner products in feature spaces can be computed with a little computational cost using kernels. Even projecting into infinite-dimsional feature spaces. This is a tremendous advantage because it will allow algorithms to substitute inner products in the feature space for kernel evaluations. The example of using the kernel trick in a particular regression algorithm will be covered in section 2.2.

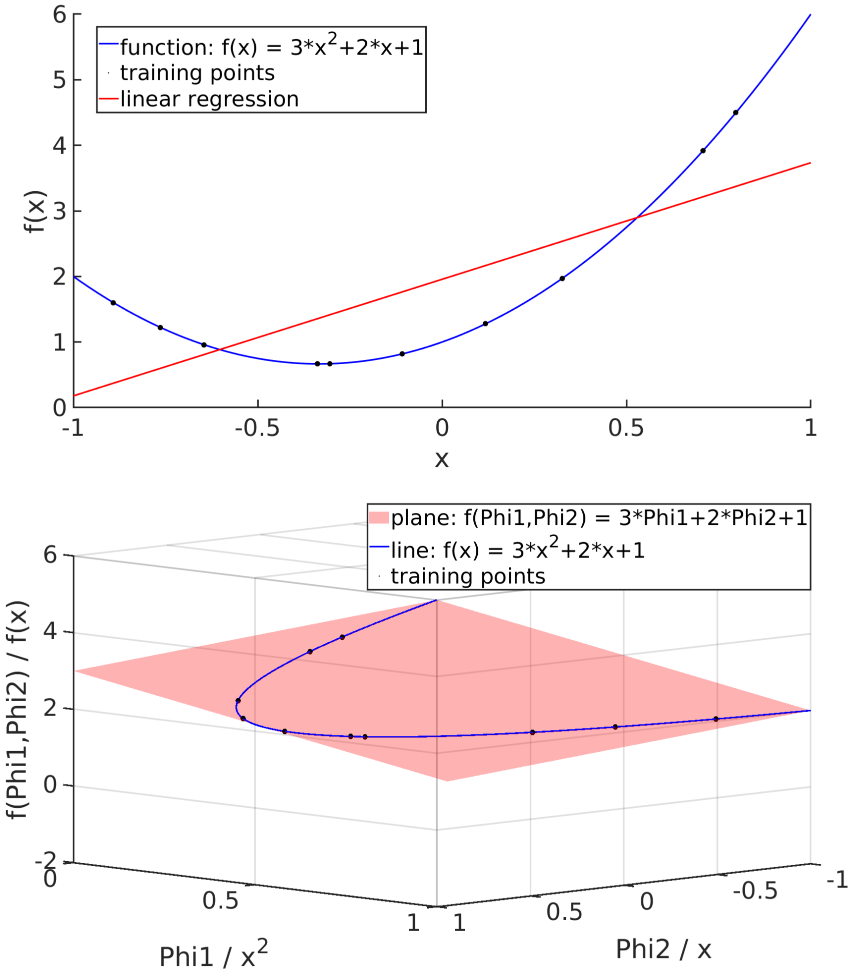


FIGURE 2.2: Example of a projection into a feature space $\Phi(x) = (x^2, x)$. *Blue line*: $f(x) = 3x^2 + 2x + 1$, two degree polynomial as a regression function. *Black points*: Training points. Ten evaluations of $f(x)$. *red line*: Best linear approximation (in the least squares sense). *red plane*: Plane $3\phi_1 + 2\phi_2 + 1$ containing the regression function projections into the feature space $\Phi(x) = (\phi_1(x), \phi_2(x)) = (x^2, x)$. The figure illustrates an example where a linear model on the inputs fails while a linear model on the feature space $\Phi(x) = (x^2, x)$ succeeds with a perfect fitting.

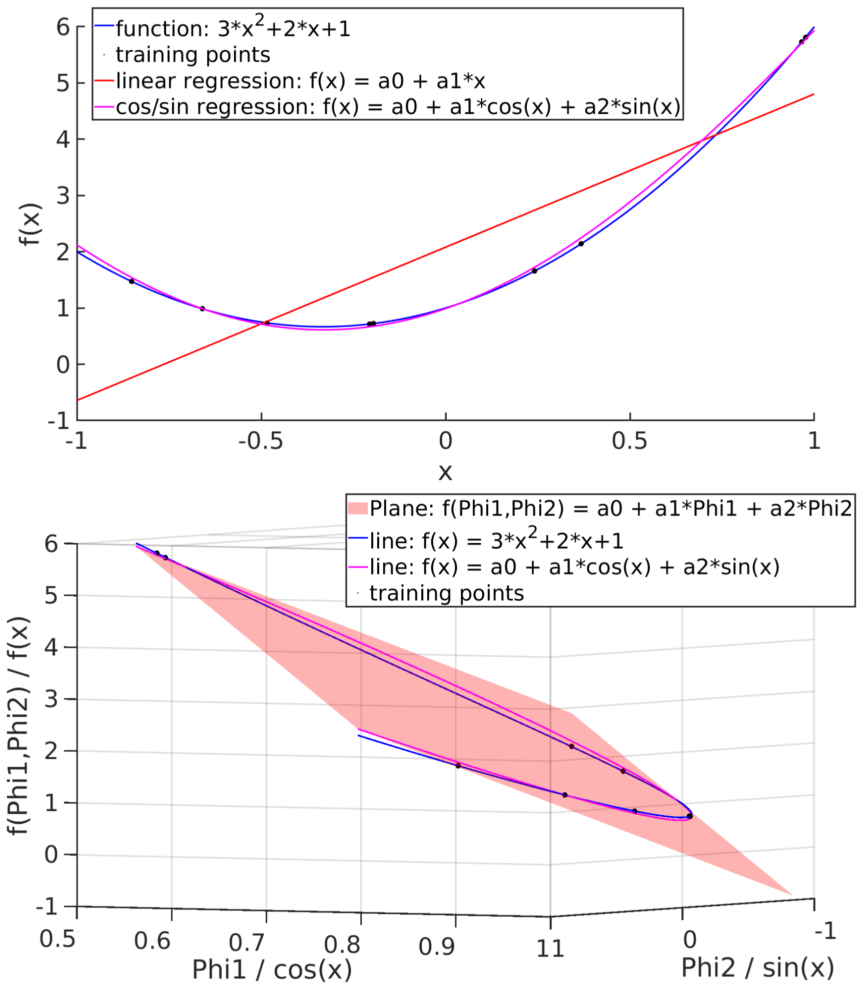


FIGURE 2.3: Example of a projection into a feature space $\Phi(x) = (\cos(x), \sin(x))$. *Blue line*: $f(x) = 3x^2 + 2x + 1$, two degree polynomial as a regression function. *Black points*: Training points. Ten evaluations of $f(x)$. *red line*: Best linear approximation (in the least squares sense). *magenta line*: Best approximation (in the least squares sense) considering the model $f(x) = a_0 + a_1 \cos(x) + a_2 \sin(x)$. The parameters which give the best fitting are: $a_0 = 7.603$, $a_1 = -6.612$ and $a_2 = 2.267$. *red plane*: Plane $a_0 + a_1\phi_1 + a_2\phi_2$ containing the regression function projections into the feature space $\Phi(x) = (\phi_1(x), \phi_2(x)) = (\cos(x), \sin(x))$. The figure illustrates an example where a linear model on the inputs fails while a linear model on the feature space $\Phi(x) = (\cos(x), \sin(x))$ succeeds with a little error.

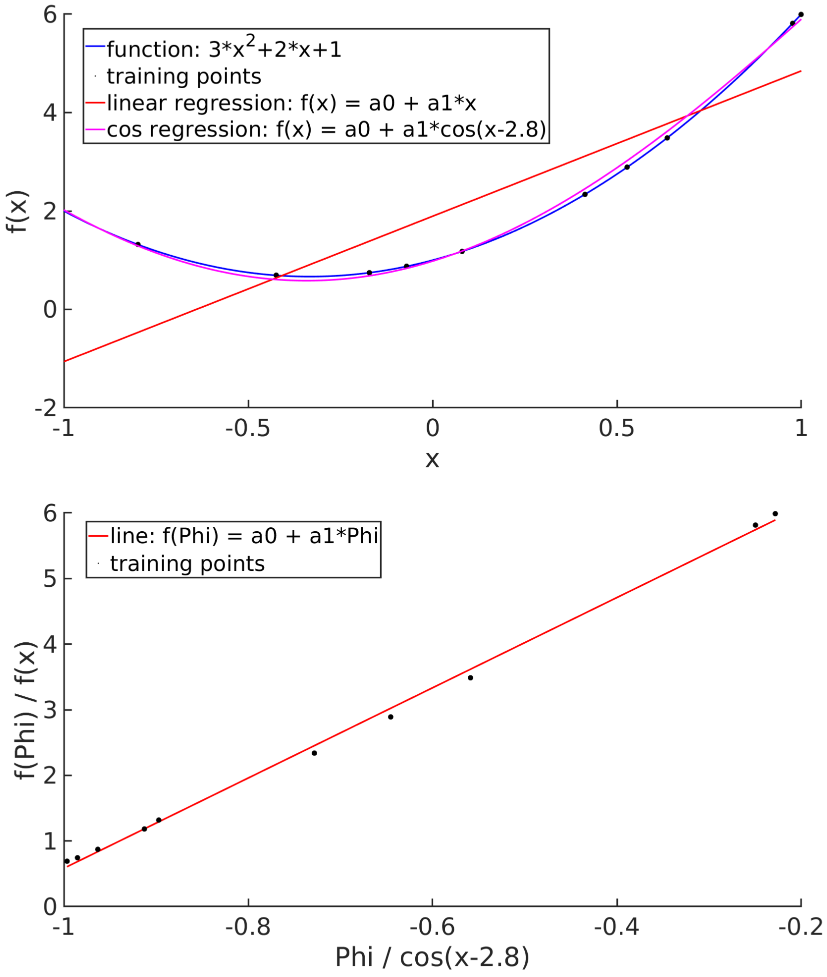


FIGURE 2.4: Example of a projection into a feature space $\Phi(x) = cos(x - 2.8)$. *Blue line*: $f(x) = 3x^2 + 2x + 1$, two degree polynomial as a regression function. *Black points*: Training points. Ten evaluations of $f(x)$. *red line (top)*: Best linear approximation (in the least squares sense). *magenta line*: Best approximation (in the least squares sense) considering the model $f(x) = a_0 + a_1 \cos(x - 2.8)$. The parameters which give the best fitting are: $a_0 = 7.462$ and $a_1 = 6.878$. *red line (bottom)*: Line $a_0 + a_1\Phi$ which approximates the training points projections into the feature space $\Phi(x) = \cos(x - 2.8)$. The figure illustrates an example where a linear model on the inputs fails while a linear model on the feature space $\Phi(x) = cos(x - 2.8)$ succeeds with a little error.

2.1.5.5 Characterization of RKHS spaces

Definition 2.1.25. *Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel. C_K is the supremum of $|K(x,t)|$, i.e.,*

$$C_K = \sup_{x,t \in \mathcal{X}} |K(x,t)|.$$

Definition 2.1.26. *Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a Mercer Kernel (definition 2.1.21). Define*

$$\begin{aligned} K_x : \mathcal{X} &\to \mathbb{R} \\ t &\mapsto K_x(t) = K(x,t). \end{aligned}$$

Theorem 2.1.4 (Moore–Aronszajn). *Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a Mercer Kernel. The Hilbert space $\mathcal{H}_K$ of functions on $\mathcal{X}$ satisfying,*

1. *$K_x \in \mathcal{H}_K, \ \forall x \in \mathcal{X}$, and the span of the set $\{K_x | x \in \mathcal{X}\}$ is dense in $\mathcal{H}_K$;*

2. *$f(x) = \langle K_x, f \rangle_{\mathcal{H}_K}, \ \forall f \in \mathcal{H}_K$;*

exists and it is unique. In addition, any function $f \in \mathcal{H}_K$ is continuous and the operator norm of the inclusion

$$I_K : \mathcal{H}_K \to \mathcal{C}(\mathcal{X}),$$

is bounded by $\sqrt{C_K}$, i.e.,

$$\|I_K\| \leq \sqrt{C_K}.$$

Proof. Full proof of this theorem including all the mathematical details can be found in Cucker and Zhou (2007). Let H_0 be the span of $\{K_x | x \in \mathcal{X}\}$. If

$$f = \sum_{i=1}^{s} \alpha_i K_{x_i} \in H_0 \quad \text{and} \quad g = \sum_{j=1}^{r} \beta_j K_{t_j} \in H_0,$$

consider the inner product in H_0 given by,

$$\langle f, g \rangle = \sum_{\substack{1 \leq i \leq s \\ 1 \leq j \leq r}} \alpha_i \beta_j K(x_i, t_j). \tag{2.21}$$

It can be proved that this inner product is well defined (see, e.g., Cucker and Zhou (2007)). Let $\mathcal{H}_K$ be the completion of H_0. Notice that $\mathcal{H}_K$ satisfies the two conditions above. Condition 1 is trivial and condition 2 is straightforward: if

$$f = \sum_{i=1}^{s} \alpha_i K_{t_i} \in H_0 \quad \text{and} \quad x \in \mathcal{X},$$

then

$$\langle K_x, f\rangle = \sum_{i=1}^{s} \alpha_i K(t_i, x) = f(x).$$

Therefore, the existance is proved.

Regarding the uniqueness, let H be another Hilbert space satisfying the conditions above. First, notice that $\mathcal{H}_K$ and H are completions of H_0 due to the condition 1. Thus, for the uniqueness of the completion $\mathcal{H}_K = H$. Secondly,

$$\langle K_x, K_t\rangle_H = K(x,t) = \langle K_x, K_t\rangle_{\mathcal{H}_K}, \ \ \forall x, t \in \mathcal{X},$$

due to the condition 2 and, by linearity,

$$\langle f, g\rangle_H = \langle f, g\rangle_{\mathcal{H}_K},$$

where,

$$f, g \in \mathcal{H}_K = H,$$

i.e., f and g belong to the completion of $H_0 = \text{span}\{K_x | x \in \mathcal{X}\})$. Therefore, $\langle,\rangle_H = \langle,\rangle_{\mathcal{H}_K}$.

With regards to the last part of the statement,

$$|f(x)| = |\langle K_x, f\rangle_{\mathcal{H}_K}| \leq \|K_x\|_{\mathcal{H}_K} \|f\|_{\mathcal{H}_K} \leq C_K \|f\|_{\mathcal{H}_K}, \ \forall f \in \mathcal{H}_K, \forall x \in \mathcal{X}$$

Hence $\|f\|_\infty \leq C_K \|f\|_{\mathcal{H}_K}$. Therefore, $\|I_K\| \leq C_K$ and convergence in $\mathcal{H}_K$ implies convergence in $\mathcal{C}(\mathcal{X})$ (definition 2.1.6). For the completeness of $\mathcal{C}(\mathcal{X})$ and the continuity of the functions K_x, $x \in \mathcal{X}$, any $f \in \mathcal{H}_K$ is continuous. □

Remark 20. Given a Mercer kernel, the Hilbert space $\mathcal{H}_K$ of functions on $\mathcal{X}$ described in theorem 2.1.4 is called the RKHS associated with the kernel K.

Remark 21. The condition 2 in theorem 2.1.4 is called the *reproducing property* of the RKHS $\mathcal{H}_K$.

Remark 22. The existence of a unique Hilbert space of functions on $\mathcal{X}$ satisfying the reproducing property in theorem 2.1.4 first apperead in Aronszajn (1950). N. Aronszajn developed a significant proportion of the RKHS general theory. Nevertheless, he attributed this theorem to E.H. Moore.

Remark 23. The Hilbert space $\mathcal{H}_K$ defined in theorem 2.1.4 is independent of any measure considered on $\mathcal{X}$. It is defined as the completion of $H_0 = \operatorname{span}\{K_x | x \in \mathcal{X}\}$, and therefore it only depends on the kernel K and the compact domain $\mathcal{X}$. However, theorem 2.1.5 shows that $\mathcal{H}_K$ can also be defined from the eigenfunctions $\{\phi_k\}_{k\geq 1}$ and the eigenvalues $\{\lambda_k\}_{k\geq 1}$ of the linear operator L_K (definition 2.1.18), which depends on a measure ν on $\mathcal{X}$.

Theorem 2.1.5. *Let $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a Mercer kernel. Let $\{\phi_k\}_{k\geq 1}$ and $\{\lambda_k\}_{k\geq 1}$ the eigenfunctions and the eigenvalues of the associated linear operator L_K. The Hilbert space of functions $\mathcal{H}_K$ defined in theorem 2.1.4, i.e., the completion of the span of $\{K_x | x \in \mathcal{X}\}$ with the inner product defined in (2.21), and the Hilbert space of functions,*

$$\mathcal{H}'_K = \left\{ f \in \mathcal{L}^2_\nu(\mathcal{X}) \mid f = \sum_{k=1}^{\infty} a_k\phi_k, \left(\frac{a_k}{\sqrt{\lambda_k}}\right) \in l^2 \right\},$$

with the inner product,

$$f = \sum_{k=1}^{\infty} a_k\phi_k,\ g = \sum_{k=1}^{\infty} b_k\phi_k \Rightarrow \langle f, g\rangle_{\mathcal{H}'_K} = \sum_{k=1}^{\infty} \frac{a_k b_k}{\lambda_k},$$

are the same space of functions with the same inner product.

Proof. First part of the proof is to show that functions in $\mathcal{H}'_K$ are continuous and if $f = \sum_{k=1}^{\infty} a_k\phi_k$ then the partials sums $\sum_{k=1}^{s} a_k\phi_k$ converge absolutely and uniformly to f when s tends to ∞. The details of this and the second part explained below can be found in Cucker and Smale (2001).

The second part is to prove the two conditions in theorem 2.1.4. The reproducing property is the most interesting for this work because it can be proved with the propositions and theorems developed in this Chapter. First, note that,

$$K_x(t) = K(x,t) = \sum_{k=1}^{\infty} \underbrace{\lambda_k\phi_k(x)}_{a_k} \phi_k(t)$$

by Mercer's theorem (theorem 2.1.2). In addition,

$$\left(\frac{\lambda_k \phi_k(x)}{\sqrt{\lambda_k}}\right)_{k\geq 1} \in l^2$$

by theorem 2.1.3. Thus $K_x \in \mathcal{H}'_K$, $\forall x \in \mathcal{X}$, and, for $f \in \mathcal{H}'_K$, $f = \sum_{k=1}^{\infty} a_k \phi_k$,

$$\begin{aligned}
\langle f, K_x \rangle_{\mathcal{H}'_K} &= \sum_{k=1}^{\infty} \langle a_k \phi_k, K_x \rangle_{\mathcal{H}'_K} \\
&= \sum_{k=1}^{\infty} \frac{a_k \langle \phi_k, K_x \rangle_{\mathcal{L}^2_\nu}}{\lambda_k} \qquad (\{\phi_k\}_{k\geq 1} \text{ orthonormal basis}) \\
&= \sum_{k=1}^{\infty} \frac{a_k}{\lambda_k} \int_{\mathcal{X}} \phi(t) K(x,t) d\nu(t) \qquad (\text{definition 2.1.17}) \\
&= \sum_{k=1}^{\infty} \frac{a_k}{\lambda_k} (L_K \phi_k)(x) \qquad (\text{definition 2.1.18}) \\
&= \sum_{k=1}^{\infty} \frac{a_k}{\lambda_k} \lambda_k \phi_k(x) \qquad \begin{array}{c}(\{\phi_k\}_{k\geq 1} \text{ and } \{\lambda_k\}_{k\geq 1} \text{ eigenfunctions} \\ \text{and eigenvalues of } L_K)\end{array} \\
&= f(x).
\end{aligned}$$

It only remains to prove that $\{K_x | x \in \mathcal{X}\}$ is dense in $\mathcal{H}'_K$. Again, refer to Cucker and Smale (2001) for the details.

Finally, $\mathcal{H}_K = \mathcal{H}'_K$ by the uniqueness of $\mathcal{H}_K$ proved in theorem 2.1.4. □

Alternatively, the RKHS spaces can be defined as follows.

Definition 2.1.27 (RKHS). *A Hilbert space of functions $\mathcal{H}$ on a compact set $\mathcal{X} \subset \mathbb{R}^N$ with the following property: the functions*

$$\begin{aligned} F_x : \mathcal{H} &\to \mathcal{Y} \\ f &\mapsto f(x) \end{aligned}, \quad \forall x \in \mathcal{X},$$

are continuous linear functionals, is called Reproducing Kernel Hilbert Space (RKHS).

Remark 24. It can be proved that definition 2.1.27 of a RKHS and the definition given by theorem 2.1.4 are equivalent, i.e., any RKHS space according to definition 2.1.27 can be given by a Mercer kernel following theorem 2.1.4 and vice versa (see, e.g., Hofmann, Schölkopf, and Smola (2008)).

2.1.5.6 Representer theorem

The ultimate goal of this Chapter is to provide a tool to tackle the stated regression problem. That means providing an algorithm which is computationally tractable, i.e., an algorithm consisting of a finite number of operations which modern computers can run in a reasonable time.

In this section, RKHS spaces were presented as spaces of functions $\mathcal{H}_K \subset \mathcal{L}^2_\nu$ completely defined by their respective kernels K and the input space $\mathcal{X}$. In the regression problem, the idea is to look for approximations of the regression function in those special spaces. In the non-degenerate case (2.1.24), i.e., when the RKHS is infinite-dimsional, the problem is intractable by computers (a priori) because the search can not be done with finite memory in a finite time.

The kernel trick (theorem 2.1.3) already gives a powerful tool for designing algorithms when the RKHS is infinite-dimsional. Next theorem 2.1.6 is also a remarkable result regarding infinite-dimensionality of the RKHS. It states that a minimizer of the regularized empirical error (definition 2.1.14) $f_{\mathbf{z},\gamma}$ in a RKHS is a finite linear combination of kernel evaluations $\sum_{i=1}^n \alpha_i K_{x_i}$ where $\{x_i\}_{1\leq i\leq n}$ are the inputs of the sample (or training set) $\mathbf{z} = \{(x_i, y_i)\}_{1\leq i\leq n}$. Therefore, it is not only proving that the infinite-dimsional minimization problem of finding $f_{\mathbf{z},\gamma}$ in a RKHS can actually be reduced to a computationally tractable problem, but also providing a method to do it: solving,

$$\underset{\alpha=(\alpha_1,\ldots,\alpha_n)}{\operatorname{argmin}} \ \mathcal{E}_{\mathbf{z},\gamma}\left(\sum_{i=1}^n \alpha_i K_{x_i}\right).$$

Theorem 2.1.6 (Representer theorem). *Let $\mathbf{z} = \{(x_i, y_i)\}_{1\leq i\leq n}$ be a sample of ρ (training data) and $\gamma > 0$. Let $\mathcal{H}_K$ be the RKHS associated with the Mercer kernel K. A empirical target function $f_{\mathbf{z},\gamma}$ minimizing the regularized empirical error $\mathcal{E}_{\mathbf{z},\gamma}$ (definition 2.1.14) in $\mathcal{H}_K$ can be written as,*

$$f_{\mathbf{z},\gamma}(x) = \sum_{i=1}^n \alpha_i K_{x_i}(x),$$

where $\alpha_i \in \mathbb{R}$ for all $1 \leq i \leq n$.

Proof. Let $\{\phi_k\}_{k\geq 1}$ be eigenfunctions of L_K which form an orthonormal basis of $\mathcal{L}^2_\nu$ (theorem 2.1.1). Let $\{\lambda_k\}_{k\geq 1}$ be the corresponding eigenvalues. If $f = \sum_{k=1}^{\infty} c_k \phi_k$ is a function in $\mathcal{H}_K$ (remark 15), then,

$$\|f\|_{\mathcal{H}_K}^2 = \sum_{k=1}^{\infty} \frac{c_k^2}{\lambda_k},$$

by theorem 2.1.5. Therefore,

$$\mathcal{E}_{\mathbf{z},\gamma}(f) = \mathcal{E}_{\mathbf{z},\gamma}(\sum_{k=1}^{\infty} c_k \phi_k) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{k=1}^{\infty} c_k \phi_k(x_i) \right)^2 + \gamma \sum_{k=1}^{\infty} \frac{c_k^2}{\lambda_k}. \tag{2.22}$$

The aim is to find the coefficients $\{c_k\}_{k \geq 1}$ which minimize (2.22). Thus, solving $\frac{\partial \mathcal{E}_{\mathbf{z},\gamma}}{\partial c_k} = 0$ for all $k \geq 1$,

$$0 = \frac{\partial \mathcal{E}_{\mathbf{z},\gamma}}{\partial c_k} = \frac{-2}{n} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{\infty} c_j \phi_j(x_i) \right) \phi_k(x_i) + 2\gamma \left(\frac{c_k}{\lambda_k} \right).$$

Therefore, the coefficients $\{c_k\}_{k \geq 1}$ which minimize (2.22) are,

$$c_k = \lambda_k \sum_{i=1}^{n} \underbrace{\frac{1}{\gamma n} \left(y_i - \sum_{j=1}^{\infty} c_j \phi_j(x_i) \right)}_{\alpha_i} \phi_k(x_i), \tag{2.23}$$

and,

$$\begin{aligned} f_{\mathbf{z},\gamma}(x) &= \sum_{k=1}^{\infty} c_k \phi_k(x) \\ &= \sum_{k=1}^{\infty} \left(\lambda_k \sum_{i=1}^{n} \alpha_i \phi_k(x_i) \right) \phi_k(x) \text{ (by (2.23))} \\ &= \sum_{i=1}^{n} \alpha_i \sum_{k=1}^{\infty} \lambda_k \phi_k(x_i) \phi_k(x) \\ &= \sum_{i=1}^{n} \alpha_i K_{x_i}(x) \text{ (by theorem 2.1.2).} \end{aligned}$$

□

A first version of the Representer theorem (theorem 2.1.6) for the special case of cubic splines can be found in Schoenberg (1964). Kimeldorf and Wahba (1971) extended to RKHS (as it is stated in this work). A first generalization of the statement in theorem 2.1.6 can be found in O'sullivan, Yandell, and William (1986). However, Schölkopf, Herbrich, and Smola (2001) generalize it even further, relaxing the hypothesis with these two changes:

1. an arbitrary strictly increasing real-valued function $g : [0, \infty) \to \mathbb{R}$ is used to define the regularized term $g(\|f\|_{\mathcal{H}_K})$. Theorem 2.1.6 would be a special case where,

$$g(\|f\|_{\mathcal{H}_K}) = \gamma\|f\|^2_{\mathcal{H}_K}.$$

2. an arbitrary function

$$\begin{aligned} E : (\mathcal{X} \times \mathbb{R}^2)^n &\to \mathbb{R} \cup \infty \\ (x, y, f(x)) &\mapsto E(x, y, f(x)) \end{aligned}$$

is used as empirical error. The empirical error defined in this work,

$$\mathcal{E}_{\boldsymbol{x}}(f) = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2,$$

would be the special case in theorem 2.1.6.

Next section 2.2 will describe in detail an algorithm to tackle regression in RKHS spaces using Gaussian Processes. Nevertheless, it is interesting to show an answer for the regression problem using the tools explained so far. A minimizer of the regularized empirical error can be derived as an immediate consequence of theorem 2.1.6. Corollary 2.1.2 shows the details.

Corollary 2.1.2 (A minimizer of $\mathcal{E}_{\mathbf{z},\gamma}$ in $\mathcal{H}_K$). *Let $\mathbf{z} = \{(x_i, y_i)\}_{1\leq i\leq n}$ be a sample of ρ (training data) and $\gamma > 0$. Let $\mathcal{H}_K$ be the RKHS associated with the Mercer kernel K. A empirical target function $f_{\mathbf{z},\gamma}$ minimizing the regularized empirical error $\mathcal{E}_{\mathbf{z},\gamma}$ (definition 2.1.14) in $\mathcal{H}_K$ can be expressed as,*

$$f_{\mathbf{z},\gamma}(x) = \sum_{i=1}^{n}\alpha_i K_{x_i}(x),$$

where $\alpha_i \in \mathbb{R}$ is the solution of the n-dimensional linear system,

$$y = (K[\boldsymbol{x}] + \gamma n Id_n)\,\alpha, \;\; \textit{with} \begin{cases} y = (y_1, \ldots, y_n), \\ \alpha = (\alpha_1, \ldots, \alpha_n), \\ \boldsymbol{x} = (x_1, \ldots, x_n), \\ K[\boldsymbol{x}] \textit{ defined in 2.15 and} \\ Id_n \textit{ the n-dimensional identity matrix.} \end{cases} \tag{2.24}$$

Proof. The proof is immediate using theorem 2.1.6. From expression (2.23),

$$\alpha_i = \frac{1}{\gamma n}\left(y_i - f(x_i)\right),$$

and using $f(x_i) = \sum_{j=1}^{n} \alpha_j K_{x_j}(x_i)$,

$$y_i = \sum_{j=1}^{n} \alpha_j K_{x_j}(x_i) + \gamma n \alpha_i.$$

Therefore,

$$\begin{aligned} \boldsymbol{y} &= K[\boldsymbol{x}]\alpha + \gamma n \alpha \\ &= \left(K[\boldsymbol{x}] + \gamma n Id_n\right)\alpha. \end{aligned}$$

□

Remark 25. Note that $K[\boldsymbol{x}]$ is a positive-definite matrix since K is a Mercer Kernel (definition 2.1.21) and due to the remark 13. Therefore, $K[\boldsymbol{x}] + \gamma n Id_n$ is also positive-definite and the solution of the linear system (2.24) exists and it is unique. The positive-definiteness of the matrix also gives some advantages in order to solve the system. For example, the conjugate gradients iterative method or Cholesky decomposition (proposition 3.1.3) can be applied.

Remark 26. If data $\boldsymbol{z} = \{(x_i, y_i)\}_{1\leq i\leq n}$ is provided and a Mercer Kernel K is chosen, then corollary 2.1.2 transform the infinite-dimsional optimization problem of finding a minimizer of the regularized empirical error (definition 2.1.14) into a linear system, which is tractable by computers.

Remark 27. Notice that the solution provided by Gaussian Process Regression in (2.42) is consistent with the corollary 2.1.2. This is a manifestation of theorem 2.1.6 and the fact that Gaussian Process Regression is indeed providing a minimizer of a regularized empirical error. Remark 41 in next section 2.2 gives the details.

2.2 Gaussian Process Regression

This section gives a review on Gaussian Process regression. Most of the ideas can be found in Rasmussen and Williams (2006). However, the mathematical steps to build the posterior predictive distribution have been explained in more detail.

Section 1.4 gives some notation clarifications. It is recommended to read it before continuing with this section.

2.2.1 The measure governing the samples

At the beginning of section 2.1, it was pointed out that the probability measure ρ governing the samples (or training set) $\mathbf{z} = \{(x_i, y_i)\}_{1 \leq i \leq n}$ is unknown. Yet, an assumption will be made in Gaussian Process regression. The conditional probability measure of ρ on $\mathcal{Y}$ conditioned to $x \in \mathcal{X}$ will be assumed to be a Gaussian distribution with mean equal to the regression function (definition 2.1.14) evaluated at x and a variance $\sigma^2(x)$ (definition 2.1.2), i.e.,

$$\rho(y|x) \sim \mathcal{N}(f_\rho(x), \sigma^2(x)). \tag{2.25}$$

Therefore, given an input $x \in \mathcal{X}$, the output $y \in \mathcal{Y}$ can be thought as an evaluation of the regression function f_ρ at x plus Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2(x))$,

$$y = f_\rho(x) + \epsilon.$$

Remark 28. Let $\rho_{\mathcal{X}}$ be the marginal probability of ρ on $\mathcal{X}$. If $\rho_{\mathcal{X}}$ is known, then it is only remaining to know the regression function f_ρ to have ρ completely specified. This is because $\rho(y|x)$ and $\rho_{\mathcal{X}}(x)$ completely specify $\rho(x, y)$ (see 2.1.1).

Let $\mathcal{X} \subset \mathbb{R}^N$ be a compact set. If $\rho_{\mathcal{X}}(x)$ is assumed to be uniform on $\mathcal{X}$ and $\rho(y|x)$ is assumed to be as (2.25), then it is only necessary to know $f_\rho(x)$ and $\sigma^2(x)$ to have the measure ρ, which governs the samples, completely specified.

2.2.2 The linear model

Definition 2.2.1 (Design matrix). *If $\mathcal{X} \subset \mathbb{R}^N$ is the input space and $\mathbf{z} = \{(x_i, y_i)\}_{1 \leq i \leq n}$ is the training set, then $\mathbf{X}$ will be the $N \times n$ matrix whose ith column is the vector x_i, i.e.,*

$$\mathbf{X} = [x_1 \ldots x_n] \in \mathbb{R}^{N \times n}.$$

Let $\mathcal{H}_l \subset \mathcal{L}^2_\nu(\mathcal{X})$ be the hypothesis space of linear functions on $\mathcal{X} \subset \mathbb{R}^N$. Therefore, for any function $f_w \in \mathcal{H}_l$ there exists a vector of weights $w = (w_1, \dots, w_N)^T$ such that,

$$f_w(x) = w^T x, \ \forall x \in \mathcal{X}.$$

If the approach of last section 2.1 is followed, the procedure to make predictions would be finding a minimizer of the empirical error in $\mathcal{H}_l$. This is straight forward, it is only necessary to solve the minimization problem on the weights w (see remark 31). Instead of doing that, the approach followed in this section will be the use of Bayesian analysis to find a posterior on the weights w (i.e., on the functions $f_w \in \mathcal{H}_l$) given the training set $\boldsymbol{z} = \{(x_i, y_i)\}_{1 \le i \le n}$ (i.e., given $\boldsymbol{x}$ and $\boldsymbol{y}$),

$$p(w|\boldsymbol{y}, \boldsymbol{x}) = \frac{p(\boldsymbol{y}|w, \boldsymbol{x})p(w)}{p(\boldsymbol{y}|\boldsymbol{x})}. \tag{2.26}$$

Therefore, it is necessary to compute the likelihood $p(\boldsymbol{y}|w, \boldsymbol{x})$ and set a prior $p(w)$ on the weights. The marginal $p(\boldsymbol{y}|\boldsymbol{x})$ is only a normalizing constant. It does not depend on w.

Remark 29. Although a new approach is presented, after applying the linear model to the projections to a feature space $\Phi(x)$ and changing, therefore, the hypothesis space $\mathcal{H}_l$ to a RKHS, it will be seen that the kernel trick (theorem 2.1.3) can be exploited and the solution given by the GP regression is consistent with the Representer theorem (theorem 2.1.6). Those are manifestation of the strong connections between GP regression and the theory explained in the last section 2.1.

2.2.2.1 The likelihood and the prior

Given an arbitrary vector of weights $w \in \mathbb{R}^N$ and assuming that f_w is the regression function in (2.25),

$$p(y|x, w) = \mathcal{N}(w^T x, \sigma^2(x)) = \frac{1}{\sqrt{2\pi\sigma^2(x)}} \exp\left(\frac{-(y - w^T x)^2}{2\sigma^2(x)}\right). \tag{2.27}$$

Thus, if

- $\boldsymbol{x} = \{x_i\}_{1 \le i \le n}$ are the input points and $\boldsymbol{y} = (y_1, \dots, y_n)^T$ is the vector of outputs in the training set $\boldsymbol{z}$,

- the samples in $\boldsymbol{z}$ are i.i.d. and,

- the variance $\sigma^2(x)$ is assumed to be constant in the input points of the training set, i.e., $\sigma^2(x_i) = \sigma^2, 1 \leq i \leq n$,

then,

$$\begin{aligned}
&p(\boldsymbol{y}|\boldsymbol{x}, w) = \\
&= \prod_{i=1}^{n} p(y_i|x_i, w) \\
&= \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(y_i - w^T x_i)^2}{2\sigma^2}\right) \qquad \text{(by 2.27)} \\
&= \frac{1}{(2\pi)^{\frac{n}{2}} \left(\det(\sigma^2 I_{n\times n})\right)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}^T w)^T (\sigma^2 I_{n\times n})^{-1} (\boldsymbol{y} - \boldsymbol{X}^T w)\right) \\
&= \mathcal{N}(\boldsymbol{X}^T w, \sigma^2 I_{n\times n}). \quad \text{(definition of multivariate Normal density)}
\end{aligned} \tag{2.28}$$

Therefore, the likelihood of (2.26) is already computed in (2.28). It only remains to set a prior on w.

Let $\mathcal{N}(0, \Sigma_w)$ be a Gaussian prior on w with zero mean and covariance matrix Σ_w, i.e,

$$p(w) = (2\pi)^{-\frac{n}{2}} \det(\Sigma_w)^{-\frac{1}{2}} \exp\left(-\frac{w^T \Sigma_w^{-1} w}{2}\right). \tag{2.29}$$

Remark 30. It has been assumed a constant variance $\sigma^2(x_i) = \sigma^2$. The adoption of this assumption in a regression model is called homoscedastic regression. In contrast, the terminology heteroscedastic regression is used when the variance is assumed not to be constant. Heteroscedastic GP regression has been studied. See, e.g., Antunes et al. (2017); or Lázaro-Gredilla and Titsias (2011).

2.2.2.2 The posterior on the weights

Computing the product of these two normal distributions, likelihood (2.28) and prior (2.29),

$$\begin{aligned}
p(w|\boldsymbol{y}, \boldsymbol{x}) &\propto p(\boldsymbol{y}|w, \boldsymbol{x}) p(w) \\
&\propto \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{X}^T w\|^2}{2\sigma^2}\right) \exp\left(-\frac{w^T \Sigma_w^{-1} w}{2}\right) \\
&\propto \exp\left(-\frac{1}{2}(w - \bar{w})^T \left(\frac{1}{\sigma^2}\boldsymbol{X}\boldsymbol{X}^T + \Sigma_w^{-1}\right)(w - \bar{w})\right),
\end{aligned}$$

where

$$\bar{w} = \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}\boldsymbol{X}\boldsymbol{X}^T + \Sigma_w^{-1}\right)^{-1}\boldsymbol{X}\boldsymbol{y}.$$

Thus, the posterior on w is Gaussian with mean $\bar{w}$ and covariance matrix equal to the inverse of

$$A = \frac{1}{\sigma^2}\boldsymbol{X}\boldsymbol{X}^T + \Sigma_w^{-1},$$

i.e.,

$$p(w|\boldsymbol{y},\boldsymbol{x}) = \mathcal{N}(\bar{w}, A^{-1}). \tag{2.30}$$

Remember that the aim is to find an approximation of the regression function in $\mathcal{H}_l$, i.e., a vector of weights w which an associated linear function f_w that approximates f_ρ.

In a non-Bayesian approach, the choice of w could be the maximum a posteriori (MAP), i.e., the mode of the posterior (2.30). Since it is a Gaussian density, note that the mode is equal to the mean. Thus, the MAP would be $\bar{w}$.

Remark 31. If $\boldsymbol{X}^T$ is full column rank, it is possible to find the minimizer of the empirical error $\mathcal{E}_{\mathbf{z}}$ (definition 2.1.5) in $\mathcal{H}_l$ and it is unique. Note that,

$$\underset{w}{\operatorname{argmin}} \, \|\boldsymbol{y} - \boldsymbol{X}^T w\|^2 \tag{2.31}$$

can be solve with the ordinary least squares method with solution,

$$w_{\mathcal{E}_{\mathbf{z}}} = (\boldsymbol{X}\boldsymbol{X}^T)^{-1}\boldsymbol{X}\boldsymbol{y}.$$

Notice that $w_{\mathcal{E}_{\mathbf{z}}}$ corresponds to the MAP $\bar{w}$ in the limiting case of $\det(\Sigma_w)$ tending to infinity. Informally, although there is not uniform distribution on $\mathbb{R}^N$, this limiting case can be thought as the prior on the weights being uniform.

Taking the estimate $w_{\mathcal{E}_{\mathbf{z}}}$ in remark 31 and using $f_{w_{\mathcal{E}_{\mathbf{z}}}}$ in order to make predictions may lead to overfitting problems (see sections 2.1.4, 2.1.4.1 and 2.1.4.2). In contrast, the posterior predictive distribution,

$$\begin{aligned} p(y|x_*,\boldsymbol{y},\boldsymbol{x}) &= \int_{\mathbb{R}^N} p(y|x_*,w)p(w|\boldsymbol{y},\boldsymbol{x})dw \\ &= \mathcal{N}\left(\frac{1}{\sigma^2}x_*^T A^{-1}\boldsymbol{X}\boldsymbol{y}, x_*^T A^{-1} x_*\right) \\ &= \mathcal{N}\left(x_*^T\bar{w}, x_*^T A^{-1} x_*\right), \end{aligned} \tag{2.32}$$

of the Bayesian approach introduce some regularization in its solution (see remark 34). Given an input point $x_* \in \mathcal{X}$ which is not in the training set $\boldsymbol{z}$, instead of taking the best linear model $f(x_*) = x_*^T w_{\mathcal{E}_{\boldsymbol{z}}}$ according to the data $\boldsymbol{z}$ and the minimization problem (2.31), it averages all possible linear models with respect to the posterior (2.30), i.e., it is calculated marginalizing the weights.

Remark 32. Notice that the mean of (2.32) coincides with the MAP estimate $\bar{w}$ in order to make predictions. This fact is due to the Gaussian distribution symmetries in the model. Nevertheless, it can not be generalize in the Bayesian approach. In addition, the posterior predictive (2.32) gives more information than just $\bar{w}$. It is not only giving an approximation of the regression function $f_{\bar{w}}(x_*) = x_*^T \bar{w}$, but also a confidence interval in the predictions given by the variance $x_*^T A^{-1} x_*$.

Remark 33. Notice that the variance in (2.32) is independent of the training set outputs $\boldsymbol{y}$. This is a particular property of GP regression.

Remark 34 (Ridge regression). Consider the following regularization (see definition 2.1.14) of (2.31),

$$\mathcal{E}_{\boldsymbol{z},\gamma} = \min_{w} \|\boldsymbol{y} - \boldsymbol{X}^T w\|^2 + \gamma \|w\|^2.$$

The optimal solution in this case is,

$$w_{\mathcal{E}_{\boldsymbol{z},\gamma}} = \left(\boldsymbol{X}\boldsymbol{X}^T + \gamma I_{N\times N}\right)^{-1} \boldsymbol{X}\boldsymbol{y}.$$

The study of this minimization problem is called ridge regression (see Hoerl and Kennard (1970)). Further information about the parameter γ can be found in, for example, Golub, Heath, and Wahba (1979). Note that $\bar{w} = w_{\mathcal{E}_{\boldsymbol{z},\gamma}}$ for the particular case of $\Sigma_w = \gamma I_{N\times N}$. Matrices of the form of Σ_w instead of $I_{N\times N}$ can appear in ridge regression if the norm,

$$\|w\|^2 = w^T \Sigma_w^{-1} w,$$

is considered instead of the Euclidean norm.

Remark 35. The overfitting problem may not be a major issue in the limited hypothesis space of linear functions (linear on the inputs). In the next section 2.2.3, however, the linear model would be applied on a feature space. The overfitting problem should be considered in the "bigger" spaces resulting from those proyections.

2.2.3 The linear model on a feature space

The idea of projecting the inputs to a feature space was introduced in section 2.1.5.4. The goal of this section is to use the results of last section 2.2.2 and the kernel trick (theorem 2.1.3) to exploit the linear model on a feature space

$$\Phi(x) = (\phi_1(x), \phi_2(x), ...)^T, \ \ \forall x \in \mathcal{X}.$$

For the following derivations, suppose that $\Phi(x)$ is given and N_Φ-dimensional. However, after applying the kernel trick, it will be seen that the feature space does not need to be known (remark 42) and it can be infinite-dimsional (remark 40).

Remember the linear model applied to the input space $\mathcal{X}$,

$$\begin{aligned} y|x,w &\sim \mathcal{N}(w^T x, \sigma^2) && \text{Likelihood.} \\ w &\sim \mathcal{N}(0, \Sigma_w) && \text{Prior on the weights.} \\ &\downarrow && \\ w|\boldsymbol{y},\boldsymbol{x} &\sim \mathcal{N}(\bar{w}, A^{-1}) && \text{Posterior on the weights. See (2.30).} \\ y|x_*,\boldsymbol{y},\boldsymbol{x} &\sim \mathcal{N}\left(\tfrac{1}{\sigma^2} x_*^T A^{-1} \boldsymbol{X}\boldsymbol{y}, \right. && \\ & \qquad \left. x_*^T A^{-1} x_*\right) && \text{Posterior predictive. See (2.32).} \end{aligned} \tag{2.33}$$

Apply the linear model to the feature space instead,

$$\begin{aligned} y|x,w &\sim \mathcal{N}(w^T \Phi(x), \sigma^2) && \text{Likelihood.} \\ w &\sim \mathcal{N}(0, \Sigma_w) && \text{Prior on the weights.} \end{aligned} \tag{2.34}$$

Notice that model (2.34) will lead to the same derivations in (2.33), with the only difference of,

$$x \to \Phi(x),$$

and therefore,

$$\begin{aligned} \boldsymbol{X} &\to \Phi_{\boldsymbol{X}} = [\Phi(x_1) \dots \Phi(x_n)], \\ A &\to A_\Phi = \frac{1}{\sigma^2} \Phi_{\boldsymbol{X}} \Phi_{\boldsymbol{X}}^T + \Sigma_w^{-1}. \end{aligned} \tag{2.35}$$

Thus, the posterior predictive of model (2.34) is

$$y|x_*,\boldsymbol{y},\boldsymbol{x} \sim \mathcal{N}\left(\frac{1}{\sigma^2} \Phi(x_*)^T A_\Phi^{-1} \Phi_{\boldsymbol{X}} \boldsymbol{y} \ , \ \Phi(x_*)^T A_\Phi^{-1} \Phi(x_*)\right). \tag{2.36}$$

Lemma 2.2.1 (Woodbury matrix identity). *Let Z be a $m \times m$ matrix, W a $k \times k$ matrix and U and V are both $m \times k$ matrices, $m, k \in \mathbb{N}$. If $(Z + UWV^T)$, Z and W are invertible, then*

$$(Z + UWV^T)^{-1} = Z^{-1} - Z^{-1}U(W^{-1} + V^T Z^{-1} U)^{-1} V^T Z^{-1}.$$

Proof. It is only necessary to expand the product,

$$(Z + UWV^T)[Z^{-1} - Z^{-1}U(W^{-1} + V^T Z^{-1} U)^{-1} V^T Z^{-1}]$$

and see that it is equal to the $m \times m$ identity matrix. More details can be found in Woodbury (1950) or Press et al. (1992). □

Proposition 2.2.1. *The posterior predictive distribution 2.36 can be expressed as,*

$$\begin{aligned} y|x_*, \boldsymbol{y}, \boldsymbol{x} \sim \mathcal{N}\Bigg(&\Phi(x_*)^T \Sigma_w \Phi_{\mathbf{X}} \left(\Phi_{\mathbf{X}}^T \Sigma_w \Phi_{\mathbf{X}} + \sigma^2 I_{n\times n}\right)^{-1} \boldsymbol{y}, \\ &\Phi(x_*)^T \Sigma_w \Phi(x_*) \\ &-\Phi(x_*)^T \Sigma_w \Phi_{\mathbf{X}} \left(\Phi_{\mathbf{X}}^T \Sigma_w \Phi_{\mathbf{X}} + \sigma^2 I_{n\times n}\right)^{-1} \Phi_{\mathbf{X}}^T \Sigma_w \Phi(x_*)\Bigg). \end{aligned} \tag{2.37}$$

Proof. Regarding to the mean, notice that,

$$\begin{aligned} A_\Phi \Sigma_w \Phi_{\mathbf{X}} &= \left(\frac{1}{\sigma^2}\Phi_{\mathbf{X}}\Phi_{\mathbf{X}}^T + \Sigma_w^{-1}\right)\Sigma_w \Phi_{\mathbf{X}} \\ &= \frac{1}{\sigma^2}\Phi_{\mathbf{X}}\Phi_{\mathbf{X}}^T \Sigma_w \Phi_{\mathbf{X}} + \Phi_{\mathbf{X}} \\ &= \frac{1}{\sigma^2}\Phi_{\mathbf{X}}\left(\Phi_{\mathbf{X}}^T \Sigma_w \Phi_{\mathbf{X}} + \sigma^2 I_{n\times n}\right), \end{aligned}$$

and therefore,

$$\begin{aligned} &\Sigma_w \Phi_{\mathbf{X}} \left(\Phi_{\mathbf{X}}^T \Sigma_w \Phi_{\mathbf{X}} + \sigma^2 I_{n\times n}\right)^{-1} = \\ &= A_\Phi^{-1} \underbrace{(A_\Phi \Sigma_w \Phi_{\mathbf{X}})}_{\frac{1}{\sigma^2}\Phi_{\mathbf{X}}\left(\Phi_{\mathbf{X}}^T \Sigma_w \Phi_{\mathbf{X}} + \sigma^2 I_{n\times n}\right)} \left(\Phi_{\mathbf{X}}^T \Sigma_w \Phi_{\mathbf{X}} + \sigma^2 I_{n\times n}\right)^{-1} \\ &= \frac{1}{\sigma^2} A_\Phi^{-1} \Phi_{\mathbf{X}}. \end{aligned}$$

Thus,

$$\frac{1}{\sigma^2}\Phi(x_*)^T A_\Phi^{-1} \Phi_{\mathbf{X}} \boldsymbol{y} = \Phi(x_*)^T \Sigma_w \Phi_{\mathbf{X}} \left(\Phi_{\mathbf{X}}^T \Sigma_w \Phi_{\mathbf{X}} + \sigma^2 I_{n\times n}\right)^{-1} \boldsymbol{y}.$$

Regarding to the variance, apply the lemma 2.2.1 to invert the matrix A_Φ,

$$\begin{aligned}
A_\Phi^{-1} &= \left(\underbrace{\Sigma_w^{-1}}_{Z} + \underbrace{\Phi_{\mathbf{X}}}_{U} \underbrace{\frac{1}{\sigma^2} I_{n\times n}}_{W} \underbrace{\Phi_{\mathbf{X}}^T}_{V^T} \right)^{-1} \\
&= \Sigma_w - \Sigma_w \Phi_{\mathbf{X}} \left(\frac{1}{\sigma^2} I_{n\times n} + \Phi_{\mathbf{X}}^T \Sigma_w \Phi_{\mathbf{X}} \right)^{-1} \Phi_{\mathbf{X}}^T \Sigma_w .
\end{aligned}$$

□

Remark 36. Computing the mean and the variance of the posterior predictive distribution from (2.36) involves the inversion of a $N_\Phi \times N_\Phi$ matrix. In contrast, computing them from (2.37) involves the inversion of a $n \times n$ matrix. Remember that n is the number of samples and N_Φ the dimension of the feature space.

The higher N_Φ, the "bigger" the hypothesis space is. Therefore, the error given by underfitting the regression function will be lower (see section 2.1.4.1 and 2.1.5.4). If $N_\Phi \gg n$, remark 36 shows the advantage of computing the posterior predictive distribution using (2.37) instead of (2.36). In fact, the most interesting case, when the feature space is infinite-dimsional ($N_\Phi = \infty$), will be illustrated below.

2.2.3.1 The kernel trick in Gaussian Process regression

In (2.37), notice that the proyections to the feature space $\Phi(x),\ x \in \mathcal{X}$ and the weights prior covariance matrix Σ_w always appear in the form of a product,

$$\Phi(x)\Sigma_w\Phi(x'),\ x, x' \in \mathcal{X}. \tag{2.38}$$

Note that Σ_w is positive definite since it is the covariance matrix of the weights prior (2.34). Therefore, it admits the following singular value decomposition (SVD),

$$\Sigma_w = UDU^T,$$

with D diagonal and U orthogonal. See, for instance, Golub and Van Loan (2013), Demmel (1997) or Trefethen and Bau (1997) for more details about the SVD decomposition. Define $\Sigma_w^{1/2} = UD^{1/2}U^T$. Define the following feature map,

$$\Psi(x) = \Sigma_w^{1/2}\Phi(x), \ \forall x \in \mathcal{X}.$$

Thus, by the kernel trick (theorem 2.1.3) and the remark 24, there exists a kernel K such that,

$$K(x, x') = \langle \Psi(x), \Psi(x') \rangle = \Phi(x)\Sigma_w\Phi(x'). \tag{2.39}$$

for all $x, x' \in \mathcal{X}$.

For $\boldsymbol{t} = \{t_1, \ldots, t_s\}$ and $\boldsymbol{t'} = \{t'_1, \ldots, t'_r\}$ where $t_1, \ldots, t_s, t'_1, \ldots, t'_r \in \mathcal{X}$, define,

$$K[\boldsymbol{t}, \boldsymbol{t'}] = \Big(K(t_i, t'_j)\Big)_{\substack{1\leq i\leq s\\ 1\leq j\leq r}} \in \mathbb{R}^{s\times r}. \tag{2.40}$$

Following the notation in (2.40) and (2.15),

$$\begin{array}{rclcll} K[\boldsymbol{x}] & = & \big(K(x_i, x_j)\big)_{ij} & = & \Phi_{\mathbf{X}}^T\Sigma_w\Phi_{\mathbf{X}} & \in \mathbb{R}^{n\times n}, \\ K[\boldsymbol{x}, x_*] & = & (K(x_1, x_*), \ldots, K(x_n, x_*))^T & = & \Phi_{\mathbf{X}}^T\Sigma_w\Phi(x_*) & \in \mathbb{R}^{n\times 1}, \\ K[x_*] & = & K(x_*, x_*) & = & \Phi(x_*)^T\Sigma_w\Phi(x_*) & \in \mathbb{R}, \end{array} \tag{2.41}$$

where $x_i,\ 1 \leq i \leq n$, are the inputs of the training set and x_* is a test point.

Remark 37. If $N_\Phi \gg n$, notice that computing the matrices involving the feature map Φ in (2.37) becomes far less expensive numerically using the kernel trick and the equalities in (2.41). For example, the computational cost of the product $\Phi_{\mathbf{X}}^T\Sigma_w\Phi_{\mathbf{X}}$ is $\mathcal{O}(N_\Phi^2 n)$ operations. In contrast, using the kernel trick, $K[\boldsymbol{x}]$ is computed with only n^2 evaluations of the kernel K.

Eventually, the posterior predictive distribution (2.37) can be expressed as follows,

$$\begin{aligned} y|x_*, \boldsymbol{y}, \boldsymbol{x} \sim \mathcal{N}\Bigg(& K[\boldsymbol{x}, x_*]^T \left(K[\boldsymbol{x}] + \sigma^2 I_{n\times n}\right)^{-1} \boldsymbol{y}, \\ & K[x_*] - K[\boldsymbol{x}, x_*]^T \left(K[\boldsymbol{x}] + \sigma^2 I_{n\times n}\right)^{-1} K[\boldsymbol{x}, x_*]\Bigg). \end{aligned} \tag{2.42}$$

Remark 38. Note that the posterior predictive distribution given by a GP (2.42) is not only giving a guess of the outcome of interest provided by its mean, but it is also giving the uncertainties around this guess which are provided by its covariance matrix.

Remark 39. Making predictions according to (2.42) involves computing the kernel evaluations in (2.41) and the inversion of the $n \times n$ matrix $\left(K[\boldsymbol{x}] + \sigma^2 I_{n\times n}\right)$. Thus, the computational complexity is $\mathcal{O}(n^3)$ given by the inversion. Reducing this complexity would be interesting to allow GP to work with large datasets. See, e.g., Hensman, Fusi, and Lawrence (2013); Liu et al. (2020) or Gal, van der Wilk, and Rasmussen (2014).

Remark 40. The kernel trick in (2.39) can be applied even when the feature space $\Phi(\mathcal{X})$ is infinite-dimsional (see the theory of the previous section 2.1). Indeed, using non degenerate kernels (see definition 2.1.24) will lead to infinite-dimsional feature spaces which would be imposible to use without the kernel trick.

Remark 41. If

$$\alpha = (\alpha_1, \ldots, \alpha_n)^T = \left(K[\boldsymbol{x}] + \sigma^2 I_{n\times n}\right)^{-1} \boldsymbol{y},$$

then the mean of the posterior predictive distribution (2.42) is

$$K[\boldsymbol{x}, x_*]^T \alpha = \sum_{i=1}^{n} \alpha_i K_{x_i}(x_*),$$

which agrees with the Representer theorem (see theorem 2.1.6 and remark 27).

Remark 42. Notice that the feature map Φ disappears in (2.42). By Mercer theorem (theorem 2.1.2), any Mercer kernel is associated to a feature space. Therefore, given a Mercer kernel K, it is possible to compute the posterior predictive distribution (2.42) without knowing the associated feature map Φ. This fact makes kernels the main object of study and the corresponding feature spaces are hidden behind.

Remark 43 (Kernel ridge regression). Apply ridge regression (remark 34) to the projections to a feature space,

$$\begin{aligned} w_{\mathcal{E}_{\mathbf{z},\gamma}} &= \underset{w}{\operatorname{argmin}} \, \|\boldsymbol{y} - \Phi_{\mathbf{X}}^T w\|^2 + \gamma \|w\|^2 \\ &= \left(\Phi_{\mathbf{X}}\Phi_{\mathbf{X}}^T + \gamma I_{n\times n}\right)^{-1} \Phi_{\mathbf{X}}\boldsymbol{y}. \end{aligned}$$

Note that,

$$\left(\Phi_{\mathbf{X}}\Phi_{\mathbf{X}}^T + \gamma I_{n\times n}\right)\Phi_{\mathbf{X}} = \Phi_{\mathbf{X}}\left(\Phi_{\mathbf{X}}^T\Phi_{\mathbf{X}} + \gamma I_{n\times n}\right),$$

and multiply $\left(\Phi_{\mathbf{X}}\Phi_{\mathbf{X}}^T + \gamma I_{n\times n}\right)^{-1}$ at the left and $\left(\Phi_{\mathbf{X}}^T\Phi_{\mathbf{X}} + \gamma I_{n\times n}\right)^{-1}$ at the right to both sides of the equality. Then,

$$\Phi_{\mathbf{X}}\left(\Phi_{\mathbf{X}}^T\Phi_{\mathbf{X}} + \gamma I_{n\times n}\right)^{-1} = \left(\Phi_{\mathbf{X}}\Phi_{\mathbf{X}}^T + \gamma I_{n\times n}\right)^{-1}\Phi_{\mathbf{X}},$$

and,

$$w_{\mathcal{E}_{\mathbf{z},\gamma}} = \Phi_{\mathbf{X}}\left(\Phi_{\mathbf{X}}^T\Phi_{\mathbf{X}} + \gamma I_{n\times n}\right)^{-1}\boldsymbol{y}.$$

Therefore, for predicting the output y_* associated with the input x_*, it is necessary to compute,

$$\begin{aligned} y_* &= x_*^T w_{\mathcal{E}_{\mathbf{z},\gamma}} \\ &= x_*^T\Phi_{\mathbf{X}}\left(\Phi_{\mathbf{X}}^T\Phi_{\mathbf{X}} + \gamma I_{n\times n}\right)^{-1}\boldsymbol{y}. \end{aligned}$$

Applying the kernel trick in the same fashion as (2.41),

$$y_* = K[\boldsymbol{x}, x_*]^T \left(K[\boldsymbol{x}] + \gamma I_{n\times n}\right)^{-1}\boldsymbol{y}.$$

This regression technique is called kernel ridge regression. Note that, choosing the same kernel K and $\sigma^2 = \gamma$, it corresponds to the mean of the GP regression predictive distribution (2.42).

2.2.4 Gaussian processes

In last sections 2.2.2 and 2.2.3, Bayesian analysis in the linear model was used to derive the posterior predictive distribution (2.42) which is the tool given by Gaussian Process regression in order to make predictions. However, the same results can be derived from a different approach. In fact, the name “Gaussian Process” regression is used due to this approach. It will be explained in this section.

Definition 2.2.2 (Stochastic process)**.** *A stochastic process is a collection of random variables on a common probability space indexed by a set T.*

Definition 2.2.3 (Gaussian Process)**.** *A GP is a stochastic process where any finite number of its random variables have a joint Gaussian distribution.*

Consider a GP indexed by the input space $\mathcal{X}$ whose random variables $y(x),\ x \in \mathcal{X}$ take values on the output space $\mathcal{Y}$. Thus, for any point x, there is random variable $y(x)$ which represents the output y given the input x. To completely specify a GP it is necessary to give the jointly Gaussian distribution of any finite number of its random variables. It is sufficient to specify the expected value of each random variable, $\mathbb{E}[y(x)],\ x \in \mathcal{X}$, and the covariance between any pair of them, $\operatorname{Cov}(x, x'),\ x, x' \in \mathcal{X}$. For a finite set of $l \in \mathbb{N}$ random variables, $\{y(x_1, \ldots, y(x_l)\}$, the jointly distribution would be

$$\begin{pmatrix} y(x_1) \\ \vdots \\ y(x_l) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \mathbb{E}[y(x_1)] \\ \vdots \\ \mathbb{E}[y(x_l)] \end{pmatrix}, \begin{pmatrix} \operatorname{Cov}(x_1, x_1) & \ldots & \operatorname{Cov}(x_1, x_l) \\ \vdots & \ddots & \vdots \\ \operatorname{Cov}(x_l, x_1) & \ldots & \operatorname{Cov}(x_l, x_l) \end{pmatrix} \right). \tag{2.43}$$

Remark 44. Notice that if the GP index set is finite (in the case of regression, if the space of inputs $\mathcal{X}$ is finite), then the GP becomes simply a multivariate Gaussian distribution. From this perspective, a GP can be thought as a jointly distribution of an infinite set of random variables. Thus, it is possible (theoretically, without numerical limitations) to select as many points of the index set as it is wanted and get a sample of the associated random variables. If those samples are considered as evaluations of a function, the GP can be thought as a distribution over functions with domain equal to the index set.

A GP is illustrated in figure 2.5 showing the idea of a distribution over functions given in remark 44. The interval $\mathcal{X} = (0, 1) \subset \mathbb{R}$ was used as the index set. If $\boldsymbol{x} = \{x_1, \ldots, x_l\} \subset \mathcal{X}$ is a finite set, the following distribution,

$$\begin{pmatrix} y(x_1) \\ \vdots \\ y(x_l) \end{pmatrix} \sim \mathcal{N}\left(0\,,\ \left(\mathrm{Cov}(x_i, x_j)\right)_{\substack{1\le i\le l \\ 1\le j\le l}}\right), \tag{2.44}$$

with,

$$\mathrm{Cov}(x_i, x_j) = \exp\left(-\left(\frac{1}{2}\right)\frac{(x_i - x_j)^2}{0.3^2}\right), \tag{2.45}$$

was used as the joint Gaussian distribution. Notice that the expectations of each random variable are equal to zero and the covariance between two of them is given by a kernel[5]. Those choices were made because it is precisely what is done in GP regression (see remark 47). The details are explained below.

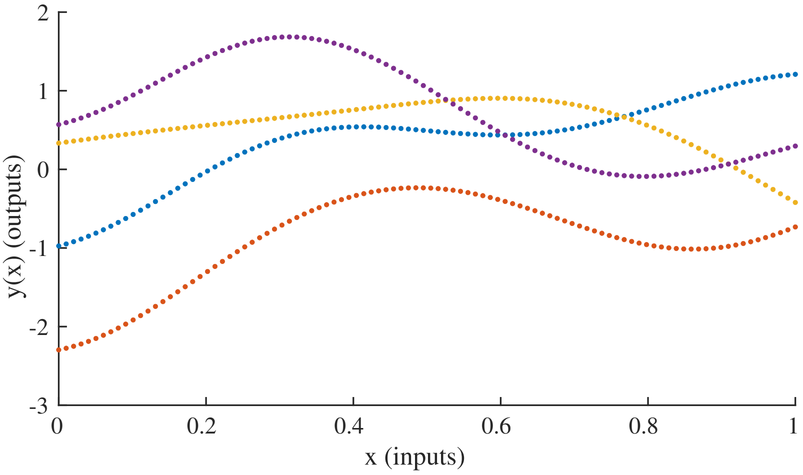


FIGURE 2.5: Four functons sampled from a GP distribution over functions. The interval $\mathcal{X} = (0, 1) \subset \mathbb{R}$ was used as a index set. 4 samples of 100 points were generated from the jointly Gaussian distribution specified in (2.44) and (2.45). Different colors were used for each sample.

Consider a training set,

$$\begin{aligned} \boldsymbol{z} &= \{(x_i, y_i)\}_{1\le i\le n}\,, \\ \boldsymbol{x} &= \{x_i\}_{1\le i\le n}\,, \\ \boldsymbol{y} &= (y_1, \ldots, y_n)^T\,, \end{aligned}$$

[5] In this particular case a squared exponential kernel with length-scale equal to 0.3. See (2.18).

and the linear model on a feature space Φ,

$$\begin{aligned} w &\sim \mathcal{N}(0, \Sigma_w), \\ \epsilon_{x_i} &\sim \mathcal{N}(0, \sigma^2), \ \ 1 \le i \le n, \\ y(x_i) &= \Phi(x_i)^T w + \epsilon_{x_i}, \ \ 1 \le i \le n, \\ y(x_*) &= \Phi(x_*)^T w, \ \ \forall x_* \in \mathcal{X} \setminus \boldsymbol{x}, \end{aligned}$$

where $\epsilon_{x_i}, \ \ 1 \le i \le n$, are i.i.d. and independent of w.

Remark 45. Notice that Gaussian noise ϵ is added only to the random variables associated with the inputs $\boldsymbol{x}$ of the training set $\boldsymbol{z}$. This is due to the uncertainties about the observations. In models where there are no uncertainties about the observations, the Gaussian noise can be removed. For example, the same results would be observed after running numerical computations with the same input.

Note that,

$$\begin{aligned} \mathbb{E}[w] &= 0, \\ \mathbb{E}[ww^T] &= \Sigma_w, \\ \mathbb{E}[\epsilon_{x_i}] &= 0, \ \ 1 \le i \le n, \\ \mathbb{E}[\epsilon_{x_i}^2] &= \mathrm{Var}(\epsilon_{x_i}) = \mathrm{Cov}(\epsilon_{x_i}, \epsilon_{x_i}) = \sigma^2, \ \ 1 \le i \le n, \\ \mathbb{E}[\epsilon_{x_i}\epsilon_{x_j}] &= \mathrm{Cov}(\epsilon_{x_i}, \epsilon_{x_j}) = 0, \ \ \text{for } i \ne j, \\ \mathbb{E}[\epsilon_{x_i} w] &= \mathbb{E}[\epsilon_x]\mathbb{E}[w] = 0, \ \ 1 \le i \le n, \end{aligned}$$

and therefore,

$$\begin{aligned} \mathbb{E}[y(x_i)] &= \Phi(x_i)^T \mathbb{E}[w] + \mathbb{E}[\epsilon_{x_i}] = 0, \ \ 1 \le i \le n, \\ \mathbb{E}[y(x_*)] &= \Phi(x_*)^T \mathbb{E}[w] = 0, \ \ \forall x_* \in \mathcal{X} \setminus \boldsymbol{x}, \\ \mathrm{Cov}(y(x_i), y(x_j)) &= \mathbb{E}[y(x_i)y(x_j)] \\ &= \mathbb{E}\left[(\Phi(x_i)^T w + \epsilon_{x_i})(\Phi(x_j)^T w + \epsilon_{x_j})\right] \\ &= \Phi(x_i)^T \mathbb{E}[ww^T]\Phi(x_j) + \mathrm{Cov}(\epsilon_{x_i}, \epsilon_{x_j}) \\ &= \begin{cases} \Phi(x_i)^T \Sigma_w \Phi(x_j) + \sigma^2, & \text{if } i = j \\ \Phi(x_i)^T \Sigma_w \Phi(x_j), & \text{if } i \ne j \end{cases}, \\ \mathrm{Cov}(y(x_i), y(x_*)) &= \Phi(x_i)^T \Sigma_w \Phi(x_*), \ \ \forall 1 \le i \le n, \ \forall x_* \in \mathcal{X} \setminus \boldsymbol{x}, \\ \mathrm{Cov}(y(x_*), y(x_*')) &= \Phi(x_*)^T \Sigma_w \Phi(x_*'), \ \ \forall x_*, x_*' \in \mathcal{X} \setminus \boldsymbol{x}. \end{aligned} \tag{2.46}$$

Applying the kernel trick to (2.46) in the same fashion as (2.39),

$$\begin{aligned}\operatorname{Cov}(y(x_i),y(x_j)) &= K(x_i,x_j)+\sigma^2, \quad \text{if } i=j,\\ \operatorname{Cov}(y(x_i),y(x_j)) &= K(x_i,x_j), \quad \text{if } i\neq j,\\ \operatorname{Cov}(y(x_i),y(x_*)) &= K(x_i,x_*), \ \forall 1\le i\le n,\ \forall x_*\in\mathcal{X}\setminus\boldsymbol{x},\\ \operatorname{Cov}(y(x_*),y(x_*')) &= K(x_*,x_*'), \ \forall x_*,x_*'\in\mathcal{X}\setminus\boldsymbol{x}.\end{aligned} \tag{2.47}$$

Remark 46. Note that the covariance between the outputs is given by the kernel K evaluated in the inputs (2.47). In GP, kernels are usually called covariance functions or covariance kernels due to this interesting feature.

Remark 47. Notice that using the joint Gaussian distribution in (2.43) with the expectations equal to zero as shown in (2.46), and the covariances given by a kernel as shown in (2.47), a GP indexed by $\mathcal{X}$ is completely specified. Remember from remark 44 that this is analogous to a distribution over functions with domain $\mathcal{X}$.

Let

$$\boldsymbol{x_*}=\{x_{*1},\dots,x_{*n_t}\},$$

be a finite number, n_t, of test points in $\mathcal{X}\setminus\boldsymbol{x}$. Let

$$\begin{aligned}\boldsymbol{y_x} &= (y(x_1),\dots,y(x_n))^T,\\ \boldsymbol{y_{x_*}} &= (y(x_{*1}),\dots,y(x_{*n_t}))^T,\end{aligned} \tag{2.48}$$

be random vectors. From (2.43) and (2.46), applying the kernel trick (2.47), and using the notation in (2.40), (2.15) and (2.48), the jointly Gaussian distribution of $\boldsymbol{y_x}$ and $\boldsymbol{y_{x_*}}$ is

$$\begin{pmatrix}\boldsymbol{y_x}\\ \boldsymbol{y_{x_*}}\end{pmatrix}\sim\mathcal{N}\left(0,\begin{pmatrix}K[\boldsymbol{x}]+\sigma^2 I_{n\times n} & K[\boldsymbol{x},\boldsymbol{x_*}]\\ K[\boldsymbol{x_*},\boldsymbol{x}] & K[\boldsymbol{x_*}]\end{pmatrix}\right). \tag{2.49}$$

Note that the outputs in the training set $\boldsymbol{y}$ have not been used yet. In order to make predictions, the idea is to derive the coditional distribution over $\boldsymbol{y_{x_*}}$ given $\boldsymbol{y_x}=\boldsymbol{y}$ from (2.49).

Proposition 2.2.2. *If Y_1 and Y_2 are random vectors with a joint Gaussian distribution equal to*

$$\begin{pmatrix}Y_1\\ Y_2\end{pmatrix}\sim\mathcal{N}\left(\begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix},\begin{pmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{pmatrix}\right),$$

with

$$\mathbb{E}[Y_1] = \mu_1 \in \mathbb{R}^r,$$
$$\mathbb{E}[Y_2] = \mu_2 \in \mathbb{R}^s,$$
$$\Sigma_{11} \in \mathbb{R}^{r\times r},\ \Sigma_{22} \in \mathbb{R}^{s\times s},\ \Sigma_{12} \in \mathbb{R}^{r\times s}, \Sigma_{21} = \Sigma_{12}^T,$$

then the conditional probability of Y_2 given $Y_1 = y$ is,

$$(Y_2 \mid Y_1 = y) \sim \mathcal{N}\left(\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(y - \mu_1)\,,\, \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}\right)$$

More details about proposition 2.2.2 and its proof can be found in Anderson (2003), Mises and Geiringer (1964) or Rasmussen and Williams (2006).

From (2.49) and applying proposition 2.2.2, the coditional distribution over $\boldsymbol{y_{x_*}}$ given the outputs of the training data $\boldsymbol{y_x} = \boldsymbol{y}$ is,

$$\begin{aligned}(\boldsymbol{y_{x_*}} \mid \boldsymbol{y_x} = \boldsymbol{y}) \sim \mathcal{N}\Bigg(& K[\boldsymbol{x_*}, \boldsymbol{x}]\left(K[\boldsymbol{x}] + \sigma^2 I_{n\times n}\right)^{-1}\boldsymbol{y}\,,\\ & K[\boldsymbol{x_*}] - K[\boldsymbol{x_*}, \boldsymbol{x}]\left(K[\boldsymbol{x}] + \sigma^2 I_{n\times n}\right)^{-1} K[\boldsymbol{x}, \boldsymbol{x_*}]\Bigg)\,.\end{aligned} \tag{2.50}$$

Remark 48. Note that the distribution (2.42) is equivalent to (2.50), i.e., the same result was got using Bayesian analysis in last section 2.2.3 and using the Gaussian Procces approach of this section. However, there is an important difference. Distribution (2.42) is only defined over one single test point. In other words, if more than one test point is wanted to be predicted, then it is necessary to give different posterior predictive distributions for each of them. Thus, no correlation between them is given. Distribution (2.50) is equal to (2.42) for a single test point. However, the former can be defined over multiple test points and it will give you a joint distribution taking into account correlations between them.

Notice that (2.50) gives a Gaussian distribution over any finite set of random variables $\{y(x_{*k})\}_k$. Therefore, it defines a Gaussian process indexed by $\mathcal{X} \setminus \boldsymbol{x}$. The GP given by (2.49) is called the Gaussian prior[6] over functions. The GP given by (2.50) is called the Gaussian posterior[7] over functions.

[6] Before using the outputs in the training data.

[7] After taking into account the outputs in the training data.

2.2.4.1 GP example

Given

- training data $\boldsymbol{z} = \{x_i, y_i\}_{1 \leq i \leq n}$,
- a Mercer kernel [8] K,
- the noise variance of the observations σ^2, and
- test points $\boldsymbol{x_*} = \{x_{*i}\}_{1 \leq i \leq n_t}$ to be predicted,

GP regression is given by the distribution (2.42).

Figure 2.6 shows an example. Consider 5 equidistant points,

$$\boldsymbol{x} = \{x_1, \ldots, x_5\} \subset (0, 1),$$

and the training data $\{(x_1, y_1), \ldots, (x_5, y_5)\}$ shown as black points in the graphs. Let K be an squared exponential kernel with length-scale $l = 0.15$ (see 2.18). Let $\sigma = 0.05$. Finally, consider 70 test points $\boldsymbol{x_*} = \{x_* i\}_{1 \leq i \leq 70}$. The figure shows 3 graphs. The first one are 3 samples from the Gaussian prior (2.49). No noise is added in the GP prior. The samples are plotted in different colors. The second one are 3 samples from the Gaussian posterior (2.50) plotted in different colors. The last one are 100 samples from the Gaussian posterior (2.50). In this case, all samples are plotted in red.

Figure 2.7 shows another example. However, in this case, a two-dimensional space $\mathcal{X} = (0, 1)^2$ was considered as input space. The GP was built from:

- *Training data:* a regression function

  $$f_\rho(x) = f_\rho(x^1, x^2) = \sin(2\pi x^1) + \cos(2\pi x^2), x = (x^1, x^2), \tag{2.51}$$

  was used for generating the training data adding Gaussian noise:

  $$\{x_i, y_i\}_{1 \leq i \leq 70}\,,\ y_i = f_\rho(x_i) + \mathcal{N}(0, 0.09). \tag{2.52}$$

  The training input points $\boldsymbol{x} = \{x_i\}_i$ were selected to compose a homogeneous grid covering the input space $\mathcal{X}$.

[8] A Kernel which is symmetric and positive definite. See definition 2.1.21.

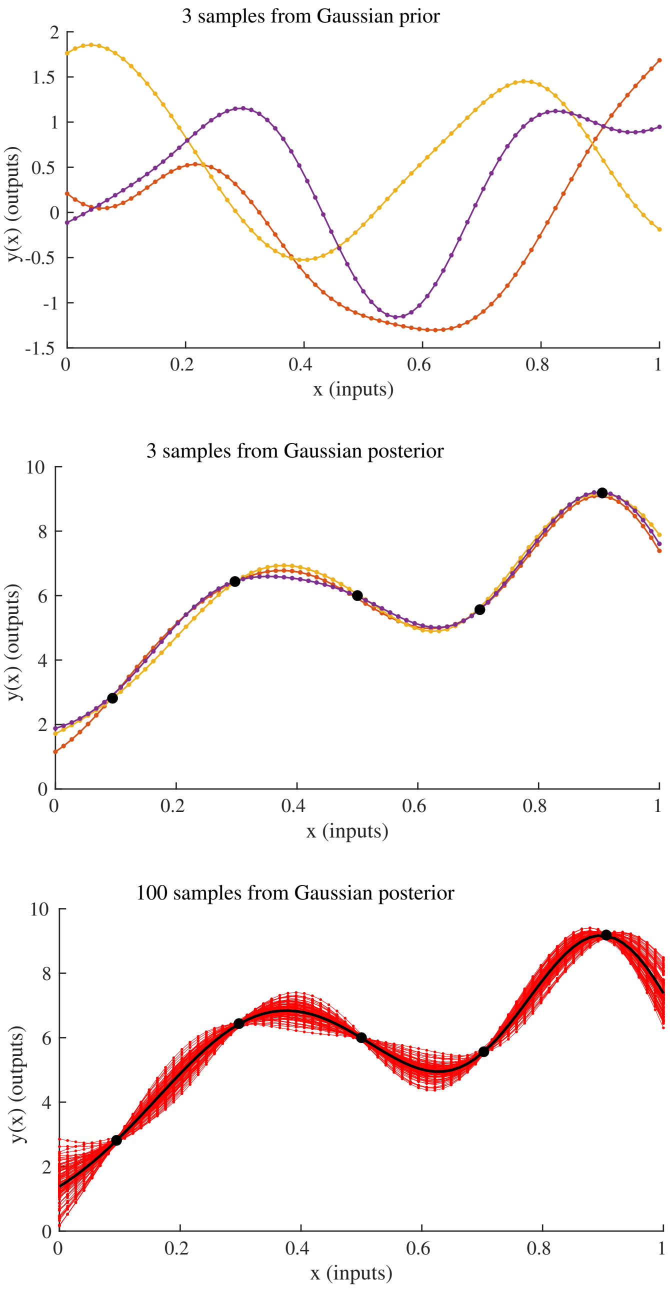


FIGURE 2.6: Example of GP prior and posterior. A squared exponential kernel was used with length-scale $l = 0.15$ (2.18), and a parameter $\sigma = 0.05$ (see 2.50). 5 points were used as a training data $\{(x_i, y_i)\}_{1 \leq i \leq 5}$ and 70 points $\{x_{*i}\}_{1 \leq i \leq 70}$ as test points. *Black points*: the 5 training points. *First graph*: 3 samples from the Gaussian prior (2.49). No noise is added in the GP prior. The samples are plotted in different colors. *Second graph*: 3 samples from the Gaussian posterior (2.50). The samples are plotted in different colors. *Last graph*: 100 samples from the Gaussian posterior (2.50). All samples are plotted in red. *Small points*: 70 points $\{(x_*, y_*)\}_{1 \leq i \leq 70}$ from each sample. *Black line*: Gaussian posterior mean. It is shown in last graph.

- *Kernel:* a squared exponential kernel was used with length-scale $l = 0.3$ (2.18).
- *Observations noise:* the variance of the observations noise used to build the GP was equal to the one used for generating the data, i.e., $\sigma^2 = 0.09$.
- *Test points:* finally, a grid of 2500 points $\boldsymbol{x_*} = \{x_{*i}\}_i$ was used as test points for the GP.

The figure is composed by 8 graphs in 3 rows. The first row are 3 samples from the Gaussian prior (2.49). No noise is added in the GP prior. The second row are 3 samples from the Gaussian posterior (2.50). The first graph of the third row is the mean of the Gaussian posterior. The last graph shows the regression function f_ρ and the training points. Although the training data was generated with a significant amount of noise[9], notice the good fitting of the GP posterior mean with respect to the regression function.

Figure 2.8 illustrates four Gaussian posteriors generated in the same conditions as figure 2.6. However, the variance of the observations noise σ^2 varies between the four GPs. Notice that all samples from the GP with $\sigma = 0$ pass through the training points. This is because the variance of the GP tends to 0 when x_* tends to a point in the training set. Yet, incresing σ allows the samples to be more flexible close to the training points. As expected, the far a point x_* is from the training set the more the samples vary from each other. Again, this is due to a greater variance of the GP posterior at these points.

Using different kernels K has also a significant impact in the Gaussian posterior. The succeed of GP regression is heavily dependent on the kernel choice. Most of the GP literature focus on this matter. Next section 2.2.5 will give a briefly summary of the most widely used kernels and the standard techniques to choose the most appropriate one.

2.2.5 Some kernel functions

Remember from last section 2.1 that the theory of RKHS is based on theorem 2.1.2, and therefore only Mercer kernels are considered.

[9] The standard deviation of the noise is $\sigma = 0.3$ and, $\max_{(x_1,x_2)\in\mathcal{X}} (f_\rho(x_1, x_2)) = 2$, $\min_{(x_1,x_2)\in\mathcal{X}} (f_\rho(x_1, x_2)) = -2$.

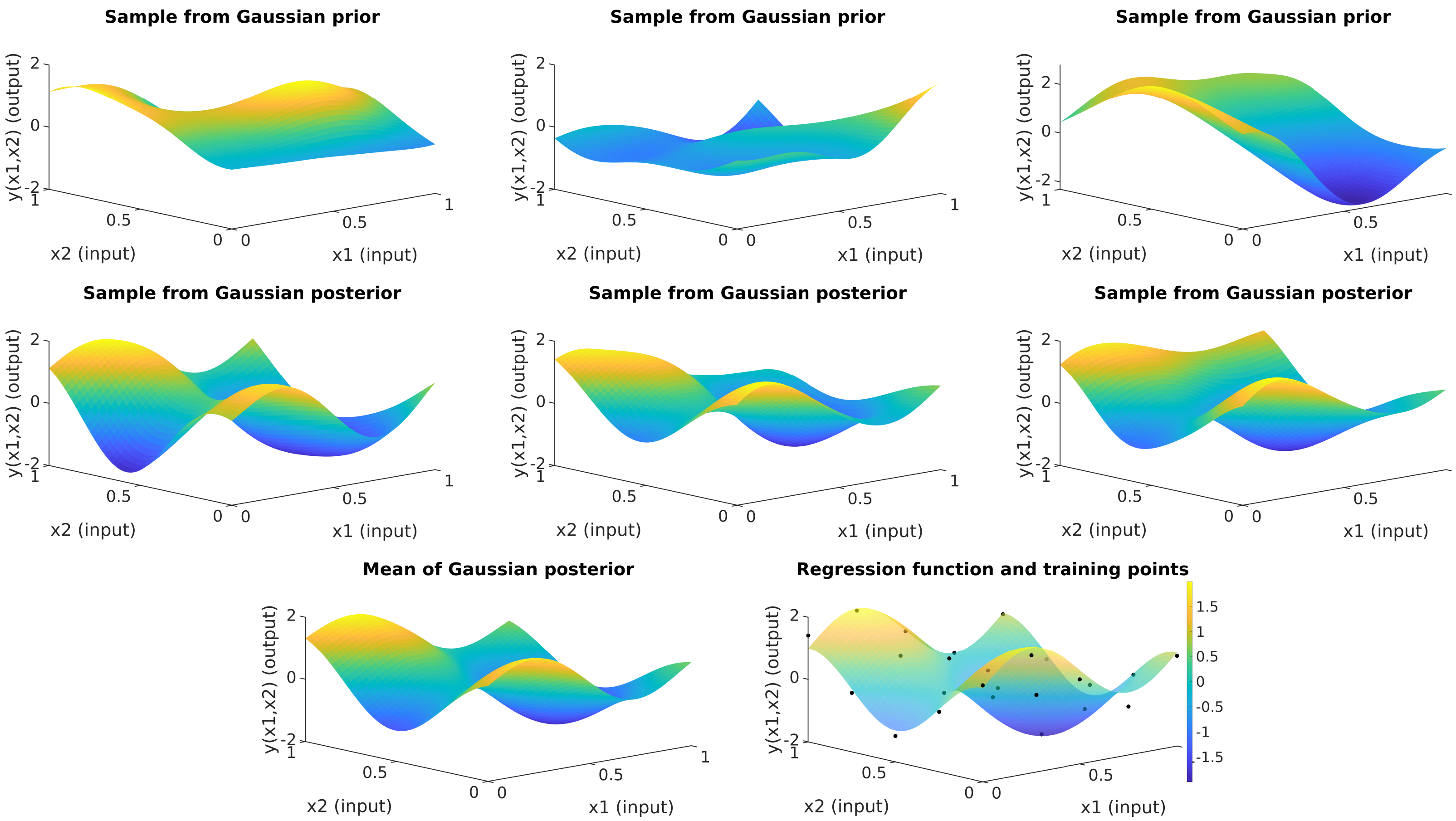


FIGURE 2.7: Example of GP regression in a two dimensional input space $\mathcal{X} = (0,1)^2$. The training data was generated from the evaluations of a regression function (2.51) plus Gaussian noise (2.52). A squared exponenetial kernel with length-scale $l = 0.3$ was used. The variance of the observations noise in the GP was equal to the variance of the noise added in the training data, i.e., $\sigma^2 = 0.09$. The graphs in the first row are 3 samples from the Gaussian prior (2.49). No noise is added in the GP prior. The graphs in the second row are 3 samples from the GP posterior (2.50). The first graph of the third row is the mean of the GP posterior. The regression function f_ρ and the training points are shown in the last graph. The surfaces were generated by linear interpolation.

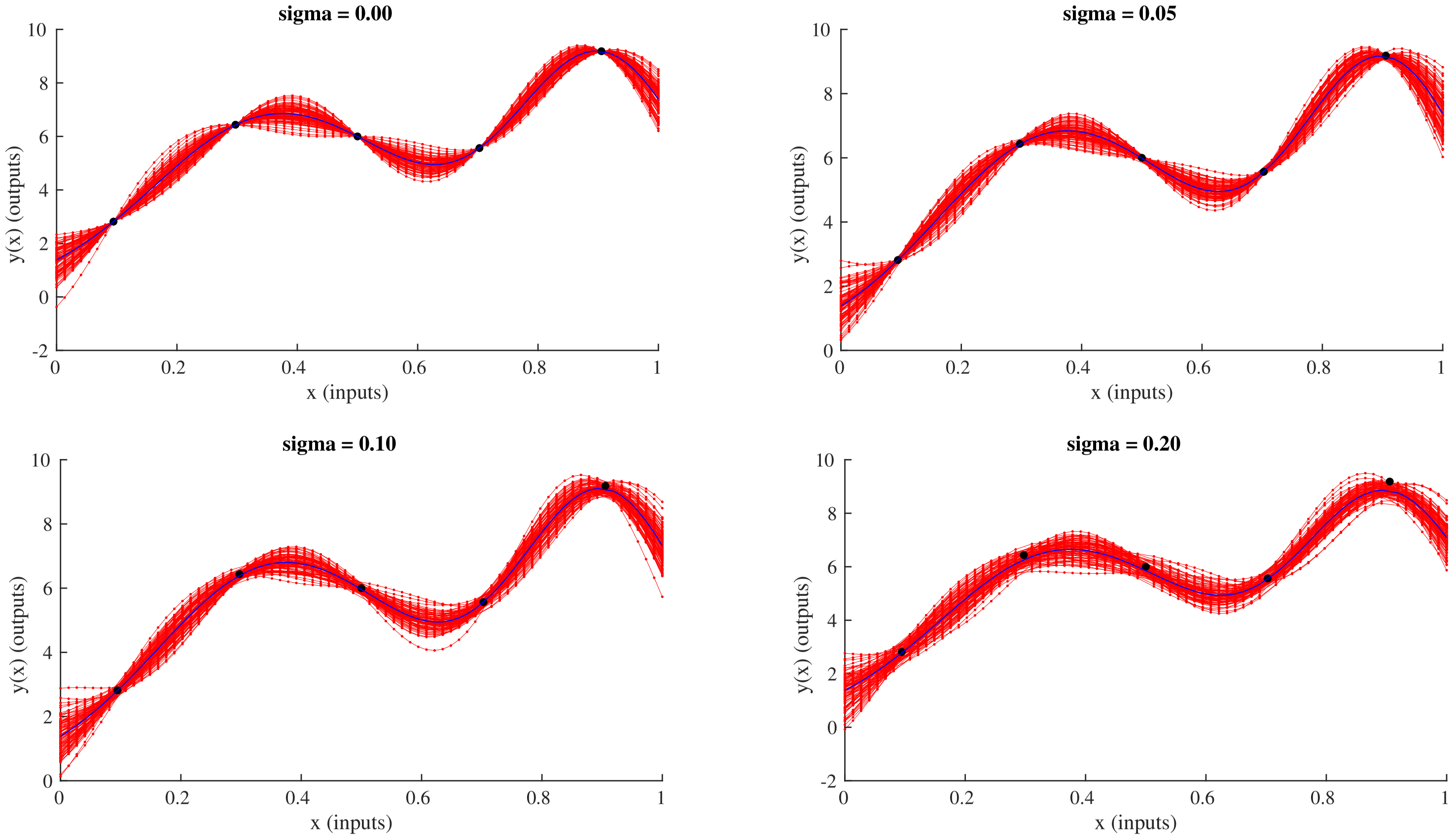


FIGURE 2.8: Examples of GP posteriors varying parameter σ (sigma). 4 Gaussian posteriors were generated with the following conditions. A squared exponential kernel was used with length-scale $l = 0.15$ (2.18). 5 points were used as a training data $\{(x_i, y_i)\}_{1\leq i\leq 5}$ and 70 points $\{x_{*i}\}_{1\leq i\leq 70}$ as test points. The parameter σ (see 2.50) varies according to each graph title. *Black points*: the 5 training points. *Blue lines*: Gaussian posterior mean. *red lines*: Each red line represents a sample from the Gaussian posterior. There were plotted 100 samples from each of the 4 Gaussian posteriors. *red points*: 70 points $\{(x_*, y_*)\}_{1\leq i\leq 70}$ from each sample.

Note that this is consistent with the role of kernels in GP regression since the covariance of a multivariate normal distribution has to be a symmetric and positive-definite matrix.[10]

Therefore, only Mercer kernels are appropriate to build a Gaussian Process.

Definition 2.2.4 (Stationary kernel)**.** *A kernel $K(x, x')$ which is a function of $x - x'$ is called stationary kernel.*

Remark 49. If $\tau = x - x'$, then a stationary kernel K is sometimes used as a function of τ, $K(\tau)$, instead of as a function of $x \in \mathcal{X}$ and $x' \in \mathcal{X}$, $K(x, x')$.

Remark 50. Note that a stationary kernel is invariant to translations in $\mathcal{X}$.

Remember from definition 2.1.24 and remark 19 that positive-semidefinite kernels (which are not positive-definite) can be considered although they will lead to the degenerate case.

Rasmussen and Williams (2006) gives a summary of commonly-used kernels. See table 2.1

2.2.6 Kernel selection

Table 2.1 shows that there are many families of kernel functions. This table only shows the most commonly used. In addition, different kernel functions are obtained varying the parameters of each parametric family.This section's goal is to give a method that optimally selects the kernel according to the data in the training set.

Given a kernel K, notice that,

$$\boldsymbol{y_x} \sim \mathcal{N}(0, K[\boldsymbol{x}] + \sigma^2 I_{n\times n}), \quad \text{(see (2.49))}$$

and therefore,

$$p(\boldsymbol{y_x}) = (2\pi)^{-\frac{n}{2}} \det(K[\boldsymbol{x}] + \sigma^2 I_{n\times n})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{y_x}^T (K[\boldsymbol{x}] + \sigma^2 I_{n\times n})^{-1} \boldsymbol{y_x}\right). \tag{2.53}$$

[10] See (2.49) and remark 13. In addition, notice that the sum of a positive-definite matrix and a diagonal matrix with positive entries is also positive-definite.

Kernel name		**Expression**	**S**	**D**
Constant		σ_0^2	●	●
Linear		$\sum_{i=1}^{N} \sigma_i^2 x_i x_i'$, where $x = (x_1, \dots, x_n)^T$		●
Polynomial		$(x^T x' + \sigma_0^2)^p$		●
Exponential	γ-exponential	$\exp\left(-\left(\frac{\|x-x'\|}{l}\right)^{\gamma}\right)$	●	
	Squared Exponential $\gamma = 2$	$\exp\left(-\frac{\|x-x'\|^2}{2l^2}\right)$		
	Exponential $\gamma = 1$	$\exp\left(-\frac{\|x-x'\|}{l}\right)$		
Matérn	Matérn	$\frac{1}{2^{\nu-1}\Gamma(\nu)}\left(\frac{\sqrt{2\nu}}{l}r\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}}{l}r\right)$, where Γ is the Gamma function, K_ν is a modified Bessel function of the second kind and $r = \|x - x'\|$.	●	
	Matérn $\nu = \frac{3}{2}$	$\left(1 + \frac{\sqrt{3}r}{l}\right) \exp\left(-\frac{\sqrt{3}r}{l}\right)$		
	Matérn $\nu = \frac{5}{2}$	$\left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right) \exp\left(-\frac{\sqrt{5}r}{l}\right)$		
Rational quadratic		$\left(1 + \frac{\|x-x'\|^2}{2\alpha l^2}\right)^{\alpha}$	●	
Neural Network		$\sin^{-1}\left(\frac{2\bar{x}^T\Sigma\bar{x}'}{\sqrt{(1+2\bar{x}^T\Sigma\bar{x})(1+2\bar{x}'^T\Sigma\bar{x}')}}\right)$, where $x = (x_1, \dots, x_N)^T$, $barx = (1, x_1, \dots, x_N)^T$		

TABLE 2.1: Commonly-used Kernels

Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be the inputs and the outputs of the training set. The aim is to find the kernel K that maximizes (2.53) when $\boldsymbol{y_x} = \boldsymbol{y}$.

Applying the logarithm,

$$\begin{aligned} \log(p(\boldsymbol{y})) = & -\frac{1}{2}\boldsymbol{y}^T(K[\boldsymbol{x}] + \sigma^2 I_{n\times n})^{-1}\boldsymbol{y} && \text{(data-fit penalty)} \\ & -\frac{1}{2}\log(\det(K[\boldsymbol{x}] + \sigma^2 I_{n\times n})) && \text{(complexity penalty)} \\ & -\frac{n}{2}\log(2\pi) \end{aligned} \tag{2.54}$$

Remark 51. Expression 2.53 and 2.54 are called "marginal likelihood" and "log marginal likelihood" in the GP literature. The terminology marginal is used because $p(\boldsymbol{y_x})$ can be seen as the marginalization of $p(\boldsymbol{y_x}|\boldsymbol{\hat{y}_x})$ w.r.t. $\boldsymbol{\hat{y}_x}$ with

$$\begin{aligned} \boldsymbol{\hat{y}_x} &\sim \mathcal{N}(0, K[\boldsymbol{x}]), \\ \boldsymbol{y_x} &\sim \mathcal{N}(\boldsymbol{\hat{y}}, \sigma^2 I_{n\times n}). \end{aligned}$$

Remark 52. Notice that $-\frac{n}{2}\log(2\pi)$ plays no role in the optimization process.

Remark 53. Notice the connections between section 2.1.4 and the data-fit and the complexity penalty terms.

Let $\{K_\theta\}_{\theta\in\Theta}$ be a family of kernel functions with parameters $\theta = (\theta_1, \dots, \theta_k)$.Let

$$K_{\boldsymbol{x}}(\theta) = K_\theta[\boldsymbol{x}] + \sigma^2 I_{n\times n}.$$

Consider the problem of optimizing the kernel in this family. The problem reduces to find,

$$\theta_{\text{opt}} = \underset{\theta\in\Theta}{\operatorname{argmax}}\ F_{\boldsymbol{x},\boldsymbol{y}}(\theta),$$

where

$$F_{\boldsymbol{x},\boldsymbol{y}}(\theta) = \underbrace{-\frac{1}{2}\boldsymbol{y}^T K_{\boldsymbol{x}}(\theta)^{-1}\boldsymbol{y}}_{\text{data-fit penalty}} \underbrace{-\frac{1}{2}\log(\det(K_{\boldsymbol{x}}(\theta)))}_{\text{complexity penalty}}.$$

The partial derivatives of $F_{\boldsymbol{x},\boldsymbol{y}}(\theta)$ are

$$\begin{aligned}\frac{\partial F_{\boldsymbol{x},\boldsymbol{y}}(\theta)}{\partial \theta_j} &= \frac{1}{2}\boldsymbol{y}^T K_{\boldsymbol{x}}(\theta)^{-1}\frac{\partial K_{\boldsymbol{x}}(\theta)}{\partial \theta_j}K_{\boldsymbol{x}}(\theta)^{-1}\boldsymbol{y} - \frac{1}{2}\text{tr}\left(K_{\boldsymbol{x}}(\theta)^{-1}\frac{\partial K_{\boldsymbol{x}}(\theta)}{\partial \theta_j}\right)\\ &= \frac{1}{2}\text{tr}\left((\alpha\alpha^T - K_{\boldsymbol{x}}(\theta)^{-1})\frac{\partial K_{\boldsymbol{x}}(\theta)}{\partial \theta_j},\right)\end{aligned}$$

where $\alpha = K_{\boldsymbol{x}}(\theta)^{-1}\boldsymbol{y}$.

The parameter optimization process can be done for different families and take the highest value. More details about this method and other methods (cross-validation techniques) to select the kernel and optimize its parameters can be found in Rasmussen and Williams (2006).

This chapter was dedicated to the problem of surrogate modeling. Different techniques can be used to build a surrogate model. This thesis adopted the regression point of view in ML. A summary of the mathematical concepts of learning theory was provided. GP regression was explained in detail. In addition, connections between GP and the learning theory concepts given in the first section of this chapter were emphasized (see, e.g., remarks 27, 28, 29, 35, 39, 41, 42, and 53).

It is assumed in the following chapter that a surrogate model is already given.

Chapter 3

Inverse problem: a novel framework for design optimization

An inverse problem is a problem where it is required to find the causal factors of some observed or desired outcomes of interest. Many problems in Engineering are inverse problems (see, e.g., Tanaka and Dulikravich (1998); Dulikravich and Tanaka (2000); Dulikravich and Tanaka (2001); or Tanaka (2003)). Therefore, the engineering has taken a growing interest in finding techniques to tackle them in the last decades (Neto and Neto (2012)).

Metaheuristics are a set of methods capable to deal with some of those inverse problems. Although they do not guarantee that an optimal solution is found, they may provide a sufficiently good solution (Blum and Roli (2001)). Yang (2010) explains metaheuristic applications in Engineering and most of its algorithms: Simulated Annealing (SA), genetic algorithms, ant algorithms, bee algorithms, particle swarm optimization, etc. In addition, this reference gives information about random number generators, Monte Carlo methods, Markov chains, and other sampling methods and concepts. This chapter provides a review of some of the aforementioned methods and concepts.

The main novelties of this thesis are given in sections 3.1.6 and 3.2. An inverse problem is stated. The aim is to find a probability distribution in the input space of a surrogate model that satisfies a prescribed performance in the output when uncertainties are propagated. A novel framework that tackles this problem and has been developed by the author of this thesis is introduced in section 3.2.

3.1 Sampling from a probability distribution

This section gives a review of different methods to sample from a PDF and explains how computers implement these methods.

3.1.1 Pseudorandom number generators

The problem of sampling given a Probability Density Function (PDF) is truly complex. Even sampling from a uniform distribution on $(0,1)$ is a complex problem. In fact, random numbers generated by computers are deterministic. They rely on sequences of numbers generated from a deterministic algorithm. Given a seed, the sequence is completely determined. However, their statistics approximate the statistics of true random numbers. Usually, random seeds are selected taking the value of a physical phenomenon that is expected to be random (although probably not uniformly distributed) such as atmospherical noise or CPU temperature.

These algorithms are called pseudorandom number generators. See, e.g., Knuth (1969) for more details. Algorithm 1 is a basic example of pseudo-random number generator. Notice that the sequence that it generates is periodic with the periodicity equal to m. Even the more complex pseudorandom number generators produce periodic sequences. Although it is named “random”, notice that given a seed x_0, the generated sequence $\{x_i\}_{i\geq 0}$ is deterministic.

Algorithm 1 does not give a very good approximation of a uniform distribution on the integers from 0 to $m-1$. However, it is the basic block for building more complex and suitable pseudorandom number generators (see, e.g., Krauth (2006)). The selection of m, n, and k and the seed x_0 drastically affects statistical properties such as mean and variance, and the period length.

```
set: $m$ (integer), $n$ (integer), $k$ (integer);
input: $x_i$ (integer);
$x_{i+1} = \mathrm{mod}(x_i \times n + k, m)$;
output: $x_{i+1}$ (integer)
```

Algorithm 1: Basic linear congruential pseudorandom number generator. A sequence is generated calling the algorithm several times. Each time the algorithm is called the input will be the output of the previous call. A seed x_0 is required.

3.1.2 Uniform distribution on $(0, 1)$

Given a sequence generated by algorithm 1 or any other pseudorandom number generator, it is possible to scale the numbers of the sequence x_i by $\frac{1}{m}$ to be in the interval $(0, 1)$, i.e., given the sequence $\{x_i\}_{i\geq 0}$, it is derived the sequence $\{\frac{x_i}{m}\}_{i\geq 0}$ of numbers belonging to the interval $[0, 1)$.

The goal of this scaling is to generate random numbers according to a uniform distribution in $(0, 1)$. Notice that the proposed scaling does not generate samples of a continuous distribution since the set of possible values is finite: $\{0, \frac{1}{m}, \frac{2}{m}, \dots, \frac{m-1}{m}\}$. However, if the pseudorandom number generator approximates a discrete uniform distribution on the integers $\{0, 1, \dots, m-1\}$, then the scaling approximates a continuous uniform distribution when $m \to \infty$. This discretization should not be a major issue when running algorithms in computers. Actually, float numbers[1] are a finite set, and therefore any probability distribution on them is discrete.

Remark 54. There is a chance of getting 0 from the method of generating uniform distributed numbers in $(0, 1)$ explained above. As a number belonging to $(0, 1)$ is required, the outputs equal to 0 should be discarded. It can be relevant to avoid overflow, e.g., if functions such as $\log(x)$ or $\frac{1}{x}$ are used.

3.1.3 Inverse transform sampling

Section 3.1.2 gives a method to generate samples from a uniform distribution on $(0, 1)$ when a random or pseudorandom number generator is available. The aim of this section is to give a method to generate samples from an arbitrary distribution when its Cumulative Distribution Function (CDF) and samples from uniform$(0, 1)$ are available.

Proposition 3.1.1. *Let $U \sim$ uniform$(0, 1)$. Let F be a CDF. Then the random variable $X = F^{-1}(U)$ has F as its CDF.*

Proof.

$$\mathbb{P}(X \leq x) = \mathbb{P}(F^{-1}(U) \leq x) = \mathbb{P}(U \leq F(x)) = F(x). \tag{3.1}$$

□

[1] The representation of real numbers in computers.

Remark 55. Proposition 3.1.1 supposes the existence of the inverse of F. CDFs are non-decreasing functions. However, they are not necessarily strictly increasing, and therefore the injectivity may fail. This issue can be tackle defining the inverse of F as,

$$F^{-1}(u) = \inf\{x \mid F(x) \geq u\}$$

Thus, for all $u \in (0,1)$ and $x \in F^{-1}((0,1))$,

$$F(F^{-1}(u)) \geq u,$$

and,

$$F^{-1}(F(x)) \leq x.$$

Therefore,

$$\{(u,x) \mid F^{-1}(u) \leq x\} = \{(u,x) \mid u \leq F(x)\}$$

and (3.1) also holds.

Algorithm 2 gives the method derived directly from proposition 3.1.1. More details can be found in, e.g., Robert and Casella (2004).

```
input: $F^{-1}$ (inverse of a CDF);
// Generate a random number on $(0,1)$
$u = \mathrm{ran}(0,1)$;
// Evaluate in the inverse of the CDF
$x = F^{-1}(u)$;
output: $x$;
```

Algorithm 2: Inverse Transform Sampling. Method to get samples from a random variable with F as its CDF.

Remark 56. The method has two weaknesses. First, it can only be used if the CDF exists. For some probability distributions, it is impossible to compute the CDF analytically[2]. Thus, this method may be computationally inefficient in the case of such distributions. Secondly, it requires the CDF inverse (or generalized inverse, see remark 55). Even with the existence proved, the same problem can arise again. It can be difficult or impossible to compute analytically.

[2] An important example is the normal distribution. This essential case will be treated in section 3.1.4.

3.1.4 Normal distribution

This section will give a method to sample from a normal distribution.

First, the CDF of a normal distribution $\mathcal{N}(\mu, \sigma^2)$ is

$$F(x) = \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{x-\mu}{\sigma\sqrt{2}}\right)\right),$$

where,

$$\operatorname{erf}(x) = \frac{1}{\sqrt{\pi}}\int_0^x e^{-t^2}dt.$$

Thus, as stated in remark 56, algorithm 2 is not suitable for sampling from a normal distribution.

However, recall the PDF of a standard normal distribution,

$$f(x) = \frac{1}{2\pi}e^{\frac{-x^2}{2}},\ x \in \mathbb{R},$$

and notice that,

$$\underbrace{\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}}dx}_{\mathcal{N}(0,1)}\underbrace{\int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}}e^{\frac{-y^2}{2}}dy}_{\mathcal{N}(0,1)} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\frac{1}{2\pi}e^{\frac{-(x^2+y^2)}{2}}dxdy$$

$$= \int_0^{2\pi}\frac{1}{2\pi}d\theta\int_0^{\infty}e^{\frac{-r^2}{2}}rdr \qquad \left(\text{variables change:}\begin{cases} x = \cos(\theta)r \\ y = \sin(\theta)r \\ dxdy = rd\theta dr \end{cases}\right)$$

$$= \underbrace{\int_0^{2\pi}\frac{1}{2\pi}d\theta}_{\text{uniform}(0,2\pi)}\ \underbrace{\int_0^{\infty}e^{-\bar{r}}d\bar{r}.}_{\text{exponential}(\lambda=1)} \qquad (\text{variable change: } \bar{r} = \frac{r^2}{2}, rdr = d\bar{r})$$

Therefore, if

$$\theta \sim \text{uniform}(0, 2\pi),$$
$$\bar{r} \sim \text{exponential}(\lambda = 1),$$

then,

$$x = \sqrt{2\bar{r}}\cos(\theta) \sim \mathcal{N}(0,1),$$
$$y = \sqrt{2\bar{r}}\sin(\theta) \sim \mathcal{N}(0,1).$$

Hence, two independent samples from a standard normal distribution can be generated from a sample of a uniform in $(0, 2\pi)$ and an exponential with parameter $\lambda = 1$.

Generating a sample from a uniform$(0, 2\pi)$ is straight forward,

$$2\pi \times \text{uniform}(0,1) \sim \text{uniform}(0, 2\pi).$$

For generating the sample from the exponential, it is possible to use algorithm 2. The CDF is

$$F(x) = \int_0^x e^{-\bar{r}} d\bar{r} = 1 - e^{-x},$$

and therefore,

$$F^{-1}(u) = -\log(1-u).$$

Remark 57. Notice that if $u \sim$ uniform$(0,1)$, then

$$(1-u) \sim \text{uniform}(0,1).$$

The formal proof of the construction above is as follows, if

$$\begin{aligned} u_1 &\sim \text{uniform}(0,1), \\ u_2 &\sim \text{uniform}(0,1), \end{aligned}$$

and considering the change of variables,

$$\left.\begin{aligned} x &= \sqrt{-2\log(u_2)}\cos(2\pi u_1) \\ y &= \sqrt{-2\log(u_2)}\sin(2\pi u_1) \end{aligned}\right\} \Rightarrow \begin{aligned} u_1 &= \tfrac{1}{2\pi}\arctan(\tfrac{y}{x}) \\ u_2 &= e^{-\frac{x^2+y^2}{2}} \end{aligned}$$

the joint PDF of x and y, f_{xy}, according to the change of variables rule is

$$\begin{aligned} f_{xy}(x,y) &= \underbrace{f_{u_1u_2}\left(u_1(x,y), u_2(x,y)\right)}_{1} \underbrace{\mid \mathrm{J}(x,y) \mid}_{\frac{u_1}{x}\frac{u_2}{y} - \frac{u_1}{y}\frac{u_2}{x}} \\ &= \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}} \\ &= \underbrace{\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}}_{\mathcal{N}(0,1)} \underbrace{\frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}}_{\mathcal{N}(0,1)}. \qquad (x \text{ and } y \text{ independent}) \end{aligned}$$

where $f_{u_1u_2}$ is the joint distribution of the uniformly distributed random variables u_1 and u_2. Note the independence of the random variables x and y.

Algorithm 3 gives the method in pseudocode.

```
$u_1 = \text{ran}(0,1)$;
$u_2 = \text{ran}(0,1)$;
$\theta = 2\pi u_1$;
$\bar{r} = -\log(u_2)$;
$x = \sqrt{2\bar{r}}\cos(\theta)$;
$y = \sqrt{2\bar{r}}\sin(\theta)$;
**output:** $x, y$;
```

Algorithm 3: Gaussian random numbers using Box-Muller transform. Method to get samples from a standard normal distribution.

This method is called Box-Muller transformation (Box and Muller, 1958). It can be improved to be marginally faster avoiding calls to the trigonometric functions. Notice that,

$$\left.\begin{array}{ll} r & = \text{uniform}(0,1) \\ \phi & = \text{uniform}(0,2\pi) \end{array}\right\} \Rightarrow (r\cos(\phi), r\sin(\phi)) \;\; \begin{array}{c}\text{random point within}\\ \text{the unit circle.}\end{array}$$

Thus, it is equivalent to

$$\begin{array}{c} u_1 = \text{uniform}(-1,1),\; u_2 = \text{uniform}(-1,1), \\ \left.\begin{array}{l} \bar{u}_1 = u_1 \\ \bar{u}_2 = u_2 \end{array}\right\} \text{rejecting } u_1^2 + u_2^2 \geq 1 \text{ and } u_1^2 + u_2^2 = 0. \end{array}$$

and therefore,

$$\left.\begin{array}{r} r\cos(\phi) = \bar{u}_1 \\ r\sin(\phi) = \bar{u}_2 \end{array}\right\} \Rightarrow \begin{array}{rl} \cos(\phi) = & \frac{\bar{u}_1}{\sqrt{\bar{u}_1^2+\bar{u}_2^2}} \\ \sin(\phi) = & \frac{\bar{u}_2}{\sqrt{\bar{u}_1^2+\bar{u}_2^2}} \end{array} .$$

In addition, notice that,

$$\bar{u}_1^2 + \bar{u}_2^2 \sim \text{uniform}(0,1) \Rightarrow \bar{r} = -\log(\bar{u}_1^2 + \bar{u}_2^2) \sim \text{exponential}(\lambda = 1).$$

Remark 58. Proving

$$\bar{u}_1^2 + \bar{u}_2^2 \sim \text{uniform}(0,1)$$

would require to go into the details of measure theory. Informally, if $(\bar{u}_1, \bar{u}_2)$ are points uniformly distributed within the unit circle and

$r = \sqrt{\bar{u}_1^2 + \bar{u}_2^2}$ then,

$$\begin{aligned} p(r) &= \text{normalization constant} \times \text{circumference of radius } r \\ &= \text{normalization constant} \times 2\pi r \\ &= \text{normalization constant} \times r. \end{aligned}$$

If $R = r^2$ and applying the change of variables rule,

$$f_R(R) = f_r(\sqrt{R})\frac{d(\sqrt{R})}{dR} = \text{constant} \times \sqrt{R}\frac{1}{2\sqrt{R}} = \text{constant},$$

where f_R and f_r are the PDFs of R and r respectively. Therefore,

$$R = r^2 = \bar{u}_1^2 + \bar{u}_2^2 \sim \text{uniform}(0, 1).$$

Finally,

$$\begin{aligned} x &= \sqrt{-2\log(\bar{u}_1^2 + \bar{u}_2^2)}\frac{\bar{u}_1}{\sqrt{\bar{u}_1^2+\bar{u}_2^2}} &= \sqrt{\frac{-2\log(\bar{u}_1^2+\bar{u}_2^2)}{\bar{u}_1^2+\bar{u}_2^2}}\ \bar{u}_1\ , \\ y &= \sqrt{-2\log(\bar{u}_1^2 + \bar{u}_2^2)}\frac{\bar{u}_1}{\sqrt{\bar{u}_1^2+\bar{u}_2^2}} &= \sqrt{\frac{-2\log(\bar{u}_1^2+\bar{u}_2^2)}{\bar{u}_1^2+\bar{u}_2^2}}\ \bar{u}_2\ . \end{aligned}$$

This improvement of the basic Box-Muller transformation (algorithm 3) is called Marsaglia polar method (Marsaglia and Bray, 1964). Algorithm 4 gives the pseudocode.

```
do
    $\bar{u}_1 = 2 \times \text{ran}(0, 1) - 1$;
    $\bar{u}_2 = 2 \times \text{ran}(0, 1) - 1$;
    $r = \bar{u}_1^2 + \bar{u}_2^2$;
while $r \geq 1$ or $r = 0$;
$x = \sqrt{\frac{-2\log(r)}{r}}\ \bar{u}_1$ ;
$y = \sqrt{\frac{-2\log(r)}{r}}\ \bar{u}_2$ ;
output: $x, y$;
```

Algorithm 4: Gaussian random numbers using Marsaglia polar method. Improvement of algorithm 3. Method to get samples from a standard normal distribution without calls to trigonometric functions.

Remark 59. Notice that algorithm 4 accepts,

$$\int_{\substack{\text{inside unit}\\\text{circle}}} du_1 du_2 = \frac{\pi}{4} \approx 78.54\%$$

and discards $1 - \frac{\pi}{4} = 21.46\%$ of the uniformly distributed random numbers generated due to the conditional loop, i.e., for every Gaussian sample generated by the algorithm, there are needed $\frac{4}{\pi} \approx 1.27$ uniformly distributed samples. This acceptance rate is sufficiently high to make algorithm 4 faster than algorithm 3 due to the avoidance of expensive trigonometric functions (see Bell, 1968). For the same reason, it is also more numerically robust.

3.1.4.1 Non-standard normal distribution

It has been shown how to generate random samples according to a standard normal distribution (see algorithms 3 and 4). Those algorithms implicitly give a method to sample from a normal distribution with an arbitrary mean μ and an arbitrary variance σ^2 with the simple linear transformation,

$$x \sim \mathcal{N}(0,1) \Longrightarrow \mu + \sigma x \sim \mathcal{N}(\mu, \sigma^2).$$

Proposition 3.1.2 gives the details.

Proposition 3.1.2. *If* $x \sim \mathcal{N}(0,1)$*, then* $y = \mu + \sigma x \sim \mathcal{N}(\mu, \sigma^2)$*.*

Proof. Let f_x and f_y be the PDFs of x and y respectively. Applying the change of variables rule,

$$f_y(y) = f_x\left(\frac{y-\mu}{\sigma}\right) \underbrace{\frac{dx}{dy}}_{\frac{1}{\sigma}} = \underbrace{\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y-\mu)^2}{2\sigma^2}}}_{\mathcal{N}(\mu,\sigma^2)}$$

□

3.1.4.2 Multivariate normal distribution

This section will explain the multivariate case, i.e., sampling from

$$x = (x_1, \dots, x_k) \sim \mathcal{N}(\mu, \Sigma),$$

where the mean μ is a k-dimensional vector and the covariance matrix Σ is a symmetric positive-definite $(k \times k)$-dimensional matrix.

Remark 60. Notice that if Σ is diagonal, then the random variables $\{x_1, \dots, x_k\}$ are independent. Therefore, samples from the random vector x can be generated from independent samples of univariate normal distributions. In particular, a sample from $x \sim \mathcal{N}(0, I_{k \times k})$ can be generated from k independent samples of a univariate standard normal distribution, $x_i \sim \mathcal{N}(0, 1)$, using, e.g., algorithm 3 or algorithm 4.

Proposition 3.1.3 (Cholesky factorization)**.** *If a matrix $A \in \mathbb{R}^{k \times k}$ is symmetric and is positive definite, then there exists a unique lower triangular matrix $L \in \mathbb{R}^{k \times k}$ with positive diagonal entries such that $A = LL^T$.*

Proposition 3.1.3 is a well-known result in the numerical linear algebra field. Proof and more details can be found in, e.g., Golub and Van Loan (2013), Trefethen and Bau (1997) or Demmel (1997). In addition to the proof of the existence of L, those references also give an algorithm to find it, i.e., to find L given A.

Proposition 3.1.4. *Excluding the degenerate case*[3]*, the covariance matrix of a multivariate normal distribution $\Sigma \in \mathbb{R}^{k \times k}$ is symmetric and positive-definite.*

Proof. The symmetry is trivial since,

$$\Sigma = \mathbb{E}\left[(x - \mathbb{E}[x])(x - \mathbb{E}[x])^T\right],$$

and $(x - \mathbb{E}[x])(x - \mathbb{E}[x])^T$ is symmetric. To proof the positive-definiteness, notice that,

$$\begin{aligned} v^T \mathbb{E}\left[(x - \mathbb{E}[x])(x - \mathbb{E}[x])^T\right] v &= \mathbb{E}\left[v^T (x - \mathbb{E}[x])(x - \mathbb{E}[x])^T v\right] \\ &= \mathbb{E}\left[\left((x - \mathbb{E}[x])^T v\right)^2\right] \geq 0, \end{aligned}$$

for all $v \in \mathbb{R}^k \setminus \{0\}$. Hence, it is positive-semidefinite. Since it is also nonsingular because the degenerate case was excluded, it is positive-definite. □

[3] When the covariance matrix is singular and the normal distribution has no density.

By proposition 3.1.4, Cholesky factorization (proposition 3.1.3) can be applied to a covariance matrix,

$$\Sigma = LL^T,$$

with L being lower triangular with positive diagonal entries.

Proposition 3.1.5. *Let $A \in \mathbb{R}^{k \times k}$ be a nonsingular matrix and $v \in \mathbb{R}^k$. Let $x \sim \mathcal{N}(\mu, \Sigma)$ with Σ nonsingular. The random variable $y = v + Ax$ is distributed according to a normal distribution with mean equal to $v + A\mu$ and covariance matrix $A\Sigma A^T$.*

Proof. Notice that $x = A^{-1}(y - v)$ and, since it is a linear transformation, $|\frac{\partial x}{\partial y}|$ is constant. Applying the change of variables rule, the PDF of y is

$$\begin{aligned} f_y(y) &\propto \exp\left(-\frac{1}{2}\left(\left(A^{-1}(y-v)-\mu\right)^T \Sigma^{-1}\left(A^{-1}(y-v)-\mu\right)\right)\right) \\ &\propto \exp\left(-\frac{1}{2}\left((y-(v+A\mu))^T\,(A^{-1})^T\Sigma^{-1}A^{-1}\,(y-(v+A\mu))\right)\right). \end{aligned}$$

Therefore, the statement holds. □

Corollary 3.1.1. *Let $\Sigma = LL^T$ be the Cholesky factorization (algorithm 3.1.3) of a symmetric and positive-definite matrix $\Sigma \in \mathbb{R}^{k \times k}$. Let $\mu \in \mathbb{R}^k$. Let $x \sim \mathcal{N}(0, I_{k \times k})$. The random vector $y = \mu + Lx$ is distributed according to a non-degenerate multivariate normal distribution with mean equal to μ and covariance matrix equal to Σ.*

Proof. The statement is a direct consequence of proposition 3.1.5. □

Corollary 3.1.1 gives a method to generate samples from any non-degenerate multivariate normal distribution using samples of a univariate standard normal distribution.

Given an arbitrary mean μ and an arbitrary non-singular covariance matrix Σ, samples $\{y_i\}_i$ from $\mathcal{N}(\mu, \Sigma)$ can be generated from samples $\{x_i\}_i$ of $\mathcal{N}(0, I_{k \times k})$ using the Cholesky factorization $\Sigma = LL^T$ given in proposition 3.1.3 and the linear transformation $y_i = \mu + Lx_i$ of corollary 3.1.1. Remember from remark 60 that samples from $\mathcal{N}(0, I_{k \times k})$ can be generated from independent samples of a univariate standard normal distribution $\mathcal{N}(0, 1)$. Finally, samples from $\mathcal{N}(0, 1)$ can be generated using, e.g., algorithm 3 or algorithm 4.

Algorithm 5 gives the pseudocode.

```
inputs: $\mu = (\mu_1, \ldots, \mu_k) \in \mathbb{R}^k,\ \Sigma = (\Sigma)_{ij} \in \mathbb{R}^{k \times k}$;
// Cholesky factorization
// $L = (L_{ij})_{ij} \in \mathbb{R}^{k \times k}$, with $L_{ij} = 0$ if $j > i$
$L = \text{Cholesky}(\Sigma)$;
for $i = 1 \ldots k$ do
    // sample from an standard normal distribution
    $x_i = \text{randn}()$;
    $y_i = \mu_i$;
    for $j = 1 \ldots i$ do
        $y_i = y_i + L_{ij} x_j$;
    end
end
output: $y = (y_1 \ldots y_k)$;
```

Algorithm 5: Algorithm to get samples from a multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$. Cholesky(Σ) returns a matrix L according to the Cholesky factorization $\Sigma = LL^T$ (3.1.3). randn() returns a random number generated from a univariate standard normal distribution (Algorithms 3 and 4 can be used for this purpose).

3.1.5 Sampling from an arbitrary PDF

Methods for sampling from a uniform distribution and from any univariate or multivariate normal distribution were given in sections 3.1.2 and 3.1.4. Algorithm 2 gives a method for sampling from other distributions. However, the drawback of that method is that the inverse of the CDF must be available.

This section presents methods to sample from an arbitrary PDF. In fact, only a function $f(x)$ proportional to the PDF is necessary.

3.1.5.1 Rejection sampling

Rejection sampling generates independent samples from an arbitrary distribution with PDF $f(x)$ using a proposal distribution with PDF $g(x)$ from which it is possible to sample.

The basic principle of this method is shown in theorem 3.1.1.

Theorem 3.1.1 (Fundamental theorem of simulation). *Sample from*

$$x \sim f(x),$$

is equivalent to sample from

$$(x,u) \sim \text{uniform}\left(\{(x,u) \mid 0 < u < f(x)\}\right), \tag{3.2}$$

and taking the values x.

Proof. It is sufficient to show that the marginal of x according to the law (3.2) is $f(x)$,

$$\int_0^{f(x)} du = f(x).$$

□

Theorem 3.1.1 and a broaden discussion about it can be found in Robert and Casella (2004). This theorem already gives a method to sample according to an arbitrary density given samples from a uniform distribution. Let Ω be the sample space of x and $M \geq \sup_x f(x)$. The method consists of taking uniformly distributed samples on

$$(x,u) \in \Omega \times (0,M),$$

and rejecting the ones such that $u \geq f(x)$. Figure 3.1 shows an example.

The main drawback of applying theorem 3.1.1 directly is that the rejection rate can be so high that makes the algorithm inefficient, especially if the density given by $f(x)$ is concentrated in a small region of Ω.[4]

A significant improvement consists of sampling on Ω from a probability density $g(x)$ that approximates $f(x)$ as much as possible and from which it is known how to obtain samples, instead of sampling from uniform(Ω). The principle is the same than theorem 3.1.1.

Let $u \in (0,1)$ and consider the following join density in the pairs (x,u),

$$p(x,u) = \left\{ \begin{array}{cc} Mg(x), & u < \frac{f(x)}{Mg(x)} \\ 0, & u \geq \frac{f(x)}{Mg(x)} \end{array} \right. , \tag{3.3}$$

[4] A problem that is accentuated in high dimensions.

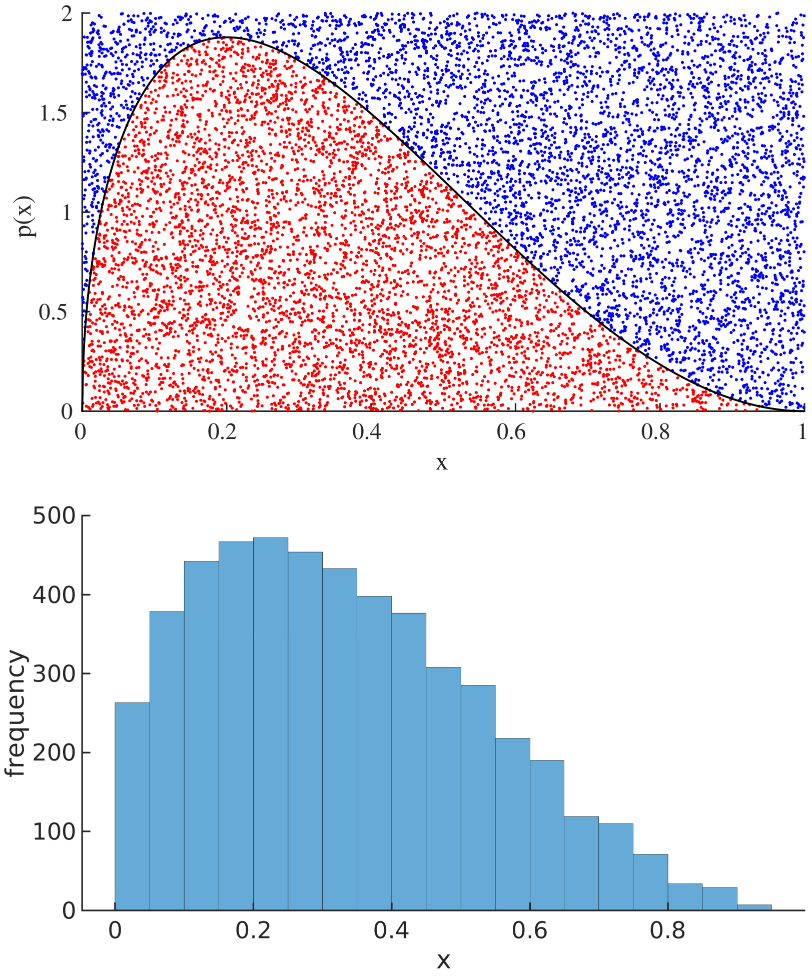


FIGURE 3.1: Example of rejection sampling using a uniform proposal distribution. The figure shows (empirically) that rejection sampling emulates the target density. *Black line*: Beta density $f(x)$ with parameters $\alpha = 1.5$ and $\beta = 3$. *Points*: Random points (x, u) uniformly distributed in $(0, 1) \times (0, 2)$. *Red points*: Accepted points (x, u) satisfying $u < f(x)$. *Blue points*: Rejected points (x, u) satisfying $u \geq f(x)$. *Second graph*: Histogram of the accepted points.

where $M \in \mathbb{R}$ is a constant such that $Mg(x) > f(x)$ for all x in the sample space Ω. The density (3.3) is well defined since,

$$\int_\Omega \int_0^1 p(x,u)dudx = \int_\Omega \int_0^{\frac{f(x)}{Mg(x)}} Mg(x)dudx = \int_\Omega f(x)dx = 1$$

Notice that the marginal coincides with $f(x)$,

$$p(x) = \int_0^{\frac{f(x)}{Mg(x)}} p(x,u)du = \int_0^{\frac{f(x)}{Mg(x)}} Mg(x)du = f(x). \tag{3.4}$$

Let $p_2(x,u) = p_2(x)p_2(u)$ be another joint density on the pairs (x,u) with x and u independent, $p_2(x) = g(x)$ and $p_2(u) = 1$. Thus, samples of (x,u) according to p_2 can be generated indepedently from samples,

$$\begin{aligned} &x \sim g(x), \\ &u \sim \text{uniform}(0,1). \end{aligned} \tag{3.5}$$

Notice that,

$$\begin{aligned} p_2\left((x,u) \mid (x,u) \in \left\{(x,u)|u < \frac{f(x)}{Mg(x)}\right\}\right) &= \\ &= \frac{p_2(x,u)}{p_2\left((x,u) \in \left\{(x,u)|u < \frac{f(x)}{Mg(x)}\right\}\right)} \\ &= \frac{g(x)}{\int_\Omega \int_0^{\frac{f(x)}{Mg(x)}} g(x)dudx} \\ &= Mg(x). \end{aligned}$$

Therefore, for sampling from 3.3, it is only necessary to sample independently according to 3.5 and discard the pairs $\left\{(x,y)|u \geq \frac{f(x)}{Mg(x)}\right\}$. Finally, as it is proved in 3.4, taking only the values x leads to the law defined by the density $f(x)$.

Refer to Robert and Casella (2004) for more information about rejection rampling. Algorithm 6 gives the pseudocode.

```
inputs: target density $f(x)$, proposal density $g(x)$ and
  constant $M$ such that $Mg(x) > f(x)$, $\forall x$;
do
    // get samples according to $g(x)$ and uniform$(0,1)$
    $x = \mathrm{ran}_{\sim g(x)}()$;
    $u = \mathrm{ran}(0,1)$;
while $u \geq \frac{f(x)}{Mg(x)}$;
output: $x$;
```

Algorithm 6: Rejection sampling. Algorithm to sample according to the target PDF $\propto f(x)$ from a proposal $g(x)$, assuming that samples according to $g(x)$ can be generated.

Remark 61. Notice that, actually, if $f(x)$ is not normalized the method still works.

Remark 62. It can be shown (see, e.g., Robert and Casella, 2004) that if,

$$\begin{aligned} x &\sim g(x), \\ u &\sim \text{uniform}(0,1), \end{aligned}$$

then,

$$\mathbb{P}\left(u < \frac{f(x)}{Mg(x)}\right) = \frac{1}{M}.$$

Therefore, the rejection sampling method is optimized by setting,

$$M = \sup_x \frac{f(x)}{g(x)}.$$

Remark 63. Improvements of rejection sampling method focus on finding suitable proposals $g(x)$ which lead to a low rejection rate.

A major drawback of rejection sampling is that it is difficult to find a proposal that leads to a good acceptance rate in high dimensions. The density of the target PDF is concentrated in a region of the sample space and many rejections are needed until a sample with high acceptance rate is generated.

Fortunately, there is a family of methods called Markov Chain Monte Carlo (MCMC) algorithms that can deal with densities in high-dimensions. However, in contrast to rejection sampling, samples generated by MCMC algorithms are correlated. This is a problem inherent in the MCMC sampling procedure.

3.1.5.2 Markov chain sampling

Given a function $f(x)$ proportional to a desired probability density, the aim is to develop an algorithm able to sample according to that density, even in a high dimensional scenario.

There is a family of algorithms based on Markov chains capable to tackle this problem. The theory of Markov chains is beyond the scope of this work. Nevertheless, a simple example in a discrete sample space will be given. It will be sufficient to understand the basic principle on which Markov chain sampling algorithms are based.

Refer to Krauth (2006) to broaden some of the ideas given below. Refer to Robert and Casella (2004) for a deep understanding of Markov chains.

Let x be a discrete variable which can take 9 values,

$$x \in \Omega = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}.$$

Let p_s be a probability distribution in the sample space Ω, e.g., $p_s(x) = \frac{1}{9}$ for all $x \in \Omega$. The distribution p_s will be called stationary distribution. Suppose that it is required to generate samples according to p_s. Obviously, in this example, it is possible to use a random number generator (see alrgorithm 1). This approach is called "direct sampling". However, remember that the goal is to sample according to densities from which there is not a direct method to generate samples. Instead of doing that, consider the following approach,

1. Select a value x_0 in the sample space Ω (randomly according to a known distribution or deterministically).

2. Assign transition probabilities $\{p(x \to x')\}_{x,x' \in \Omega}$ to move from a value $x \in \Omega$ to another value $x' \in \Omega$ (or staying in the same value).

3. Generate a chain $\{x_0, x_1, x_2, \dots\}$ starting at x_0 and moving according to the transition probabilities $p(x \to x')$[5].

Is there any selection of transition probabilities $\{p(x \to x')\}_{x,x' \in \Omega}$ such that, after several movements, the probability of being in a value x is the same (or very close) to the stationary probability $p_s(x)$ for all $x \in \Omega$?.

[5] The series $\{x_0, x_1, \dots\}$ is called Markov chain and this process of moving around the sample space Ω is called random walk.

The answer to this question, i.e., a suitable choice of transition probabilities, is precisely what allows Metropolis-Hasting (M-H) to sample according to any given probability density $f(x)$, even if $f(x)$ is not normalized[6] or the random variable x is continuous, moving around the sample space Ω. The details of M-H shall be explained in next section 3.1.5.3.

Consider that the values in the chain $\{x_1, x_2, \dots\}$ are random variables distributed according to the initial value x_0 and the random walk given by the transition probabilities $\{p(x \to x')\}_{x,x' \in \Omega}$.

If $\{p(x \to x')\}_{x,x' \in \Omega}$ are transition probabilities satisfying the stated question, then the probability densities $\{p_i = p(x_i)\}_{i>1}$ associated to the random variables $\{x_i\}_{i>1}$ tend to $p_s(x)$ when $i \to \infty$. Thus, intuitively, in the limit, the following condition must be necessary,

$$p_s(x) = \sum_{x'=1}^{9} p_s(x')p(x' \to x),$$

for all $x \in \Omega$. Therefore,

$$p_s(x)\,(1 - p(x \to x)) = \sum_{\substack{x' \in \Omega \\ x' \neq x}} p_x(x')p(x' \to x), \tag{3.6}$$

and, since,

$$1 - p(x \to x) = \sum_{\substack{x' \in \Omega \\ x' \neq x}} p(x \to x'),$$

equality (3.6) leads to

$$\sum_{\substack{x' \in \Omega \\ x' \neq x}} p_s(x)p(x \to x') = \sum_{\substack{x' \in \Omega \\ x' \neq x}} p_x(x')p(x' \to x). \tag{3.7}$$

Notice that a sufficient condition (but not necessary) for 3.7 is,

$$p_s(x)p(x \to x') = p_s(x')p(x' \to x), \tag{3.8}$$

for all pairs $x, x' \in \Omega$. Condition (3.8) is usually called "detailed balance" in the Markov chains literature. It is also called "reversibility", "microscopic reversibility" or "time reversibility" by some authors. See, e.g., Chib and Greenberg (1995). It is the key of designing

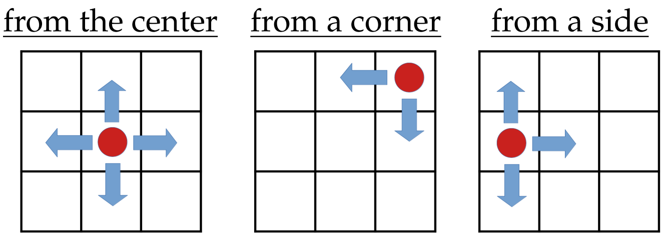
[6] $f(x)$ is proportional to a probability density which it is wanted to simulate, i.e., $f(x)$ is not a PDf but C $\times$ $f(x)$ is a PDF for a normalizing constant C.

sampling methods based on Markov chains (see M-H in next section 3.1.5.3) because it is a sufficient condition[7] for the Markov process to asymptotically reach a unique stationary distribution which, in addition, coincides with the desired distribution.

It will be constructed transition probabilities $\{p(x \to x')\}_{x,x' \in \Omega}$ for the example above. For a better visualization of the random walk, consider the values in Ω to be in a grid,

1	2	3
4	5	6
7	8	9

and transition probabilities allowing only the following movements,



from the center

from a corner

from a side

or staying in the same value.

For example, the transition probabilities from the corner 1 are,

$$\begin{aligned} &p(1 \to 1) \geq 0, \\ &p(1 \to 2) \geq 0, \\ &p(1 \to 4) \geq 0, \\ &p(1 \to x) = 0, \ \forall x \in \{3,5,6,7,8,9\}. \end{aligned}$$

Therefore, from the value $x_i = 1$, the process can stay at 1, move to 2 or move to 4, i.e., x_{i+1} can take the values 1, 2 or 4. In addition,

$$\sum_{x=1}^{9} p(1 \to x) = p(1 \to 1) + p(1 \to 2) + p(1 \to 4) = 1, \tag{3.9}$$

since $p(1 \to x), x \in \Omega$, is a probability distribution itself. Consider a desired distribution given by $p_s(x), x \in \Omega$, from which it is wanted to sample. Following the reasoning explained above, the procedure will be to move around the sample space Ω, i.e., around the grid,

[7] Assuming the ergodicity of the Markov process. See Robert and Casella (2004).

with the aim of approximating p_s after several moves. Therefore, in the limit, when the number of movements tends to ∞, the following condition must be satisfied,

$$p_s(1) = p_s(1)p(1 \to 1) + p_s(2)p(2 \to 1) + p_s(4)p(4 \to 1), \quad (3.10)$$

since 1 can only be reached from 2, 4 and 1 itself. Hence, from (3.9) and (3.10),

$$p_s(1)p(1 \to 2) + p_s(1)p(1 \to 4) = p_s(2)p(2 \to 1) + p_s(4)p(4 \to 1),$$

condition that can be satisfied imposing the detailed balance,

$$p_s(1)p(1 \to 2) = p_s(2)p(2 \to 1),$$

and,

$$p_s(1)p(1 \to 4) = p_s(4)p(4 \to 1).$$

Notice that the same construction can be done with the other values, $2, 3, \ldots, 9$.

Let $p_s(x) = \frac{1}{9}$, for all $x \in \Omega$, be the desired distribution. In this case, when p_s is uniform in Ω, the detailed balance condition reduces to

$$p(x \to x') = p(x' \to x),$$

for all $x, x' \in \Omega$.

If the transition probabilities are, e.g., $p(x \to x') = \frac{1}{4}$, $x \neq x'$, for the allowed movements,

$$
\begin{aligned}
p(1 \to 1) &= 1 - p(1 \to 2) - p(1 \to 4) = 1 - \frac{1}{4} - \frac{1}{4} = \frac{1}{2}, \\
p(2 \to 2) &= 1 - p(2 \to 1) - p(2 \to 5) - p(2 \to 3) = 1 - \frac{3}{4} = \frac{1}{4}, \\
p(3 \to 3) &= \cdots = \frac{1}{2}, \\
p(4 \to 4) &= \cdots = \frac{1}{4}, \\
p(5 \to 5) &= \cdots = 0, \\
p(6 \to 6) &= \cdots = \frac{1}{4}, \\
p(7 \to 7) &= \cdots = \frac{1}{2}, \\
p(8 \to 8) &= \cdots = \frac{1}{4}, \\
p(9 \to 9) &= \cdots = \frac{1}{2},
\end{aligned}
$$

and $p(x \to x') = 0$ for the other cases, then the detailed balance is satisfied.

In the discrete case, the transition probabilities can be shown in a matrix,

$$
K = (p(i \to j))_{ij} = \begin{pmatrix}
\frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 & 0 \\
\frac{1}{4} & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 & 0 & 0 \\
0 & \frac{1}{4} & \frac{1}{2} & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 \\
\frac{1}{4} & 0 & 0 & \frac{1}{4} & \frac{1}{4} & 0 & \frac{1}{4} & 0 & 0 \\
0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 & \frac{1}{4} & 0 \\
0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & 0 & 0 & \frac{1}{4} \\
0 & 0 & 0 & \frac{1}{4} & 0 & 0 & \frac{1}{2} & \frac{1}{4} & 0 \\
0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\
0 & 0 & 0 & 0 & 0 & \frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{2}
\end{pmatrix}. \tag{3.11}
$$

Notice that a distribution in Ω can be represented by a vector,

$$
v_p = (p(1), p(2), \ldots, p(9))^T.
$$

Following this notation,

$$v_{p_s} = \left(\frac{1}{9}, \frac{1}{9}, \ldots, \frac{1}{9}\right)^T.$$

Let $p_0 = p(x_0)$ be the distribution of the intial value x_0. Remember that the choice of x_0 can be deterministic, e.g., if it is wanted to impose $x_0 = 3$, then

$$v_{p_0} = (0, 0, 1, 0, 0, 0, 0, 0, 0)^T.$$

Define $p_i = p(x_i)$, i.e., the distribution of the random variable x_i (the value after i movements). Note that

$$v_{p_i} = K v_{p_{i-1}} = K^i v_{p_0}.$$

Eventually, in order to meet the requirement stated above and tending to the desired density,

$$K^i v_{p_0} \xrightarrow{i \to \infty} v_{p_s}, \tag{3.12}$$

should hold.

Remark 64. Indeed, condition (3.12) is true for any initial distribution p_0. Studying the spectrum of K, [8] it can be seen that the dominant eigenvalue is 1 and an eigenvector associated to it is

$$(1, 1, 1, 1, 1, 1, 1, 1)^T.$$

Remark 65. The second largest eigenvalue of K is 0.75. Thus, the error between p_s and the asymptotic approximation p_i is given by $\approx 0.75^i u_2$, where u_2 is an eigenvector of eigenvalue 0.75.

3.1.5.3 Metropolis Hastings

This section will explain M-H sampling algorithm. First, the idea of building a Markov chain $\{x_0, x_1, \ldots\}$ from a random walk around the sample space introduced in last section 3.1.5.2 will be extended to the continuous case. Secondly, a methodology to make transitions

[8] Refer to the power method (e.g. Golub and Van Loan, 2013) for more details about the connection between $K^i p_0$ and the eigenvalues and eigenvectors of K.

which satisfy the detailed balance condition (3.8) will be given. Finally, a pseudocode of the algorithm will be given.

Remember that transition probabilities can be shown in a matrix in the discrete case (see (3.11)). The following definition will extend this concept to the continuous case, where this matrix is infinite-dimensional and will be given by a function called transition kernel.

Definition 3.1.1 (Transition kernel). *Let $\Omega \subset \mathbb{R}^l$, for some positive integer l, be a sample space. A transition kernel is a function K defined on $\Omega \times \mathcal{B}(\Omega)$ such that $K(x, \cdot)$ is a probability measure on $\mathcal{B}(\Omega)$.*

Remark 66. In the discrete case, the probability of the transition from $x \in \Omega$ to $y \in \Omega$ is given by the element $K_{x,y}$ of the transition matrix. In the continuous case, the transition probability from $x \in \Omega$ to $\mathcal{A} \in \mathcal{B}(\Omega)$ is given by the transition kernel evaluation $K(x, \mathcal{A})$.

As it has been done in the example where Ω was discrete (see last section 3.1.5.2), the probability that the chain remains at the same point can be strictly positive. Consider

$$\begin{aligned} t : \Omega^2 &\to \mathbb{R}^+ \\ (x,y) &\mapsto t(x,y) \end{aligned}$$

and

$$\begin{aligned} r : \Omega^2 &\to \mathbb{R}^+ \\ x &\mapsto r(x) \end{aligned}$$

such that,

$$K(x, dy) = t(x,y)dy + r(x)\delta_x(dy), \tag{3.13}$$

where $\delta_x(dy) = 1$ if $x \in dy$ and 0 otherwise.

Remark 67. Note that $\int_\Omega t(x,y)dy < 1$ if $r(x) > 0$. Therefore, $t(x, \cdot)$ is not necessarily a probability density function.

Remark 68. Notice that

$$r(x) = 1 - \int_\Omega t(x,y)dy \tag{3.14}$$

is the probability that the chain remains at x.

Define the detailed balance condition for a continuous probability density $f(x), x \in \Omega$ and a transition kernel (3.13) by

$$f(x)t(x,y) = f(y)t(y,x) \tag{3.15}$$

Proposition 3.1.6. *If a transition kernel K (definition 3.1.1) satisfies the detailed balance condition (3.15) for a probability density $f(x)$, then $f(x)$ is the stationary density of K.*

Proof. Note that,

$$\begin{aligned}
\int_\Omega K(x,\mathcal{A})f(x)dx &= \int_\Omega \left(\int_\mathcal{A} t(x,y)dy + r(x)\delta_x(dy)\right) f(x)dx \\
&= \int_\Omega \int_\mathcal{A} t(x,y)f(x)dydx + \int_\Omega \delta_x(\mathcal{A})r(x)f(x)dx \\
&= \int_\mathcal{A} \int_\Omega t(x,y)f(x)dxdy + \int_\mathcal{A} r(x)f(x)dx \\
&= \int_\mathcal{A} \int_\Omega \underbrace{t(y,x)f(y)}_{\text{detailed balance}} dxdy + \int_\mathcal{A} r(x)f(x)dx \\
&= \int_\mathcal{A} (1-r(y))f(y)dy + \int_\mathcal{A} r(x)f(x)dx \qquad (3.14) \\
&= \int_\mathcal{A} f(y)dy,
\end{aligned}$$

for any $\mathcal{A} \in \mathcal{B}(\Omega)$. □

Let $\{g(x,\cdot)\}_{x\in\Omega}$ be a set of probability density functions indexed by x. Suppose that samples according to those densities can be generated. Those PDFs will be called proposal PDFs.

For a desired PDF $f(x)$ consider the problem of finding

$$\begin{aligned}
\alpha : \Omega^2 &\to [0,1] \\
(x,y) &\mapsto \alpha(x,y),
\end{aligned}$$

such that,

$$t(x,y) = g(x,y)\alpha(x,y),$$

satisfies the detailed balance condition (3.15).

If the proposal densities $g(x,y)$ already satisfies the detailed balance, then setting $t(x,y) = g(x,y)$ and $r(x) = 0, \forall x \in \Omega$ will solve the problem, i.e., choosing a starting point x_0 and making transition according to $x_{i+1} \sim g(x_i,\cdot)$ will converge to the desired PDF. Unfortunetely, in general, $t(x,y) = g(x,y)$ would not satisfy the detailed balance. Metropolis et al. (1953) propose the following choice,

$$\alpha(x,y) = \min\left(\frac{f(y)g(y,x)}{f(x)g(x,y)}, 1\right), \qquad (3.16)$$

which is the key of M-H algorithm.

Remark 69. It is assumed that Ω is set such that $f(x) > 0, \forall x \in \Omega$, and $g(x,y) > 0, \forall x,y \in \Omega$. If not, Metropolis choice should be,

$$\alpha(x,y) = \begin{cases} \min\left(\frac{f(y)g(y,x)}{f(x)g(x,y)}, 1\right), & \text{if } f(x)g(x,y) > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Remark 70. Using Metropolis transitions, $f(x)$ does not need to be normalized because of the cancelation in the quotient $\frac{f(y)g(y,x)}{f(x)g(x,y)}$.

Proposition 3.1.7. *Let $f(x), x \in \Omega$, be a function proportional to a desired PDF. Let $\{g(x,\cdot)\}_{x\in\Omega}$ be a collection of proposal densities indexed by x. If*

$$\alpha(x,y) = \min\left(\frac{f(y)g(y,x)}{f(x)g(x,y)}, 1\right), \tag{3.17}$$

then the desired PDF (proportional to $f(x)$) is the stationary density of the transition kernel (3.13) according to

$$t(x,y) = g(x,y)\alpha(x,y). \tag{3.18}$$

Proof. By proposition 3.1.6, it is sufficient to prove the detailed balance condition (3.15). Substituting according to (3.18), it can be rewritten as

$$\frac{g(x,y)f(x)}{g(y,x)f(y)} = \frac{\alpha(y,x)}{\alpha(x,y)},$$

which is satisfied by the choice of α (3.17). □

M-H uses Metropolis choice of α (3.16). Proposition 3.1.7 justifies this choice. Algorithm 7 shows the pseudocode.

```
inputs: Function $f(x)$ proportional to a PDF; proposal
  densities $g(x,\cdot)$; maximum number of iterations $n_{max}$; initial
  state $x_0$;
for $i = 1, \cdots, n_{max}$ do
    // Sample from $g(x_{i-1},\cdot)$
    $y \sim g(x_{i-1},\cdot)$;
    // Sample from a uniform distribution
    $u = \text{ran}(0,1)$;
    // Compute the acceptance ratio
    $a = \min\left(\frac{f(y)g(y,x)}{f(x)g(x,y)}, 1\right)$;
    if $u \leq a$ then
        // Accept
        $x_i = y$;
    else
        // Reject
        $x_i = x_{i-1}$;
    end
end
output: $\{x_0, x_1, x_2, \cdots\}$;
```

Algorithm 7: Metropolis-Hastings (M-H) algorithm. Method to sample according to the target PDF $\propto f(x)$ from proposals $g(x,\cdot)$. It assumes that samples according to $g(x,\cdot)$, for any x in the sample space, can be generated.

Remark 71. Notice the connection between the acceptance-rejection step in M-H (algorithm 7) and the acceptance-rejection sampling (algorithm 6).

Remark 72. Usually, a number of points at the beginning of the chain are discarded after running M-H (algorithm 7). This is to reduce the effect of correlation with the fixed initial point. Those first iterations are called burning period.

Proposition 3.1.7 is not sufficient proof for the convergence to the desired density. Full proof of convergence needs the use of Markov chain theory. As mentioned in last section 3.1.5.2, it is out of the scope of this work and only justification for the detailed balance is given. The other necessary conditions are irreducibility and aperiodicity. Informally, the first one means that the chain must be able to move from any area of the sample space to any other area of the sample space in a finite number of moves with non-zero probability. The second one means that this finite number of moves must not be required to be a multiple of some integer. The fulfillment of both conditions is called ergodicity. Ergodicity can be satisfied in M-H by setting proposal densities with strictly positive values on any point of Ω. More details about the theory can be found in, e.g., Robert and Casella (2004), Chib and Greenberg (1995) or Smith and Roberts (1993).

Improvements of M-H focus on finding proposal densities $g(x, \cdot)$ which ensure a faster convergence. For example, adaptative M-H tune the parameters of the proposal densities according to the chain history during the algorithm run, taking special care in not breaking the ergodicity condition (see Haario, Saksman, and Tamminen, 2001).

In addition to that, there are other improvements. Two important algorithms related to M-H[9] are Gibbs sampling (see, e.g., Casella and George, 1992) and Hamiltonian Monte Carlo (HMC) (see, e.g., Betancourt, 2018; or Brooks, 2011).

Gibbs sampling uses the conditional probabilities when they are easier to simulate.

HMC uses hamiltonian dynamics to reduce correlation between successive samples[10]. However, the derivatives are needed to run HMC due to the hamiltonian procedure, and therefore it is unsuitable when they are not available.

3.1.6 Sampling with variables' constraints

This section will give a method to use the sampling algorithms exposed in this Chapter when there are constraints in the random variables. The ideas of this section about sampling from a manifold given by variables constraints are novel ideas of this thesis. They are based on measure theory. Sampling with variables' constraints can be relevant, e.g., when designing proposal distributions (see section 3.1.5.3) in constrained spaces and satisfying the detailed balance condition

[9] In fact, they are particular cases of M-H.

[10] It is specially useful for computing integrals because a fewer number of samples are needed for obtaining a good approximation.

(see 3.15). Or, from another perspective, conditioning a random variable to be in a manifold [11] (see remark 87). This problem was motivated by the mixture weight space (see (3.30)) that will be presented in section 3.2.4.2.

Let $f(x)$ be the PDF of a random variable

$$x = (x_1, \ldots, x_m) \in \Omega \subset \mathbb{R}^m.$$

Remark 73. Remember the abuse of notation explained in section 1.4. There is no distinction in notation between the random variable x and the variable of the PDF f.

Let $\mathcal{A} \subset \Omega$ be a k-dimensional manifold in $\mathbb{R}^m$ with $k < m$. This manifold can be given, e.g., by variables' constraints, $g(x) = 0$.

3.1.6.1 A generalization of the conditional PDF

The aim is sampling according to the PDF $f(x)$ and conditioning to $x \in \mathcal{A}$.

The measure-theoretic approach of probability theory is necessary for tackling this problem. See, e.g., Bobrowski (2005) for more details measure about the connection between measure theory and probability theory.

Let $(\Omega \subset \mathbb{R}^m, \mathcal{B}(\Omega), \mathbb{P})$ be the probability space of the random variable x. As usual, if there are no indications, PDFs defined in this work are defined w.r.t. the Lebesgue measure in $\mathbb{R}^m$, in the sense that

$$\mathbb{P}(x \in A) = \int_A d\mathbb{P}(x) = \int_A f(x)dx.$$

Notice that the Lebesgue measure of $\mathcal{A}$ is equal to 0 since $k < m$. Therefore, $\mathbb{P}(x \in \mathcal{A}) = 0$. Conditioning to events of probability 0 is not trivial. Borel–Kolmogorov paradox is an example (see, e.g., Rescorla (2015)).

Remark 74. Let $x = (x_1, x_2) \in \mathbb{R}^2$ be a random variable with PDF f_x w.r.t. the Lebesgue measure in $\mathbb{R}^2$. Note that the PDF $f_{x_1|x_2}(x_1)$ of x_1 conditioned to $x_2 = a$ is well known. It is

$$p(x_1|x_2 = a) = \frac{p(x_1, a)}{\int_{\mathbb{R}} p(x_1, a)dx_1}$$

[11] A necessary condition will be that this constrained space must be a manifold.

w.r.t. the Lebesgue mesure in $\mathbb{R}$. And note that the Lebesgue measure in $\mathbb{R}^2$ of

$$\left\{(x_1, x_2) \in \mathbb{R}^2 | x_2 = a\right\}$$

is 0.

Definition 3.1.2. *Let $x = (x_1, \ldots, x_m) \in \Omega \subset \mathbb{R}^m$ be a random variable. Let $\mathcal{A} \subset \Omega$ be a manifold in $\mathbb{R}^m$. Let $f_x(x)$ be the PDF of x w.r.t. the Lebesgue measure in $\mathbb{R}^m$. Let μ be the Lebesgue measure in the manifold $\mathcal{A}$. Define the conditional PDF $f_{x|x\in\mathcal{A}}(x)$ of x conditioned to $x \in \mathcal{A}$ w.r.t μ as*

$$f_{x|x\in\mathcal{A}}(x) = \frac{f_x(x)}{\int_{\mathcal{A}} f_x(x) d\mu(x)}. \tag{3.19}$$

Remark 75. Note that the PDF (3.19) is defined w.r.t. μ, the Lebesgue measure in the manifold $\mathcal{A}$. Therefore, if $C \subset \mathcal{A}$,

$$\mathbb{P}(x \in C | x \in \mathcal{A}) = \frac{\int_C f_x(x) d\mu(x)}{\int_{\mathcal{A}} f_x(x) d\mu(x)}.$$

Remark 76. Note that definition 3.1.2 agrees with the "usual" conditional probability (remark 74) as a particular case.

3.1.6.2 Parametrization of $\mathcal{A}$

Assume that the manifold $\mathcal{A}$ can be parametrized with only one coordinate chart ϕ,

$$\begin{aligned} \phi : \mathcal{U} \subset \mathbb{R}^k &\to \phi(\mathcal{U}) = \mathcal{A} \\ u &\mapsto \phi(u). \end{aligned} \tag{3.20}$$

Notice that the Lebesgue measure in the manifold is

$$\mu(A) = \int_{\phi^{-1}(A)} \sqrt{\left|\det\left(J_\phi(u)^T J_\phi(u)\right)\right|} du,$$

for all μ-medible sets $A \subset \mathcal{A}$, where J_ϕ is the Jacobian of the parametrization ϕ.

The measure μ can be seen as a measure in the parameters, i.e., a measure in $\mathcal{U} \subset \mathbb{R}^k$,

$$\hat{\mu}(B) = \mu(\phi(B))$$

for all $\hat{\mu}$-medible sets $B \subset \mathcal{U}$.

3.1.6.3 PDF w.r.t. the Lebesgue measure on the parameters u

The Radon-Nikodym derivative of $\hat{\mu}$ with respect to the Lebesgue measure in $\mathbb{R}^k$ is

$$\frac{d\hat{\mu}}{du} = \sqrt{|\det\left(J_\phi(u)^T J_\phi(u)\right)|}, \tag{3.21}$$

and therefore

$$\begin{aligned}
\mathbb{P}(x \in C | x \in \mathcal{A}) &= K \int_C f_x(x) d\mu(x) \\
&= K \int_{\phi^{-1}(C)} f_x(\phi(u)) d\hat{\mu}(u) \\
&= K \int_{\phi^{-1}(C)} f_x(\phi(u)) \frac{d\hat{\mu}}{du} du \\
&= K \int_{\phi^{-1}(C)} f_x(\phi(u)) \sqrt{|\det\left(J_\phi(u)^T J_\phi(u)\right)|} du.
\end{aligned} \tag{3.22}$$

where the normalization constant is

$$K = \frac{1}{\int_{\mathcal{U}} f_x(\phi(u)) \sqrt{|\det\left(J_\phi(u)^T J_\phi(u)\right)|} du}.$$

Remember that the aim is to sample according to a PDF $f_x(x)$ conditioned to a manifold $x \in \mathcal{A}$. The method is as follows:

1. Let
$$f_u(u) = K f_x(\phi(u)) \sqrt{|\det\left(J_\phi(u)^T J_\phi(u)\right)|}$$
be a PDF on the parameters.

2. Generate samples according to
$$x = \phi(u),\ u \sim f_u(u).$$

Remark 77. The sampling methods explained in section 3.1.5 can be used to sample from f_u. Remember that using these techniques the constant K plays no role.

Example 5. Let $\Omega = (-11, 11) \times (-2, 2)$. Let

$$\mathcal{A} \subset \Omega \subset \mathbb{R}^2$$

be the ellipse given by the image of the following coordinate chart:

$$\begin{array}{rrcl} \phi : & (0,2\pi) & \to & \phi(0,2\pi) = \mathcal{A} \subset \mathbb{R}^2 \\ & u & \mapsto & (x,y) = \phi(u) = (10\cos(u), \sin(u)). \end{array}$$

Let $f_{x,y}(x,y) = C$ (C constant) be the PDF of a uniform distribution in Ω. Consider the problem of obtaining samples from $f_{x,y}$ conditioning to $(x,y) \in \mathcal{A}$. In other words, the problem of sampling uniformly in the ellipse $\mathcal{A}$ [12]. A naive approach is to generate samples according to

$$(x,y) = \phi(u),\ u \sim f_u(u) = f_{x,y}(\phi(u)).$$

Figures 3.2 and 3.3 show that this method does not give a uniform distribution in the ellipse. The coordinate chart ϕ expands and shrinks volumes, and that alters the probability density. The term derived in (3.21) is needed to compensate it [13]. Following the derivations in this section, a suitable approach would be

$$\begin{aligned} &(x,y) = \phi(u), \\ &u \sim f_u(u) \propto f_{x,y}(\phi(u))\sqrt{|\det\left(J_\phi(u)^T J_\phi(u)\right)|} \\ &\qquad\qquad \propto \|\phi'(u)\| \\ &\qquad\qquad \propto \sqrt{10^2\sin^2(u) + \cos^2(u)}. \end{aligned}$$

Figures 3.4 and 3.5 show samples obtained by this method.

[12] I.e., the probability of two segments of the ellipse is the same if and only if the arc length of the two segments is the same. Notice that the arc length is the Lebesgue measure in the ellipse μ.

[13] It is the same situation as the change of variables rule and the Jacobian determinant.

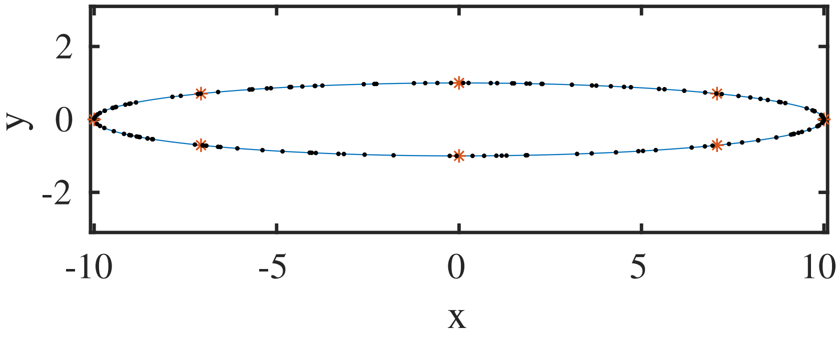


FIGURE 3.2: 150 samples of $\phi(u) = (10\cos(u), \sin(u))$, $u \sim \text{Uniform}(0, 2\pi)$. The red points represent $\phi(\frac{n\pi}{4})$, $n \in \{0, 1, 2, 3, 4, 5, 6, 7\}$.

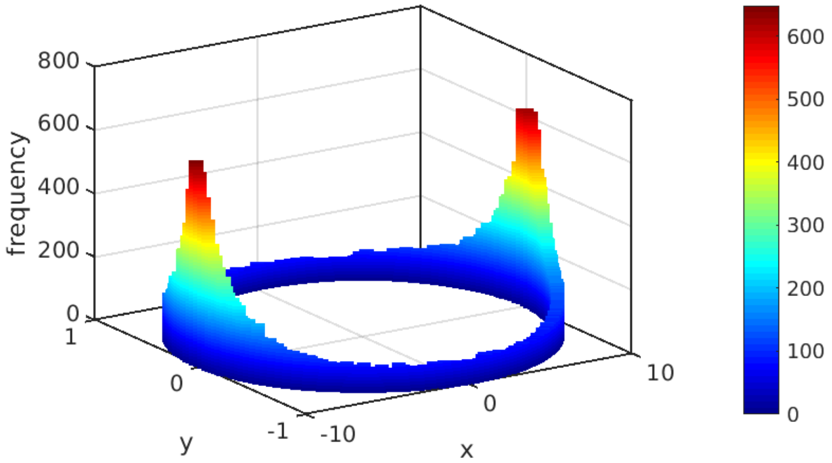


FIGURE 3.3: 10^5 samples of $\phi(t) = (10\cos(t), \sin(t))$, $t \sim \text{Uniform}(0, 2\pi)$, in an histogram whose bins on the ellipse have the same arc length (10^3 bins).

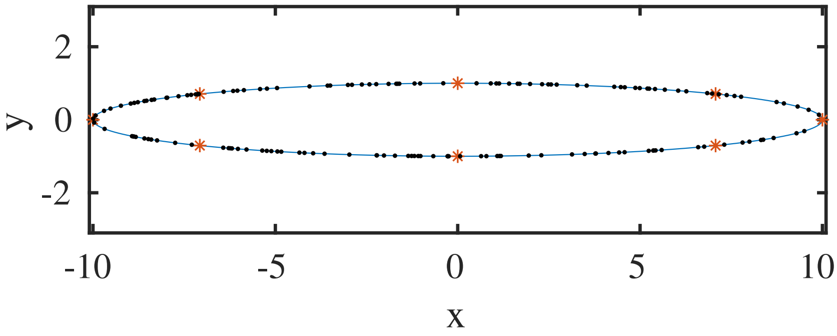


FIGURE 3.4: 150 samples of $\phi(u) = (10\cos(u), \sin(u))$, $u \sim f_u(u) \propto \sqrt{|\det(J_\phi^T(u)J_\phi)(u)|} = \|\phi'(u)\|$. The red points represent $\phi(\frac{n\pi}{4})$, $n \in \{0,1,2,3,4,5,6,7\}$.

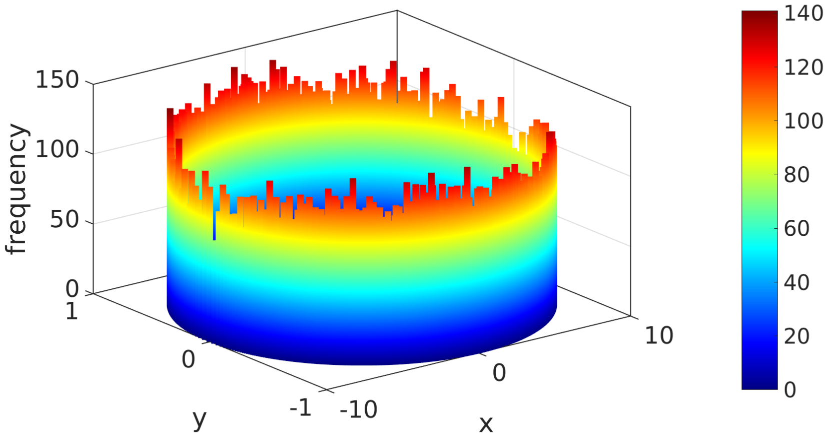


FIGURE 3.5: 10^5 samples of $\phi(t) = (10\cos(t), \sin(t))$, $u \sim f_u(u) \propto \sqrt{|\det(J_\phi^T(u)J_\phi)(u)|} = \|\phi'(u)\|$, in an histogram whose bins on the ellipse have the same arc length (10^3 bins).

3.2 A probabilistic framework for robust inverse design under uncertainty

The problem of building a stochastic surrogate model from data[14] has been addressed in Chapter 2. It focused in GP and the mathematical foundations of learning theory.

The main novelty of this thesis is given in this section with the development of a framework for design optimization. Its aim is to tackle the inverse problem of finding a probability distribution in the input space of a surrogate that satisfies a prescribed performance in the output when uncertainties are propagated.

The aim is to provide a framework that can be used with any surrogate model. Therefore, no assumptions will be taken in the model. It will be considered as a black-box which generates an output given an input and no more information is known about its properties.

The notation $\mathcal{X}$ will be used to refer to the space of inputs and the notation $\mathcal{Y}$ will be used to refer to the space of outputs.

Either it is deterministic or stochastic, the only assumption taken on the surrogate model is that given samples $\{x_1, \ldots, x_n\}$ from a particular probability distribution on $\mathcal{X}$, it can propagate uncertainties generating samples $\{y_1, \ldots, y_n\}$ in $\mathcal{Y}$. See the following diagram:

$$x \in \mathcal{X} \to \boxed{\substack{\textit{SURROGATE}\\ \textit{MODEL}}} \to y(x) \in \mathcal{Y}$$

The notation $\boldsymbol{x} = \{x_1, \ldots, x_n\}$ will be used to refer to a set of samples in the input space of a surrogate model. The notation $\boldsymbol{y} = \{y_1, \ldots, y_n\}$ will be used to refer to a set of samples in the output space. The notation $\boldsymbol{y}(\boldsymbol{x}) = \{y(x_1), \ldots, y(x_n)\}$ will be used to denote that $\boldsymbol{y}(\boldsymbol{x})$ are samples in the output space propagated from the set of samples $\boldsymbol{x}$ through the surrogate model.

3.2.1 The general framework

A parametric family of probability density functions $\mathcal{F}$ indexed by a parameter $\lambda \in S(\lambda)$ is set in $\mathcal{X}$,

$$\mathcal{F} = \{f_\lambda \mid \lambda \in S(\lambda)\}.$$

[14] For example, data from computationally expensive simulations.

Consider a function,

$$\begin{aligned}\psi : \mathcal{Y} \times S(\lambda) &\to \mathbb{R}\\ (y, \lambda) &\mapsto \psi(y, \lambda),\end{aligned}$$

and the optimization problem of finding,

$$\lambda_{\text{opt}} = \underset{\lambda \in S(\lambda)}{\operatorname{argmin}} \; \mathbb{E}_{x \sim f_\lambda} \left[\psi(y(x), \lambda) \right]. \tag{3.23}$$

Remark 78. The function ψ must be defined according to the target performance.

Remark 79. The problem of finding a distribution in the input space $\mathcal{X}$ that optimizes the expectation of ψ is reduced to the problem of finding the parameters λ which lead to the optimal distribution within the family $\mathcal{F}$. An important assumption has been taken with this parametric approach. It may well be that no member of the family $\mathcal{F}$ gives a satisfactory performance. Therefore, the choice of $\mathcal{F}$ is decisive.

For instance, consider

$$\begin{aligned}\psi(y(x), \lambda) &= \psi_y(y(x)) + \gamma \psi_\lambda(\lambda),\\ \mathbb{E}_{x \sim f_\lambda} \left[\psi(y(x), \lambda) \right] &= \mathbb{E}_{x \sim f_\lambda} \left[\psi_y(y(x)) \right] + \gamma \psi_\lambda(\lambda),\end{aligned}$$

for some constant $\gamma \in \mathbb{R}^+$, and the two following examples.

Example 6. Let the surrogate model $y(x) = (y_1(x), y_2(x))$ be two quantities of interest which are wanted to be minimized and are given by the design variables $x \in \mathcal{X} = \mathbb{R}^n$. In addition, it is required to have as much flexibility as possible in the design. The parametric family of densities $\mathcal{F}$ is decided to be the family of n-dimensional multivariate normal densities indexed by the mean and the covariance matrix, i.e., $\lambda = (\mu, \Sigma)$. Let

$$\psi_y(y(x)) = \|y(x)\|, \;\; \psi_\lambda(\mu, \Sigma) = \frac{1}{\det(\Sigma)}.$$

The optimization problem becomes,

$$\begin{aligned}(\mu_{opt}, \Sigma_{opt}) &= \underset{\mu,\Sigma}{\operatorname{argmin}} \ \mathbb{E}_{x\sim\mathcal{N}(\mu,\Sigma)} \left[\psi(y(x),\lambda)\right] \\ &= \underset{\mu,\Sigma}{\operatorname{argmin}} \ \mathbb{E}_{x\sim\mathcal{N}(\mu,\Sigma)} \left[\|y(x)\|\right] + \frac{\gamma}{\det(\Sigma)}.\end{aligned}$$

The norm $\|y(x)\|$ can be given by specific features of the design problem, e.g.,

$$\|y(x)\|_W = y(x)^T W y(x)$$

for some positive-definite matrix W. The flexibility on the design variables x can be measured by another quantity derived from the covariance matrix Σ, e.g., the lowest eigenvalue.

Example 7. Consider the same situation than example 6. However, instead of minimizing $\|y(x)\|$, the quantities given by $y(x)$ are wanted to be (with high probability) in an specific region $A \subset \mathcal{Y}$, which is considered a region of "good" performance. Consider,

$$\psi_y(y(x)) = \mathbb{1}_{\mathcal{Y}\setminus A}(y(x)) = \begin{cases} 1 & \text{if } y(x) \in \mathcal{Y}\setminus A, \\ 0 & \text{otherwise.} \end{cases}$$

Thus,

$$\begin{aligned}(\mu_{opt}, \Sigma_{opt}) &= \underset{\mu,\Sigma}{\operatorname{argmin}} \ \mathbb{E}_{x\sim\mathcal{N}(\mu,\Sigma)} \left[\psi(y(x),\lambda)\right] \\ &= \underset{\mu,\Sigma}{\operatorname{argmin}} \ \mathbb{E}_{x\sim\mathcal{N}(\mu,\Sigma)} \left[\mathbb{1}_{\mathcal{Y}\setminus A}(y(x))\right] + \frac{\gamma}{\det(\Sigma)}. \\ &= \underset{\mu,\Sigma}{\operatorname{argmin}} \ \mathbb{P}_{x\sim\mathcal{N}(\mu,\Sigma)} \left(y(x) \notin A\right) + \frac{\gamma}{\det(\Sigma)}.\end{aligned}$$

The main problem of tackling (3.23) is that no information about the surrogate model is given. However, it is possible to propagate uncertainties with point evaluations. The expectation will be approximated with some sample statistic from a sample $\boldsymbol{y}(\boldsymbol{x})$ where,

$$\begin{gathered}\boldsymbol{x} = \{x_1, \ldots, x_n\}, \\ x_i \text{ i.i.d. }, \ x_i \sim f_\lambda, \ \forall\, 1 \leq i \leq n.\end{gathered}$$

The notation $\boldsymbol{x}_\lambda = \{x_{\lambda,1}, \dots, x_{\lambda,n}\}$ will be used to denote that the samples have been generated from f_λ. In the same fashion,

$$\boldsymbol{y}_\lambda = \{y_{\lambda,1}, \dots, y_{\lambda,n}\} = \boldsymbol{y}(\boldsymbol{x}_\lambda).$$

For instance, the expectation in example 6 can be approximated by,

$$\mathbb{E}_{x\sim\mathcal{N}(\mu,\Sigma)}\left[\|y(x)\|\right] \approx \frac{1}{n}\sum_{i=1}^{n}\|y_{\mu,\Sigma,i}\|,$$

and, in example 7 by,

$$\mathbb{E}_{x\sim\mathcal{N}(\mu,\Sigma)}\left[\mathbb{1}_{\mathcal{Y}\setminus A}(y(x))\right] = \mathbb{P}_{x\sim\mathcal{N}(\mu,\Sigma)}\left(y(x)\notin A\right) \approx \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\mathcal{Y}\setminus A}(y_{\mu,\Sigma,i}).$$

Since it has been assumed that the problem can only be tackled with samples, the objective function in the optimization problem will be redefined using the samples as input variables,

$$\begin{aligned} H : \mathcal{Y}^n \times S(\lambda) &\to \mathbb{R} \\ (y_1, \dots, y_n, \lambda) = (\boldsymbol{y}, \lambda) &\mapsto H(\boldsymbol{y}, \lambda), \end{aligned}$$

and therefore,

$$\widehat{\lambda}_{\text{opt}} = \underset{\lambda\in S(\lambda)}{\operatorname{argmin}}\ H(\boldsymbol{y}_\lambda, \lambda). \tag{3.24}$$

Estimators can be used to define H. For instance, in example 6,

$$H(\boldsymbol{y}, \mu, \Sigma) = \frac{1}{n}\sum_{i=1}^{n}\|y_i\| + \frac{\gamma}{\det(\Sigma)}, \tag{3.25}$$

and in example 7,

$$H(\boldsymbol{y}, \mu, \Sigma) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{\mathcal{Y}\setminus A}(y(x_i)) + \frac{\gamma}{\det(\Sigma)}. \tag{3.26}$$

A choice of a parametric family $\mathcal{F}$ and an objective function H uniquely determine the optimization problem (3.24). Function H becomes the object of interest and it must be defined according to the desired performance in each particular case. Although the notation $\widehat{\lambda}_{\text{opt}}$ was used in (3.24) to differentiate from (3.23), the notation without hat will always refer to the solution of (3.24) in the following sections.

The following sections will give methods to tackle (3.24) (see section 3.2.2), potential objective functions H when the aim is to approximate a target PDF in $\mathcal{Y}$ (see section 3.2.3), and finally, two potential parametric families of distributions $\mathcal{F}$ (see section 3.2.4).

Remark 80. Notice that the objective function H in (3.24), in fact, only depends on λ since y_λ depends on λ. However, the samples $\boldsymbol{y}_\lambda$ are not uniquely determined since they are generated by a random process. Remark 81 gives more details about how this issue affects the optimization problem (3.24).

Remark 81. Notice that the optimal value $\widehat{\lambda}_{\text{opt}}$ in the optimization problem 3.24 is not well defined. The objective function H can be chosen to ensure the existence of a global minimum in some point $(\boldsymbol{y}, \lambda)$. However, the samples $\boldsymbol{y}_\lambda$ depend on λ and on the surrogate, and they are not uniquely determined since it also depends on a random process (sampling from f_λ).

When uncertainties are propagated through the surrogate, not all information given by the distribution f_λ is used, only the information given by the samples is propagated. The minimum in 3.24 can change depending on the samples that are used for each possible distribution f_λ. Methods which uniquely determine the samples [15] $\boldsymbol{y}_\lambda$ for each distribution f_λ will be given (see remark 88) and it will be assumed that the minimum exists for those choices. In fact, the aim will not be to find the best value of λ, finding a "good" value will be considered sufficient. The optimization algorithms that will be proposed do not ensure finding the minimum even if it exists. However, they are able to explore the space $S(\lambda)$ efficiently with the objective to find, at least, a satisfactory solution.

3.2.2 Stochastic optimization

The aim of this section is to give a method to solve the optimization problem 3.24.

On the one hand, the method should satisfy two conditions which are necessary to succeed in the stated minimization problem:

[15] This concept may be confusing because randomness is intrinsic in the process of sampling. The idea is to give a method that generates the same samples if the same distribution is simulated in different occasions. This choice is the representation of the distribution by samples and will never change during the algorithm.

1. The method must not require any assumption on the objective function. For example, gradient based methods are not suitable since regularity conditions can not be assumed. This is because no information about the surrogate model is given.

2. The injectivity of the model can not be assumed and the possibility of many local minimum should be considered. A method able to escape from a local minimum is necessary.

In addition, its use has to be feasible in terms of computational cost when $S(\lambda)$ is a high-dimensional space.

On the other hand, guaranteeing the global minimum is not necessary, it will be sufficient to find a satisfactory solution.

Algorithms in stochastic optimization (see, e.g., Spall (2003); or Žilinskas and Zhigljavsky (2016)), which introduce randomness in the search process, are an option which satisfy the necessary conditions stated above. In this document, the algorithm of Simulated Annealing (SA) will be explained (section 3.2.2.1). However, other methods can be valid or even show a better performance than SA. There are many options in the field of optimization, and other approaches could be explored. It also depends on the specific problem. SA offers a general solution which is the goal of this work.

3.2.2.1 Simulated Annealing

Simulated Annealing (SA) is based in M-H (section 3.1.5.3) to find the global minimum of an objective function $F(\lambda), \lambda \in S(\lambda)$ exploring the space $S(\lambda)$.

Algorithm 7 is used with

$$f(\lambda) = E(\lambda) = e^{-\frac{F(\lambda)}{T}}. \tag{3.27}$$

The parameter T is usually called temperature and $f(\lambda)$ is called the energy at λ (the notation $E(\lambda)$ is commonly used in SA theory).

Remark 82. Notice that e^{-t} is strictly monotonically decreasing, and therefore the global minimum of $F(\lambda)$ is the highest mode of $E(\lambda)$.

The main difference between SA and M-H using (3.27) is that the parameter T is decreased in each iteration (or every k iterations). Note that decreasing T also decrease the probability of accepting worse solutions. At the beginning, a high value of T leads to a high acceptance ratio. This allows the algorithm to escape from a local minimum in the first iterations, and therefore it can explore the whole space $S(\lambda)$ (ideally). However, temperature T progressively decreases to zero. Thus, in the last iterations, the probability of accepting a worse solution is almost zero converging to the global minimum (ideally).

Remark 83. The convergence or transitions are not affected by the continuity or differentiability of function $F(\lambda)$. Therefore, SA can be used even when regularity conditions cannot be guaranteed.

The acceptance criteria can be the same than M-H,

$$p(\lambda \to \lambda'|T) = \min\left(e^{-\frac{F(\lambda')-F(\lambda)}{T}}, 1\right), \ (see(3.16)),$$

considering the symmetry of the proposal densities

$$g(\lambda, \lambda') = g(\lambda', \lambda).$$

Or another option can be chosen. It does not need to satisfy the detailed balance condition since the goal is not simulating a distribution. In fact, the acceptance criteria can be deterministic (see Dueck and Scheuer (1990); and Franz, Hoffmann, and Salamon (2001)).

The way T decreases plays an important role in the successful convergence of SA. It is called cooling schedule. Most of the improvements in SA are based on improvements of the cooling schedule, e.g., adaptative SA (see, Ingber (1989); and Ingber (2000)). More information of SA can be found in, e.g., Henderson, Jacobson, and Johnson (2003) or Rao (2009).

```
inputs: Function $F(\lambda)$ to minimize; proposal distributions
  $g(\lambda,\cdot)$; maximum number of iterations $n_{max}$; initial point $\lambda_0$;
  initial temperature $T_0$; cooling schedule; acceptance criteria;
$T = T_0$;
for $i = 1, \cdots, n_{max}$ do
    // Sample from $g(\lambda_{i-1},\cdot)$
    $\lambda' \sim g(\lambda_{i-1},\cdot)$;
    if $\lambda'$ satisfies the acceptance criteria based in $F(\lambda_{i-1})$, $F(\lambda')$
     and $T$ then
        // Accept
        $\lambda_i = \lambda'$;
    else
        // Reject
        $\lambda_i = \lambda_{i-1}$;
    end
    Decrease $T$ according to the cooling schedule;
end
output: $\lambda_{n_{max}}$;
```

Algorithm 8: Simulated Annealing (SA) algorithm. Stochastic optimization method to find the global minimum of a function $F(\lambda)$.

3.2.3 Target PDF approximation

The objective function H in the optimization problem (3.24) must be defined according to the desired performance. For instance, two possible options (3.25) and (3.26) were given for two different scenarios (examples 6 and 7).

This section will cover in detail another scenario: when the desired performance is given by a target density in the output space $\mathcal{Y}$ of the surrogate model.

Let $f_T(y), y \in \mathcal{Y}$ be a target PDF in the output space of a surrogate model. The aim is to find $\lambda_{opt} \in S(\lambda)$ such that

$$x \sim f_{\lambda_{\text{opt}}} \Rightarrow y(x) \sim f_T,$$

or, at least, the PDF of $y(x)$ is as close [16] as possible to f_T.

3.2.3.1 A distance between probability distributions

Let $(\mathbb{P}, \mathbb{Q}) \mapsto d(\mathbb{P}, \mathbb{Q})$ be a distance between probability distributions. This distance should be able to be estimated from samples since this is the only information available of the distribution of $y(x)$.

Let $\boldsymbol{y} \mapsto \widehat{d}(\boldsymbol{y}, \mathbb{P}_{f_T})$ be an estimator of the distance d between the distribution which generated $\boldsymbol{y}$ and the distribution given by f_T.

Consider the objective function,

$$H(\boldsymbol{y}, \lambda) = \widehat{d}(\boldsymbol{y}_\lambda, \mathbb{P}_{f_T}) + \gamma H_\lambda(\lambda).$$

Following this approach, it is necessary to find a distance between distributions that can be estimated from samples. If not a distance, at least, a function able to measure the "closeness" of two probability distributions.

Traditional goodness-of-fit tests can be explored for this purpose. However, they can become computationally intractable in high dimensions. It is needed to run the test in each iteration on the optimization process (see section 3.2.2) when the objective function is evaluated. Therefore, the computational cost of the test determines if the exploration of $S(\lambda)$ is feasible.

Another important aspect is that goodness-of-fit tests are designed to fit a distribution into a sample and not the opposite [17].

Novel methods arose in the Machine Learning community that can solve some of those problems. In this work, it was decided to use Maximum Mean Discrepancy (MMD). Probability distributions become points in an RKHS and MMD is the distance given by the inner product in this Hilbert space.

3.2.3.2 Maximum Mean Discrepancy

MMD is the distance of a Hilbert space known as "kernel mean embedding" in the ML community. Probability distributions are mapped

[16] It will be shown that it is possible to set a distance between distributions. See section 3.2.3.1.

[17] For example, maximizing f_T will push the design to concentrate as much as possible the output density at the mode of f_T. Although it can be a desired behavior, this is not what it is wanted in this section. The aim is to approximate f_T, not only to concentrate the density around the mode.

into an RKHS $\mathcal{H}$ (the kernel mean embedding) through the following operator,

$$\phi(\mathbb{P}) = \mu_{\mathbb{P}} = \int_{\mathcal{Y}} K(y,\cdot)d\mathbb{P}(y),$$

where $K : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a PDS kernel (see remark 16).

Remark 84. In contrast to Chapter 2 where kernels were used in the input space of the surrogate model, the notation $\mathcal{Y}$ is used in this section to emphasize that the kernel mean embedding is applied to distributions in the output space.

It can be proved that the map $\mathbb{P} \mapsto \mu_{\mathbb{P}}$ is injective, and therefore

$$\|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}} = 0 \text{ if and only if } \mathbb{P} = \mathbb{Q}.$$

Definition 3.2.1. *Define the MMD as the distance in $\mathcal{H}$ between the mean embedding of two probability distributions $\mathbb{P}$ and $\mathbb{Q}$,*

$$MMD[\mathcal{H},\mathbb{P},\mathbb{Q}] = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}}.$$

It can be shown that,

$$\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}} \rangle_{\mathcal{H}} = \langle \mathbb{E}_{y\sim\mathbb{P}}\left[K(y,\cdot)\right], \mathbb{E}_{y\sim\mathbb{Q}}\left[K(y,\cdot)\right]\rangle = \mathbb{E}_{\substack{y\sim\mathbb{P}\\ y'\sim\mathbb{Q}}}\left[K(y,y')\right].$$

Thus,

$$\begin{aligned} MMD^2[\mathcal{H},\mathbb{P},\mathbb{Q}] &= \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|^2_{\mathcal{H}} \\ &= \langle \mu_{\mathbb{P}}, \mu_{\mathbb{P}}\rangle_{\mathcal{H}} - 2\langle \mu_{\mathbb{P}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{H}} + \langle \mu_{\mathbb{Q}}, \mu_{\mathbb{Q}}\rangle_{\mathcal{H}} \\ &= \mathbb{E}_{\substack{y\sim\mathbb{P}\\ y'\sim\mathbb{P}}}\left[K(y,y')\right] - 2\mathbb{E}_{\substack{y\sim\mathbb{P}\\ y\sim\mathbb{Q}}}\left[K(y,y)\right] + \mathbb{E}_{\substack{y\sim\mathbb{Q}\\ y'\sim\mathbb{Q}}}\left[K(y,y')\right]. \end{aligned}$$

More details about these derivations and the theory of kernel mean embedding of distributions can be found in, e.g., Muandet et al., 2017; or Song, 2008.

If

$$\begin{aligned} \boldsymbol{y}_T &= \{y_{T,1}, \dots, y_{T,m}\} \\ y_{T,i} &\sim f_T,\ 1 \le i \le m, \end{aligned}$$

are i.i.d. samples from the target PDF, and

$$\begin{aligned} \boldsymbol{y}(\boldsymbol{x}_\lambda) &= \{y_{\lambda,1}, \dots, y_{\lambda,n}\} \\ \boldsymbol{x}_\lambda = \{x_{\lambda,1}, \dots, x_{\lambda,n}\},\ x_{\lambda,i} &\sim f_\lambda,\ 1 \le i \le n, \end{aligned}$$

are i.i.d. samples from a distribution in the input space propagated through the surrogate, then consider,

$$H(\boldsymbol{y},\lambda) = \widehat{MMD}^2(\mathcal{H},\boldsymbol{y}(\boldsymbol{x}_\lambda),\boldsymbol{y}_T) + \gamma H_\lambda(\lambda).$$

where,

$$\begin{aligned}\widehat{MMD}^2(\mathcal{H},\boldsymbol{y}(\boldsymbol{x}_\lambda),\boldsymbol{y}_T) =& \frac{1}{m(m-1)}\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{n} K(y_{\lambda,1},y_{\lambda,j})\\ &+\frac{1}{n(n-1)}\sum_{i=1}^{m}\sum_{\substack{j=1\\ j\neq i}}^{m} K(y_{T,i},y_{T,j}) \qquad (3.28)\\ &-\frac{2}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m} K((y_{\lambda,1},y_{T,j}).\end{aligned}$$

is an estimator of MMD^2.

Remark 85. When implementing (3.28) in SA or another optimization algorithm, note that the double summation on m needs to be computed only once.

Remark 86. Assuming $m = n$, the complexity of (3.28) is $\mathcal{O}(n^2)$. SA (or another optimization algorithm) may not be able to explore the space $S(\lambda)$ if $n,m \gg 0$.

3.2.4 Parametric families of probability distributions

This section will give two options of families $\mathcal{F}$.

The two main restrictions in the selection of $\mathcal{F}$ are:

1. The distributions in $\mathcal{F}$ should have support equal to the input space $\mathcal{X}$.
2. Generating samples from the distributions in $\mathcal{F}$ should be computationally efficient.

Remark 87. If there are required constraints in the design variables, i.e., the input variables of the surrogate model, then families of distributions in the constrained space can be created from families of distributions in the ambient space using the theory explained in section 3.1.6.

3.2.4.1 Multivariate normal

In theory, it should be possible to generate samples from any PDF using the methods explained in section 3.1.5. However, it is necessary to simulate a PDF in each iteration of SA (or any other optimization algorithm). In addition, the evaluation of H can already be expensive (see, e.g., MMD (3.28)).

Algorithm 5, and algorithms 3 or 4, make a very efficient method to generate samples from any multivariate normal PDF.

In section 2.1.5, it was assumed the compactness of $\mathcal{X}$. In contrast, the support of a multivariate normal distribution is the whole Euclidean space. Constraints in $\mathcal{X}$ should not be a major issue using multivariate normal distributions if $\mathcal{F}$ is restricted such that most of the probability density of the normal distributions is concentrated on $\mathcal{X}$. For example, allowing only normal PDFs with the mean belonging to $\mathcal{X}$. An option to avoid possible evaluation problems in the optimization algorithm is to discard densities that generated a sample outside $\mathcal{X}$. In any case, if it is possible, it is recommended that the surrogate model can evaluate any point in the Euclidean space to avoid this issue.

Let $\mathcal{X} = \mathbb{R}^N$. The multivariate normal densities have two parameters: the mean and the covariance matrix (see section 3.1.4.2),

$$\lambda = (\mu, \Sigma),\ \mu \in \mathbb{R}^N,\ \Sigma \in \mathbb{R}^{N\times N} \text{ symmetric positive-definite.}$$

In propositions 3.1.3 and 3.1.4, it was proved that the covariance matrix of any non-degenerate normal distribution can be decomposed by Cholesky factorization,

$$\Sigma\ {}_{\substack{\text{symmetric and}\\ \text{positive-definite}}} \Rightarrow \Sigma = LL^T \begin{cases} L \text{ unique,} \\ L \text{ lower triangular } (l_{ij} = 0 \text{ if } i > j), \\ \text{positive diagonal entries } (l_{ii} > 0). \end{cases}$$

The implication to the left is also true, notice that,

$$v^T LL^T v = (L^T v)^T (L^T v) > 0,\ \forall v \in \mathbb{R}^N \setminus \{0\},\ \text{ and} \\ (LL^T)^T = LL^T.$$

Therefore, it can be considered

$$\lambda = (\underbrace{\mu_1, \dots, \mu_N,}_{N \text{ elements}} \underbrace{l_{21}, \dots, l_{N,N-1},}_{\substack{l_{ij},\ i>j, \\ \frac{N^2-N}{N} \text{ elements}}} \underbrace{l_{11}, \dots, l_{NN}}_{N \text{ elements}}) \in \mathbb{R}^{\frac{N^2+N}{2}} \times (\mathbb{R}^+ \setminus \{0\})^N$$

$$\mu = (\mu_1, \dots, \mu_N),\ L = (l_{ij})_{\substack{1 \le i \le N, \\ 1 \le j \le N}},\ \Sigma = LL^T. \tag{3.29}$$

instead.

Algorithm 9 gives the pseudocode to tackle the design problem of this section using multivariate normal densities as $\mathcal{F}$ and SA to optimize its parameters.

Remark 88. Notice that the process of generating samples is done only once, from $\mathcal{N}(0, I_{N \times N})$. There are no more uncertainties when the SA optimization starts. Therefore, the samples that are obtained for each choice of λ are uniquely determined before the SA process starts. See remark 81.

3.2.4.2 Mixtures

Let $\mathcal{F}_\alpha$ be a parametric family of probability distributions with parameter α with support $\mathcal{X}$. Consider the family of mixture densities whose M components are members of $\mathcal{F}_c$,

$$\mathcal{F} = \left\{ f_{\boldsymbol{\alpha},\boldsymbol{\pi}} = \sum_{i=1}^{M} \pi_i f_{\alpha_i} \;\middle|\; \begin{array}{l} f_{\alpha_i} \in \mathcal{F}_\alpha \\ \boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_M) \in S(\alpha)^M \\ \boldsymbol{\pi} = (\pi_1, \dots, \pi_M) \in \left\{ \sum_{i=1}^{M} \pi_i = 1,\ \pi_i \ge 0 \right\} \end{array} \right\},$$

where $\lambda = (\boldsymbol{\alpha}, \boldsymbol{\pi})$.

A mixture model mitigates the problem stated in remark 79. $\mathcal{F}$ is more flexible in order to approximate the true density that optimizes the objective function over the whole space of density functions with support $\mathcal{X}$. For example, it is well known that a mixture of normal densities can approximate any multivariate density with any degree of accuracy given enough components (see, e.g., Titterington, Smith, and Makov (1985)). I.e., mixtures of normal densities with a finite number of components is dense in the whole space of density functions. The more components, the more flexibility.

```
inputs: Surrogate model $\boldsymbol{y}(\boldsymbol{x})$; Objective function to
  minimize: $H(\boldsymbol{y}, \lambda)$, $\lambda \in \mathbb{R}^{\frac{N^2+N}{2}} \times (\mathbb{R}^+ \setminus \{0\})^N$; Proposal
  distributions $g(\lambda, \cdot)$; Maximum number of iterations $n_{max}$;
  Initial point $\lambda_0$; Initial temperature $T_0$; Cooling schedule;
  Acceptance criteria;
// In this pseudocode, $\mu$ and $L$ are obtained from $\lambda$.
   See (3.29).
$T = T_0$;
// Generate samples from $\mathcal{N}(0, I_{N\times N})$.  See Algorithms
   3 and 4.
$\boldsymbol{x}_s = \{x_{s,1}, \ldots, x_{s,n}\}$, $x_{s,i} \in \mathbb{R}^N$, $x_{s,i} \sim \mathcal{N}(0, I_{N\times N})$ i.i.d.;
// Transform to samples from $\mathcal{N}(\mu_0, L_0 L_0^T)$.  See
   section 3.1.4.2.
for $i = 1, \cdots, n$ do
    // $\boldsymbol{x}_0 = \{x_{0,1}, \ldots, x_{0,n}\}$
    $x_{0,i} = \mu_0 + L_0 x_{s,i}$;
end
// Evaluate in the surrogate model
$\boldsymbol{y}_0 = \boldsymbol{y}(\boldsymbol{x}_0)$;
for $i = 1, \cdots, n_{max}$ do
    // Sample from $g(\lambda_{i-1}, \cdot)$
    $\lambda_p \sim g(\lambda_{i-1}, \cdot)$;
    for $i = 1, \cdots, n$ do
        // $\boldsymbol{x}_p = \{x_{p,1}, \ldots, x_{p,n}\}$
        $x_{p,i} = \mu_p + L_p x_{s,i}$;
    end
    $\boldsymbol{y}_p = \boldsymbol{y}(\boldsymbol{x}_p)$;
    if $\lambda_p$ satisfies the acceptance criteria based in $H(\boldsymbol{y}_{i-1}, \lambda_{i-1})$,
      $H(\boldsymbol{y}_p, \lambda_p)$ and $T$ then
        // Accept
        $\lambda_i = \lambda_p$;
    else
        // Reject
        $\lambda_i = \lambda_{i-1}$;
    end
    Decrease $T$ according to the cooling schedule;
end
output: $\lambda_{n_{max}}$;
```

Algorithm 9: Algorithm to optimize the parameters of normal PDFs with SA. See section 3.2.2.1 for more details about SA.

Therefore, mixtures are interesting from a mathematical point of view. However, the suitability of mixtures will depend on each particular problem. A mixture may not give useful information for engineering design.

Algorithm 10 gives a method to generate samples from a mixture.

```
inputs: Weights $\{\pi_1, \dots, \pi_M\}$; Components $\{f_{\alpha_i}\}_{1 \le i \le M}$.
    // Generate a uniform random number on $(0,1)$. See section 3.1.2.
$u = rand()$;
$i = 1$;
while $u > \pi_i$ do
    $u = u - \pi_i$;
    $i = i + 1$;
end
// Generate a sample from the $i$th component.
$x \sim f_{\alpha_i}$;
output: $x$;
```

Algorithm 10: Algorithm to generate samples from a mixture PDF. It is assumed that it is possible to simulate the components of the mixture.

In the same fashion than 9, algorithm 11 gives the pseudocode of the stated problem using mixture densities.

Remark 89. Notice that the pseudorandom number generator is reset with the same seed just before generating samples in algorithm 11. Therefore, given a seed, the samples are uniquely determined. See remark 79.

Remark 90. A method to construct a proposal distribution on the space of weights,

$$\left\{ \boldsymbol{\pi} = (\pi_1, \dots, \pi_M) \,\middle|\, \sum_{i=1}^{M} \pi_i = 1,\ \pi_i \ge 0 \right\} \tag{3.30}$$

is to use a proposal distribution $g(\boldsymbol{\pi}_{\text{previous}}, \cdot)$ on $(\mathbb{R}^+)^N$ and normalize,

$$\hat{\boldsymbol{\pi}} = (\hat{\pi}_1, \dots, \hat{\pi}_M) \sim g(\boldsymbol{\pi}_{\text{previous}}, \cdot),$$
$$\boldsymbol{\pi}_{\text{next}} = \frac{\hat{\boldsymbol{\pi}}}{\sum_{i=1}^{M} \hat{\pi}_i}.$$

```
inputs: Surrogate model $\boldsymbol{y}(\boldsymbol{x})$; Objective function to minimize
  $H(\boldsymbol{y}, \lambda),\ \lambda = (\boldsymbol{\alpha}, \boldsymbol{\pi})$; Proposal distributions $g(\lambda, \cdot)$;
  Maximum number of iterations $n_{max}$; Initial point
  $\lambda_0 = (\boldsymbol{\alpha}_0, \boldsymbol{\pi}_0)$; Initial temperature $T_0$; Cooling schedule;
  Acceptance criteria; Seed of the pseudorandom number
  generator $S$;
$T = T_0$;
// Generate samples from $f_{\boldsymbol{\alpha}_0,\boldsymbol{\pi}_0}$. See algorithm 10
Reset the random number generator with seed equal to $S$;
$\boldsymbol{x}_0 = \{x_{0,1}, \ldots, x_{0,n}\},\ x_{0,i} \sim f_{\boldsymbol{\alpha}_0,\boldsymbol{\pi}_0}$ i.i.d.;
// Evaluate in the surrogate model
$\boldsymbol{y}_0 = \boldsymbol{y}(\boldsymbol{x}_0)$;
for $i = 1, \cdots, n_{max}$ do
    // Sample from $g(\lambda_{i-1}, \cdot)$
    $\lambda_p \sim g(\lambda_{i-1}, \cdot)$;
    // Generate samples from $f_{\boldsymbol{\alpha}_p,\boldsymbol{\pi}_p}$. See algorithm 10
    Reset the random number generator with seed equal to $S$;
    $\boldsymbol{x}_p = \{x_{p,1}, \ldots, x_{p,n}\},\ x_{p,i} \sim f_{\boldsymbol{\alpha}_p,\boldsymbol{\pi}_p}$ i.i.d.;
    $\boldsymbol{y}_p = \boldsymbol{y}(\boldsymbol{x}_p)$;
    if $\lambda_p$ satisfies the acceptance criteria based in $H(\boldsymbol{y}_{i-1}, \lambda_{i-1})$,
      $H(\boldsymbol{y}_p, \lambda_p)$ and $T$ then
        // Accept
        $\lambda_i = \lambda_p$;
    else
        // Reject
        $\lambda_i = \lambda_{i-1}$;
    end
    Decrease $T$ according to the cooling schedule;
end
output: $\lambda_{n_{max}}$;
```

Algorithm 11: Algorithm to optimize the parameters of mixture PDFs with SA. See section 3.2.2.1 for more details about SA.

Remark 91. If it is wanted to use a sampling algorithm [18] in the space of weights (or any other constrained space) instead of SA, then it is necessary to consider the observations in section 3.1.6. Using a sampling algorithm can be interesting if extracting information from the chain is required [19].

Example 8. This is an example of the design optimization problem stated in this section. This example is interesting only from a theoretical point of view. The role of the surrogate model is taken by the following deterministic function,

$$y(x_1, x_2) = 2\frac{-\exp(-\frac{-v_1^T v_1}{\sigma^2}) - \exp(-\frac{-v_2^T v_2}{\sigma^2})}{s}, \tag{3.31}$$

where

$$\begin{aligned} v_1 &= (x_1 - \frac{1}{3}, x_2 - \frac{2}{3})^T, \\ v_2 &= (x_1 - \frac{2}{3}, x_2 - \frac{1}{3})^T, \\ \sigma^2 &= 0.05, \\ s &= 1 + \exp(\frac{-w^T w}{\sigma^2}), \\ w &= (\frac{1}{3}, \frac{1}{3})^T, \end{aligned} \tag{3.32}$$

which has only two inputs and one output. Consider a displaced log-normal distribution with parameters $\mu = -1.025$ and $\sigma = 0.7644$ and a displacement of -2.1, as a target PDF. It is required to find a PDF in the input space of the surrogate model such that the PDF obtained in the output space, after propagating uncertainties, is as close as possible to the target density. Figure 3.6 shows the surrogate model and the target PDF. The low dimensionality was chosen to allow its visualization. However, it is a difficult scenario because most of the regions in the input space lead to the tail of the target PDF. The parametric family $\mathcal{F}$ was a mixture of 3 normal distributions. Maximum Mean Discrepancy (see (3.28)) was used as the objective function. SA was used to optimize the parameters of the mixture. Figure 3.7 shows the results. Notice that the random initial point led to a very poor approximation. From this very poor initial point, SA moved to a region of $S(\lambda)$ that led to the desired performance.

[18] E.g., M-H or any other MCMC algorithm.

[19] The chain in M-H (or other MCMC sampling algorithms) approximates the density. The chain in SA gives no information.

Eventually, after the optimization process, the last point given by SA approximates the target PDF satisfactorily.

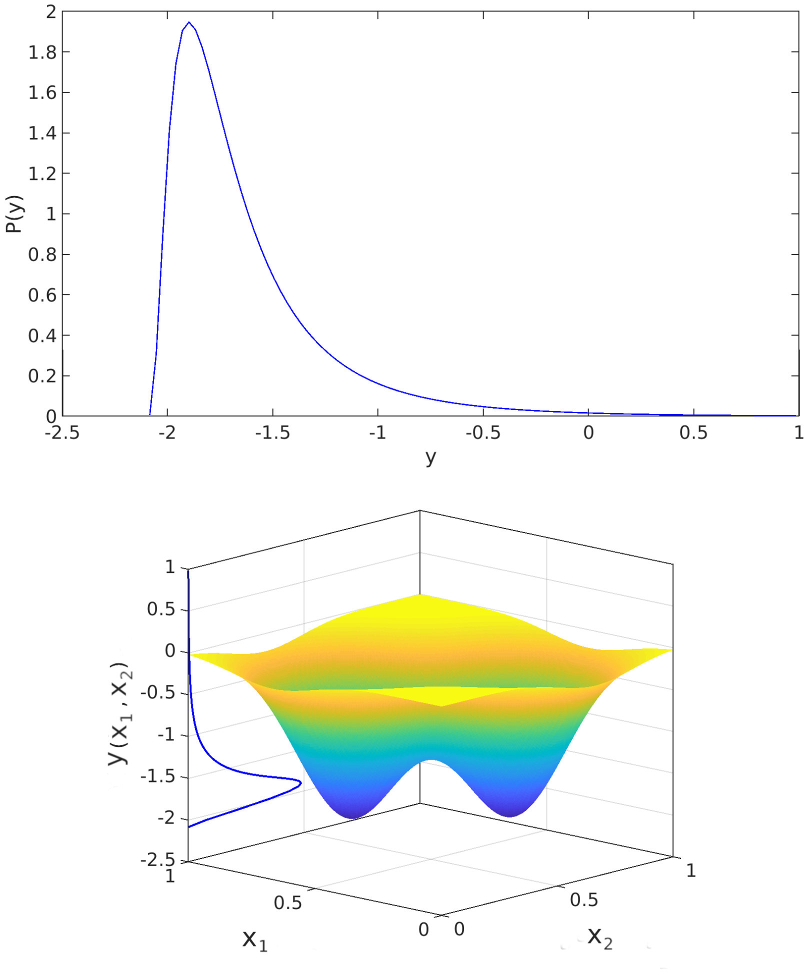


FIGURE 3.6: Surrogate model and target PDF of example 8. *Blue line*: Target PDF. *Surface*: Surrogate model and target PDF.

Example 8 shows that the algorithm 11 succeeded in a difficult but low dimensional scenario. The following example 9 aims to test algorithm 11 for PDF approximation on a bigger scale. The complexity of each iteration should not be affected by the dimensionality of the input space. Notice that the only step that is affected by the dimensionality in each iteration is the sampling process. The sampling complexity grows linearly for normal distributions. Sampling from an N-dimensional normal distribution is equivalent to sampling from N unidimensional distributions (see corollary 3.1.1).

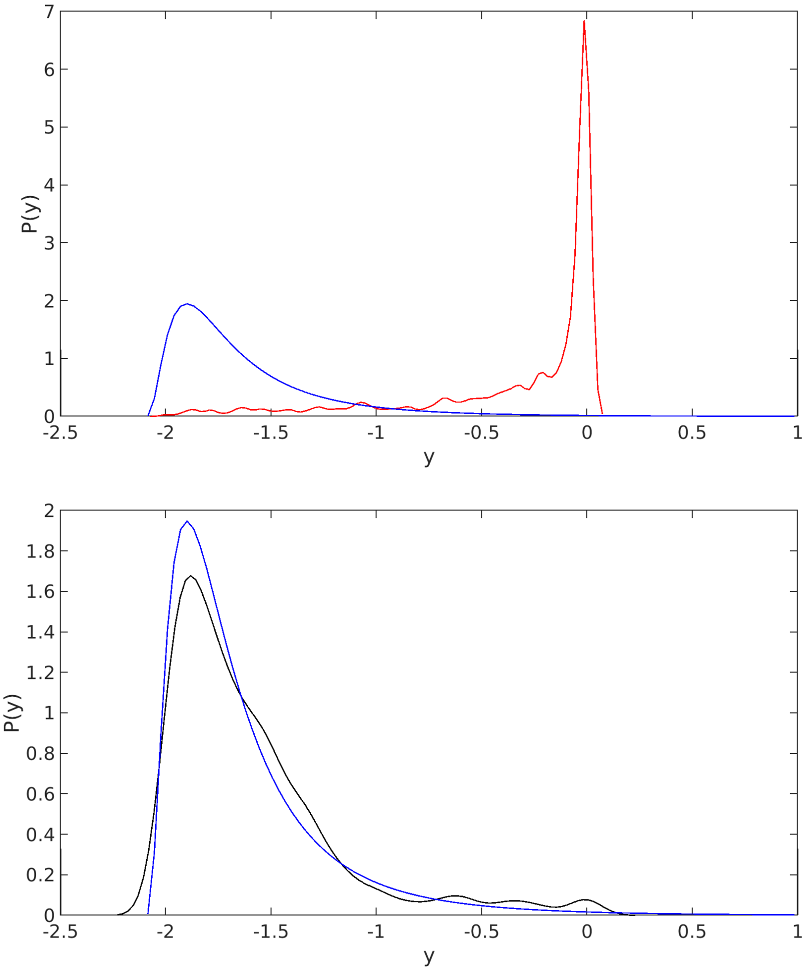


FIGURE 3.7: Results of example 8. *Dark blue line*: Target PDF. *Red line*: PDF from the random intial point in the SA optimization process. *Black line*: PDF from the last point in the SA optimization process.

Example 9 will use mixtures of normal distributions in the same fashion as example 8. Theoretically, each iteration should have similar complexity since the sampling grows linearly, and therefore, it is asymptotically irrelevant. Computing the MMD will be the asymptotically relevant operation with regards to computational complexity. Notice that computing the MMD is not affected by the input space dimensionality.

Although each iteration complexity will not be affected by the input space dimensionality, it may be that the algorithm does not succeed in a high dimensional scenario. Exploring a high dimensional space may need many more iterations before getting a good approximation of the target PDF. The following example uses a 10-dimensional input space and proves that algorithm 11 can explore and get good results in high-dimensional spaces. Notice that $S(\lambda)$ is a 59-dimensional space if the covariance matrices considered are of the form $\sigma^2 I_{10\times 10}$ [20]. 10^4 iterations were used in example 8. However, they were not sufficient in a higher-dimensional space. There were needed 10^6 iterations to get the results shown in the example 9.

Example 9. This example is analogous to the example 8. However, the surrogate's input space will be a 10-dimensional space. Using the same function

$$\hat{y}(x_1, x_2) = 2\frac{-\exp(-\frac{-v_1^T v_1}{\sigma^2}) - \exp(-\frac{-v_2^T v_2}{\sigma^2})}{s},$$

used in example 8 (see (3.31) and (3.32)), the surrogate model, in this case, will be

$$y(x_1, ..., x_{10}) = \sum_{i=1}^{5} \hat{y}(x_{2i-1}, x_{2i}).$$

A normal distribution $\mathcal{N}(-5.5, 1)$ is considered as a target PDF. A mixture of 5 normal distributions is considered as the parametric family $\mathcal{F}$. Maximum Mean Discrepancy (see (3.28)) was used as the objective function. SA was used to optimize the parameters of the mixture. Figure 3.8 shows the target PDF approximation from the random initial point to the last point given by SA. Notice that the

[20] 4 parameters in the simplex mapping for the weights, 50 parameters for the 5 means of a 10-dimensional space and 5 parameters associated to the covariance matrices of the form $\sigma^2 I_{10\times 10}$

algorithm has been able to approximate the target PDF satisfactorily with conclusions analogous to example 8.

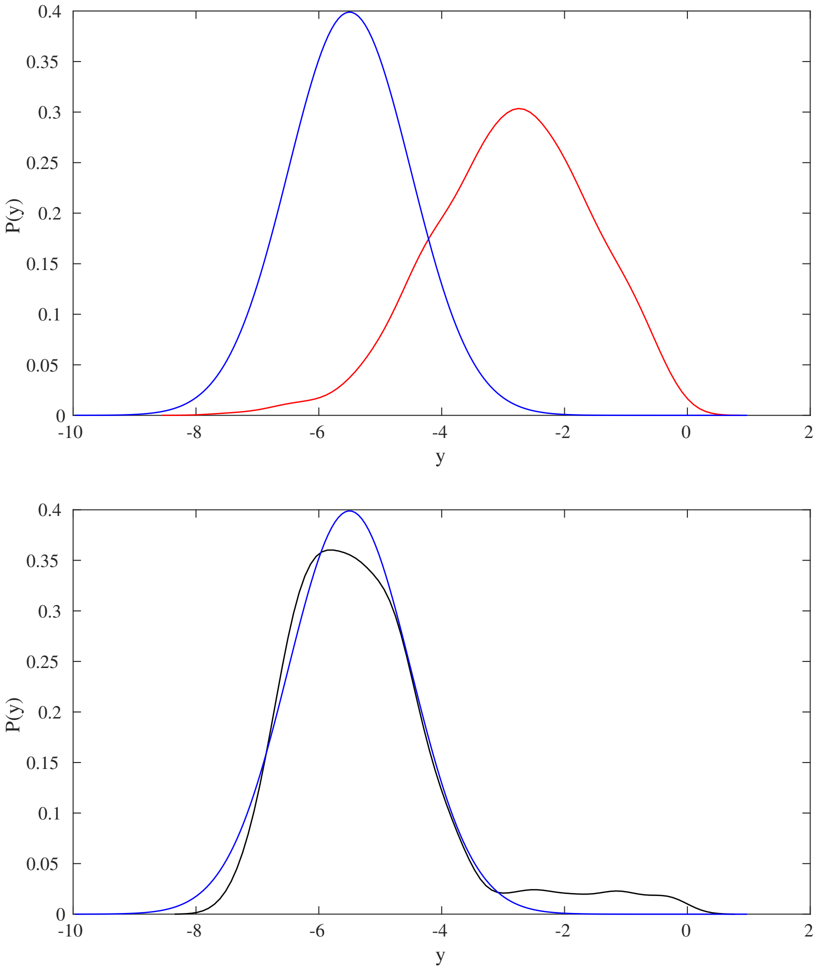


FIGURE 3.8: Results of example 9. *Dark blue line*: Target PDF. *Red line*: PDF from the random intial point in the SA optimization process. *Black line*: PDF from the last point in the SA optimization process.

Example 8 and example 9 show the potential of algorithm 11. Example 8 proposes a challenging scenario. Most of the regions in the input space lead to the tail of the target PDF. The algorithm has been able to move from a very deficient random initial point to a point in the parameter space $S(\lambda)$ that approximates the target PDF with a small error (see figure 3.7). Example 8 uses a surrogate model with a two-dimensional input space. However, example 9 proves that the algorithm can also give satisfactory results in high-dimensional spaces (see figure 3.8).

3.2.5 Gradient-based optimization

Remember that the aim is to optimize the parameters λ of the parametric family $\mathcal{F}$. It has been used the terminology "objective function" to refer to $H(\boldsymbol{y}, \lambda)$ during this chapter because $\boldsymbol{y}$ depends on λ in the optimization process (see remark 80). Thus, to be rigorous, a function $G(\lambda) = H(\boldsymbol{y}_\lambda, \lambda)$ should be defined.

Let $\mathcal{F}$ be the parametric family of N-dimensional normal distributions. Let

$$\lambda \in \mathbb{R}^{\frac{N^2+N}{2}} \times (\mathbb{R}^+ \setminus \{0\})^N$$

be the parameter of this family. Let μ_λ and L_λ be the mean and the lower triangular matrix associated to λ [21] (see (3.29)). Let $\boldsymbol{x}_s = \{x_{s,1} \dots x_{s,n}\}$ be a set of samples from the standard N-dimensional normal distribution, $\mathcal{N}(0, I_{N\times N})$. Using the same strategy as algorithm 9, the idea is to generate samples only once, before the optimization process starts.

In this Chapter, it was assumed that information about the surrogate model is not available, and therefore the regularity of G is not guaranteed. However, consider a surrogate model $y(x)$ such that,

$$\begin{aligned} y_{x_{s,i}} : \mathbb{R}^{\frac{N^2+N}{2}} \times (\mathbb{R}^+ \setminus \{0\})^N &\to \mathcal{Y} \\ \lambda &\mapsto y(\mu_\lambda + L_\lambda x_{s,i}) \end{aligned} \tag{3.33}$$

is differentiable for all $1 \leq i \leq n$.

Example 10. For example, let $\mathcal{Y} = \mathbb{R}$ and $\boldsymbol{y}(\boldsymbol{x}_*) = (y_{\boldsymbol{x}_*,1}, \dots, y_{\boldsymbol{x}_*,n})$ be the mean of a GP surrogate (see (2.50)) for a set of test points $\boldsymbol{x}_* = \{x_{*1}, \dots, x_{*n}\}$. The notation $\boldsymbol{x}_*$ from Chapter 2 was retrieved to avoid confusion with the training data notation $\boldsymbol{x}$ in the GP. In this case, note that $\lambda \mapsto \boldsymbol{y}(\mu_\lambda + L_\lambda x_{s,i})$ is differentiable if the kernel is differentiable.

Consider the function $\lambda \mapsto \boldsymbol{y}_\lambda = (y_{x_{s,1}}(\lambda), \dots, y_{x_{s,n}}(\lambda))$. I.e., $\boldsymbol{y}_\lambda$ are the fix set of points $\{x_{s,1}, \dots, x_{s,n}\}$ transformed to be samples according to λ and propagated through the surrogate model. Notice that the transformation and the surrogate evaluation are differentiable with respect to λ. Therefore,

$$G(\lambda) = H(\boldsymbol{y}_\lambda, \lambda) \text{ differentiable, if and only if, } H(\boldsymbol{y}, \lambda) \text{ differentiable.}$$

[21] The matrix $\Sigma_\lambda = L_\lambda L_\lambda^T$ is the covariance matrix associated to λ.

Using this approach and a differentiable function H is possible to use gradient-based optimization algorithms which are generally more computationally efficient than the metaheuristics presented in this work (section 3.2.2). However, using these algorithms requires special care to avoid getting trap in a local minimum.

A design optimization problem was stated in this chapter. It consisted of optimizing the parameters of a PDF in the input space of a surrogate model in order to satisfy prescribed performance in the outputs. Section 3.2 introduced a novel framework to tackle this inverse problem. Section 3.1.1 was dedicated to the explanation of different methods to sample from a PDF, which is a fundamental process for this framework. Special care was taken to define a framework as general as possible. It can be applied to many scenarios. First, it does not make assumptions on the model. Secondly, it was defined such that different optimization methods can be used (SA is proposed as a general solution for the optimization problem). Finally, the objective function can be tuned in order to satisfy different requirements.

The following chapter gives guidelines to apply this framework to a specific design problem.

Chapter 4

Perspectives on early-stage aircraft wing design

The work presented in this thesis is part of a project in collaboration with Airbus. They provided a dataset with information about arly-stage aircraft wing design under a non-disclosure agreement between Airbus and Cardiff University for research usage.

The first project's challenge is to build a surrogate model from this dataset. Chapter 2 provides information about this topic. The GP regression presented there consider unidimensional outputs. It will be seen that there is more than one quantity of interest for a given choice of design parameters. Therefore, the learning and GP theory given in this thesis should be broadened to consider multiple outputs. Refer, e.g., to Micchelli and Pontil (2005); Carmeli, De Vito, and Toigo (2006); Bilionis et al. (2013); or Alvarez, Rosasco, and Lawrence (2012) regarding this matter. There were conducted some experiments training different GP for each quantity of interest [1]. For this dataset, a study of the GPs log marginal likelihoods (see section 2.2.6 and remark 51) shows that the squared exponential kernel and the Matern kernel with parameter 5/2 (both with a separate length scale per predictor) give the better results after optimizing the kernel parameters.

The second challenge, when the surrogate model is already built, is to identify probability distributions in the wing design parameters (the input variables of the surrogate model) that lead to a prescribed performance (the output quantities of the surrogate model).

The goals of this section are:

[1] Note that no correlation between outputs is assumed if this methodology is adopted. However, it is known that there are strong correlations between the different quantities of interest in this dataset.

1. Explaining the structure of this dataset and the optimization problem associated.
2. Giving guidelines to use the theory and the novel techniques introduced in Chapter 3 for tackling the design optimization problem stated above.

Future researchers of this project will benefit from the following analysis of the dataset and the ideas given in this section and in Chapter 3.

4.1 Parametrization

Eight parameters are considered corresponding to epistemic uncertainty in wing stiffness. Four parameters for wing bending stiffness (EI) and four for wing torsinal stiffness (GJ) in four sections along the wing. These will be referred as "uncertainty parameters".

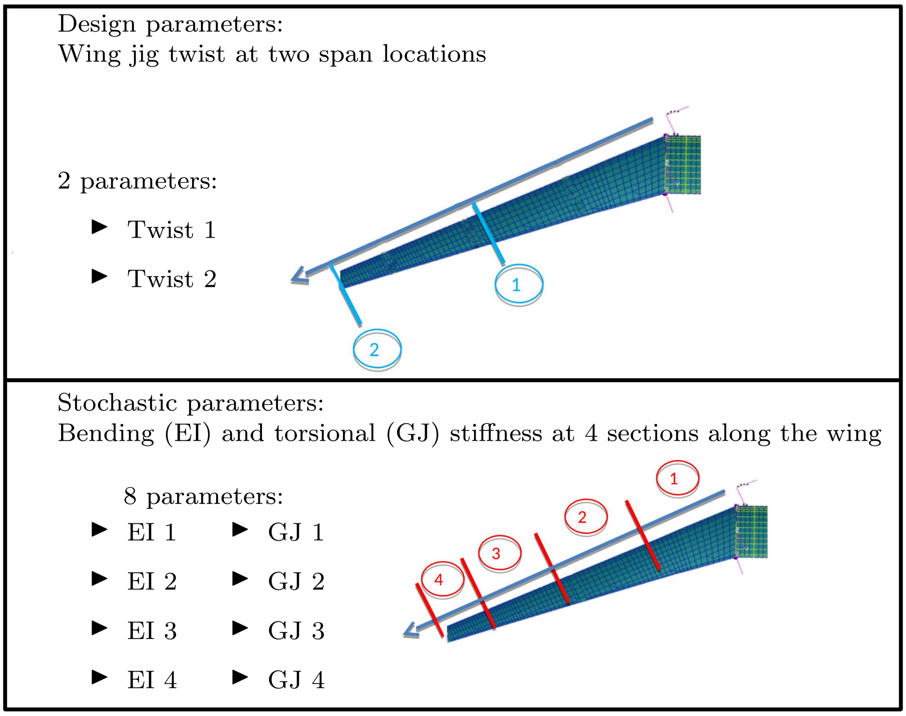
Two parameters are considered corresponding to the wing jig twist at two span locations of the wing. These will be referred as "optimization" parameters.

Figure 4.1 illustrates the location of those parameters graphically.

4.1.1 Design Of Experiments

The ten parameters considered in this dataset form a ten dimensional space $\mathcal{X}$. For each point in this parameters space, it is possible to measure or predict the aerodynamic performance as well as the external loads, Shear, Moment and Torque (SMT) along the wing. This paradigm determines a function in which $x \in \mathcal{X}$ are the inputs and the aerodynamic performance and the loads (SMT) are the outputs. The prediction of the outputs are computationally expensive. Therefore, evaluations of this function are not available in a reasonable time to run an optimization algorithm. This dataset provides outputs for only some points in the parameters space. Those points in which the predictions were conducted are called Design Of Experiments (DOE) points.

The problem of obtaining evaluations of this function using only a finite (and usually few) number of previous computed evaluations is explained in Chapter 2.



Design parameters:
Wing jig twist at two span locations

2 parameters:

- Twist 1
- Twist 2

Stochastic parameters:
Bending (EI) and torsional (GJ) stiffness at 4 sections along the wing

8 parameters:

- EI 1
- EI 2
- EI 3
- EI 4
- GJ 1
- GJ 2
- GJ 3
- GJ 4

FIGURE 4.1: Parameters and its spanwise location along the wing. Uncertainty parameters in red. Optimisation parameters in blue.

It is not trivial the problem of selecting the set of DOE points. The question of what set will provide more information about the underlying function is an open field of research. Optimized Latin Hypercube Sampling (OLHS) was used to select the DOE points aiming to a good space filling of the parameters space.

4.1.1.1 Optimized Latin Hypercube Sampling

Different techniques can be used to determine the DOE points. Santner, Williams, and Notz (2003) describes the theory of this subject and explain some of these techniques. An approach that can be considered is to find the best coverage of the parameters/input space $\mathcal{X}$. The algorithms that pursue this goal are called Space Filling Designs (SFD).

OLHS is an SFD algorithm. It was the technique used to select the DOE points of this dataset. Although the study of DOE algorithms is not the purpose of this work, information about OLHS can be found in, e.g., Santner, Williams, and Notz (2003); Damblin, Couplet, and Iooss (2013); Li et al. (2017); or Xiong et al. (2009)

Three OLHS were conducted. One generated 50 DOE points for the aerodynamic performance simulations. Two OLHS generated 200 DOE points for the loads data, 150 the first one and 50 the second one.

The benefit of having two coverages of $\mathcal{X}$ is that one can be used for training a surrogate and the other for validation. Only one OLHS was conducted for the aero data due to the computational cost of these simulations.

An additional DOE point was included in both, aero and loads data. This is the baseline, unmodified aircraft. In addition, two more DOE points were included in the aero dataset. This are the cases with zero stiffness changes and maximum twist-on and twist-off respectively.

Therefore, the loads and aero datasets include 201 DOE points and 53 DOE points respectively.

4.1.2 Bending and torsional stiffness

The wing stiffness (both EI as well as GJ) was varied across four spanwise sections of the wing (see figure 4.1) according to a certain percent change relative to the baseline. A smaller percent change is allowed at the root and a larger change is allowed at the tip (see figures 4.2 and 4.3).

This percent change has been normalized by Airbus for external usage to give a relative variation of ± 1 at the wing tip and the relative reductions at the other spanwise positions (see figures 4.2 and 4.3). The wing spanwise locations have also been normalized being 0 the root and 1 the tip. The innermost section of stiffness change is from 0 to 0.2419, the middle from 0.2649 to 0.7075, the outer from 0.7282 to 0.8956, and the winglet from 0.9217 to 1 (see figures 4.1, 4.2 and 4.3).

4.1.3 Wing twist

The wing jig twist describes the shape of the wing while it is being made in its jig (i.e. with no loads applied). The wing has some bending and twisting along its span induced by the jig in this form. During the flight, the aerodynamic forces deform the wing. The aim of the jig bending and twisting is to have the desired shape after this deformation in flight.

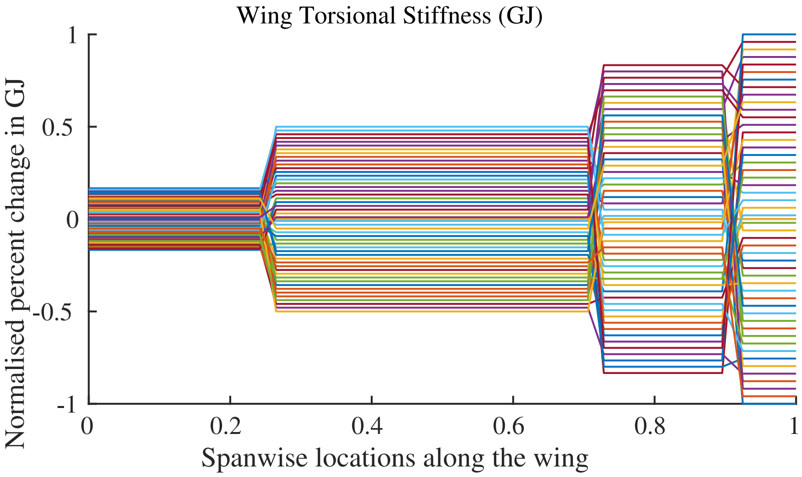


FIGURE 4.2: Percentage change in torsional stiffness vs spanwise locations along the wing. The data has been normalized by Airbus. Each line represents one of the 53 DOE points of the aero dataset. Notice that a smaller percent change is allowed at the root and a larger change is allowed at the tip.

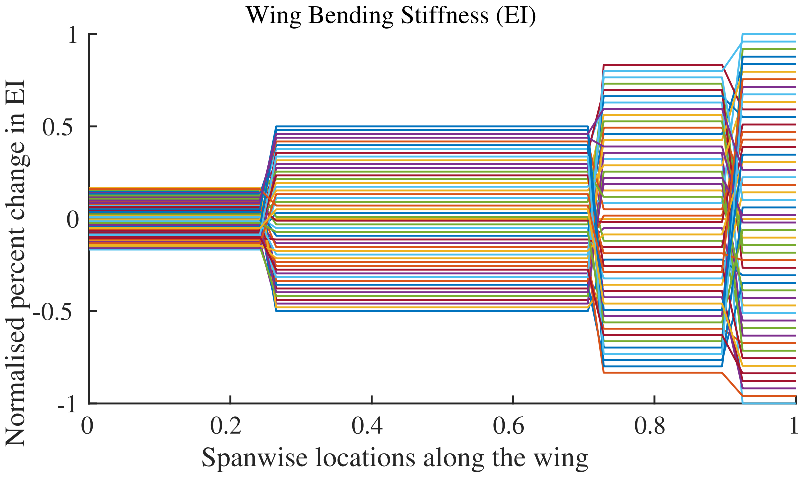


FIGURE 4.3: Percentage change in bending stiffness vs spanwise locations along the wing. The data has been normalized by Airbus. Each line represents one of the 53 DOE points of the aero dataset. Notice that a smaller percent change is allowed at the root and a larger change is allowed at the tip.

The main purpose of this dataset is to investigate variations on the wing's jig twist, or the local angles of incidence of the unloaded wing. The wing jig twist was varied at two stations on the wing. The locations along the wing and the twist variations have been normalized by Airbus for external usage. These two stations are located at 0.5830 (approximately mid-span) and 1 (tip) (see figure 4.4). The twist at this two points are the two optimization parameters (see figure 4.1). The twist variations take values between -1 and $+1$. -1 represents the maximum negative variation on the twist with respect to unloaded shape of the wing. $+1$ represents the maximum positive variation.

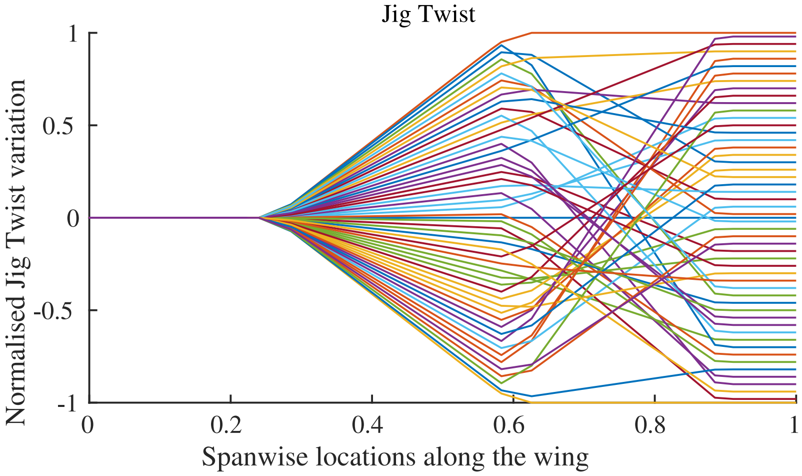


FIGURE 4.4: Jig twist variation vs spanwise locations along the wing. The data has been normalized by Airbus. Each line represents one of the 53 DOE points of the aero dataset.

4.2 Simulations

4.2.1 Aerodynamic performance data

The aero performance data was generated using coupled Computational Fluid Dynamics (CFD) and Computational Structural Mechanics (CSM) to capture the impact of stiffness changes on the efficiency of the aircraft in cruise. First, CFD calculates aerodynamic loads and then CSM estimates the resulting deformation by iteration, changing the shape until convergence is reached.

The data consists of a polar trimmed at a variety of different Coefficient of lift (Cl) values. Cl 1 is the design cruise point and all other

Cl values are relative to that design point. The Lift over Drag ratios (L/D) are relative to the baseline aircraft and normalized. The figure 4.5 shows L/D as a function of Cl for each DOE point in the aero data.

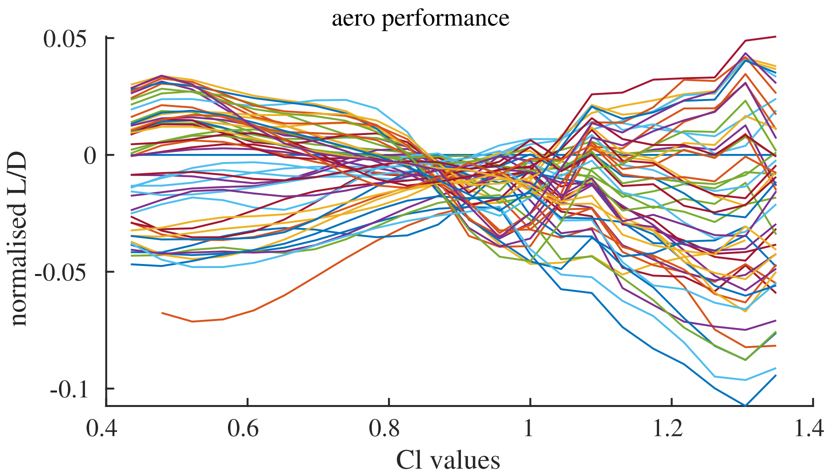


FIGURE 4.5: Lift over Drag ratio vs Cofficient of Lift. The data has been normalized by Airbus. Each line represents one of the 53 DOE points of the aero dataset. Cl 1 is the design cruise point and all other Cl values are relative to that design point. The Lift over Drag ratios (L/D) are relative to the baseline aircraft.

4.2.2 Loads data

The loads dataset was generated using Airbus' certification standard simulation tools for a range of typical gust and manoeuver cases. They are a variety of aircraft mass cases as well as different Mach and altitude points. They include discrete gust, continuous turbulence, as well as a number of different manoeuvers.

For each of these cases the dataset provides the envelope shear, moment and torque values at different locations on the wing. Although these may not be the highest stresses in the wing, they are sufficient for the robust optimization purpose.

4.2.2.1 Gust cases

The loads simulations contain 23 gust cases with 4 different gust types, 17 mass cases, 6 Mach numbers, 2 thrust settings, and 8 altitudes. This includes both continuous turbulence as well as discrete gusts.

4.2.2.2 Manoeuver cases

The loads simulations contain 21 different steady manoeuver cases across the flight envelope in a range of different Mach and altitudes.

4.2.3 Description of Interesting Quantities (IQ)

A brief description of the IQ values provided in this dataset.

4.2.3.1 Load types (IQ type)

- **Shear**: Vertical shear force.
- **Moment**: Bending moment along the aircraft longitudinal (fuselage) axis.
- **Torque**: Torque along the lateral (spanwise) axis.
- **Cl**: Coefficient of lift value.

4.2.3.2 Locations (IQ component)

- Spanwise stations along the **right wing**, starting at the root and moving towards the tip.
- Spanwise stations along the **right winglet**, starting at the tip of the wing and moving further outboard to the tip of the winglet.
- Spanwise stations along the right **horizontal tail**, starting at the root and moving towards the tip.

4.3 Design optimization guidelines

This section will assume that a surrogate model $y(x)$ has been built from the dataset [2]. Consider the parameters explained in sections 4.1, 4.1.2 and 4.1.3 to be the variables in the input space of the surrogate model $x \in \mathcal{X} \subset \mathbb{R}^n$. Consider the quantities of interest in the different cases (see sections 4.2), or a function of them, to be the variables of the output space $y \in \mathcal{Y} \subset \mathbb{R}^m$.

The aim is to use the theory explained in Chapter 3 to find a probability distribution in the input space of the surrogate model such

[2] See the introduction of this Chapter.

that the distribution observed in the output satisfies a prescribed performance.

There are three fundamental choices that must be made to apply the theory introduced in section 3.2:

1. The parametric family $\mathcal{F}$.
2. The objective function
$$H(\boldsymbol{y}, \lambda) = H_{\boldsymbol{y}}(\boldsymbol{y}) + \gamma H_\lambda(\lambda). \tag{4.1}$$
3. The algorithm to optimize the parameters of $\mathcal{F}$.

It is proposed to use multivariate normal distributions as the parametric family $\mathcal{F}$ due to the reasons that are given in section 3.2.4. In addition, a normal distribution provides information easy to interpret by the designers. [3]

The optimization algorithm proposed in this thesis is SA (section 3.2.2.1) and it is also the recommendation for this project. However, other possibilities can be explored (see section 3.2.2).

The major challenge is to design the objective function H. It must be designed according to the desired performance. Functions H_λ, $H_{\boldsymbol{y}}$ and constant γ that well define the target performance must be selected. Future researchers in this project can consider the following suggestions:

1. Flexibility is desired in early-stage design. Therefore, the function H_λ should be designed according to this regard. For example, extracting information from λ about the variances and covariances of the distributions f_λ. Although jig twist and bending and torsional stiffness are considered all input variables, Airbus makes an important distinction between them. Jig twist parameters are considered the "true" design parameters while bending and torsional stiffness are considered "uncertainty parameters". An approach could be to set a fixed uniform distribution on the uncertainty parameters. However, this approach is too extreme. It will represent no control at all in those parameters. Let $\mathcal{F}$ be a family of multivariate normal distribution and

[3] A normal distribution gives a target (the mean) and the degree of accuracy needed, given by the covariance matrix, which provides with information about the variances and correlations between variables.

$\lambda = (\mu, \Sigma)$. Let Σ_d be the submatrix of the covariance matrix Σ corresponding to the design parameters. Let Σ_u be the submatrix corresponding to the uncertainty parameters. A more reasonble approach is to consider $H_\lambda = \frac{b_d}{v_d} + \frac{b_u}{v_u}$ where v_d is the lowest eigenvalue of Σ_d, v_u is the lowest eigenvalue of Σ_u and $b_u \gg b_d$. This will push the algorithm to give distributions with more flexibility in the uncertainty parameters. Variations with the determinant or other values extracted from the covariance matrix could be used.

2. Airbus already has functions $q(y)$ that measure the performance of a given output. They use the terminology penalty functions. The higher the value, the worse the performance. It is suggested to use the expectation of these functions,

$$\mathcal{H}_{\boldsymbol{y}} = \frac{1}{n} \sum_{i=1}^{n} q(y_i).$$

3. If the target performance is given by a target PDF, see section 3.2.3.

4. If the target performance is given by a desired region of the outputs, consider

$$\mathcal{H}_{\boldsymbol{y}} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_A(y_i),$$

where A is the desired region, or

$$\mathcal{H}_{\boldsymbol{y}} = \frac{1}{n} \sum_{i=1}^{n} d_A(y_i),$$

where $d(y) = 0$ for $y \in A$ and a distance o penalty if $y \notin A$.

5. If the target performance is given by a target value y_t, consider

$$\mathcal{H}_{\boldsymbol{y}} = \frac{1}{n} \sum_{i=1}^{n} \|y_i - y_t\|_W, \ \|v\|_W = \sqrt{v^T W v}, \ W \text{ positive-definite.}$$

For example, consider the data normalized and two quantities of interest: $y \in \mathbb{R}^2$. If it is wanted to have similar errors in both quantities [4], then the eigenvectors of W could be $v_1 = \frac{1}{\sqrt{2}}(1,1)$

[4] Optimizing one quantity more than the other is penalized.

with eigenvalue c_1 and the orthogonal $v_2 = \frac{1}{\sqrt{2}}(-1,1)$ with eigenvalue c_2, with $0 < c_1 < c_2$,

$$W = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}^T .$$

The same strategy could be used for other purposes. For instance, in the same situation but considering the first quantity more important than the second one,

$$\begin{gathered} v_1 = (a,1), \\ v_2 = (-1,a), \\ a > 1, \\ 0 < c_1 < c_2, \\ W = \tfrac{1}{1+a^2} \begin{pmatrix} a & -1 \\ 1 & a \end{pmatrix} \begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix} \begin{pmatrix} a & -1 \\ 1 & a \end{pmatrix}^T . \end{gathered}$$

Figure 4.6 illustrates examples for some choices of parameters a, c_1 and c_2.

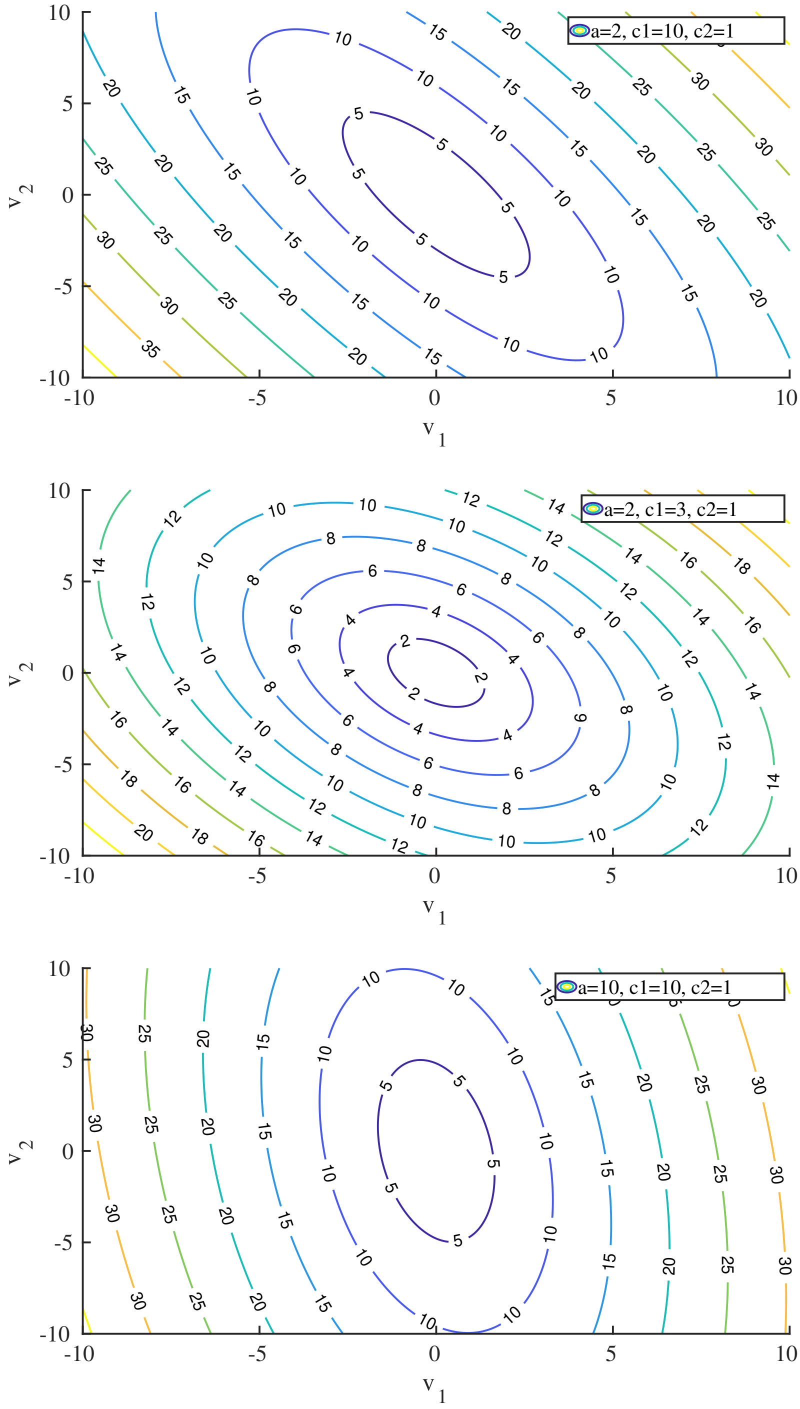


FIGURE 4.6: Examples of norms $v = (v_1, v_2)$, $\|v\|_W = \sqrt{v^T W v}$, W positive-definite, where, $v_1 = (a, 1)$, $v_2 = (-1, a)$, $a > 1$, $0 < c_1 < c_2$, $W = \frac{1}{1+a^2}\begin{pmatrix} a & -1 \\ 1 & a \end{pmatrix}\begin{pmatrix} c_1 & 0 \\ 0 & c_2 \end{pmatrix}\begin{pmatrix} a & -1 \\ 1 & a \end{pmatrix}^T$.

Chapter 5

Conclusions

The aim of this thesis is to provide information about surrogate modeling from the learning theory point of view, and to propose a useful method for identifying distributions on design parameters which satisfy target performance metrics.

Regression is a vast field that can be tackled with many different approaches. This thesis explains the key concepts to understand learning theory, the regression algorithms based in RKHS spaces and, particularly, GP regression. The mathematical steps to build a GP were covered in detail. Examples and figures were given for a better understanding.

Regression is an active research topic that has received special attention since the irruption of ML. Other tools in ML could be adopted for surrogate modeling from input-output data such as Neural Networks (NN) and Deep Learning, especially for large datasets. In addition, a surrogate model built from functions whose derivatives are easy to compute (e.g., NN) could be a suitable candidate for using the ideas introduced in section 3.2.5.

The novelties of this work are in Chapter 3. A novel framework for design optimization is proposed. Chapter 4 analyzes a dataset provided by Airbus and gives clear guidelines on the use of this framework with it. The goal is to identify probability distribution in the input space of a surrogate model such that a desired performance is observed in the output. This inverse problem (stated at the beginning of section 3.2) is transformed into an optimization problem of PDF parameters. Although this optimization problem is well defined using the expectation of a function as the objective function, it becomes inconsistent when the expectation is approximated by samples [1]. Two methods to eradicate uncertainties in the objective function were proposed:

1. Transforming the same set of samples during the optimization process from a standard normal distribution to any other normal distribution by a linear transformation (in the case of multivariate normal PDFs).

2. Resetting the pseudorandom number generator seed to a fix value before any sampling step (in the case of an arbitrary PDF).

General guidelines for applying this framework were provided. However, its success in each particular case depends on 3 fundamental points:

1. Designing the objective function H is crucial.Examples 6 and 7 illustrate objective functions for different scenarios. Section 3.2.3 proposes MMD for the particular case of PDF approximation. However, it would be relevant to design functions H for more purposes and study its behavior.

2. The optimization process is key. The use of SA as a general tool, which would be suitable in many situations, was proposed. The general theory of SA was explained. However, a deeper study of the different improvements of the basic SA and the tunning of the SA parameters (such as the cooling schedule) would be interesting to make it as efficient as possible for this optimization problem. Other optimization algorithms could be explored and compared with SA.

3. Methods to accelerate the exploration of the parameters space $S(\lambda)$ may be necessary for high dimensions. The possibility of an initial search with a small number of samples and increasing this number during the process could be considered and studied. [2]

Furthermore, the possibility of optimizing PDF parameters with fast methods based on the gradient of the objective function could be more computationally efficient, allowing the optimization process to

[1] There are uncertainties in the evaluations of the objective function due to the uncertainties in the sampling process.

[2] In a similar fashion to the cooling schedule.

work in higher dimensions. The key point for this approach was the idea of transforming the problem of generating samples from a process involving uncertainties to a deterministic process given by a linear function (see (3.33)). This work focused on the assumption that the surrogate's model may not be differentiable, and therefore this idea was not completely developed. However, this line of research could lead to interesting results if differentiability is assumed.

References

Alvarez, M. A., Rosasco, L., and Lawrence, N. D. (Apr. 2012). "Kernels for Vector-Valued Functions: A Review". en. In: *arXiv:1106.6251 [cs, math, stat]*. arXiv: `1106.6251 [cs, math, stat]`.

Anderson, T. W. (2003). *An Introduction to Multivariate Statistical Analysis*. English. Hoboken N.J: Wiley-Interscience. ISBN: 0-471-36091-0 978-0-471-36091-9.

Antunes, F. et al. (Oct. 2017). "A Review of Heteroscedasticity Treatment with Gaussian Processes and Quantile Regression Meta-Models". In: *Springer Geography*, pp. 141–160. ISBN: 978-3-319-40900-9.

Aronszajn, N. (Mar. 1950). "Theory of Reproducing Kernels". en. In: *Transactions of the American Mathematical Society* 68.3, pp. 337–337. ISSN: 0002-9947. DOI: `10.1090/S0002-9947-1950-0051437-7`.

Bell, J. R. (July 1968). "Algorithm 334: Normal Random Deviates". In: *Communications of the ACM* 11.7, p. 498. ISSN: 0001-0782. DOI: `10.1145/363397.363547`.

Betancourt, M. (July 2018). "A Conceptual Introduction to Hamiltonian Monte Carlo". en. In: *arXiv:1701.02434 [stat]*. arXiv: `1701.02434 [stat]`.

Bilionis, I. et al. (May 2013). "Multi-Output Separable Gaussian Process: Towards an Efficient, Fully Bayesian Paradigm for Uncertainty Quantification". In: *Journal of Computational Physics* 241, pp. 212–239. DOI: `10.1016/j.jcp.2013.01.011`.

Blum, C. and Roli, A. (Jan. 2001). "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison". In: *ACM Computing Surveys* 35, pp. 268–308. DOI: `10.1145/937503.937505`.

Bobrowski, A. (2005). *Functional Analysis for Probability and Stochastic Processes : An Introduction*. English. Cambridge: Cambridge Univ. Press. ISBN: 0-521-53937-4 0-521-83166-0 978-0-521-83166-6 978-0-521-53937-1.

Box, G. E. P. and Muller, M. E. (June 1958). "A Note on the Generation of Random Normal Deviates". en. In: *Ann. Math. Statist.* 29.2, pp. 610–611. ISSN: 0003-4851. DOI: `10.1214/aoms/1177706645`.

Brooks, S. (2011). *Handbook for Markov Chain Monte Carlo*. English. Boca Raton: Taylor & Francis. ISBN: 978-1-4200-7941-8 1-4200-7941-7.

Carmeli, C., De Vito, E., and Toigo, A. (Oct. 2006). "Vector Valued Reproducing Kernel Hilbert Spaces of Integrable Functions and Mercer Theorem". en. In: *Analysis and Applications* 04.04, pp. 377–408. ISSN: 0219-5305, 1793-6861. DOI: 10.1142/S0219530506000838.

Casella, G. and George, E. I. (1992). "Explaining the Gibbs Sampler". In: *The American Statistician* 46.3, pp. 167–174. ISSN: 00031305.

Chib, S. and Greenberg, E. (Nov. 1995). "Understanding the Metropolis-Hastings Algorithm". In: *American Statistician* 49, pp. 327–335. DOI: 10.1080/00031305.1995.10476177.

Cucker, F. and Smale, S. (Oct. 2001). "On the Mathematical Foundations of Learning". en. In: *Bulletin of the American Mathematical Society* 39.01, pp. 1–50. ISSN: 0273-0979. DOI: 10.1090/S0273-0979-01-00923-5.

Cucker, F. and Zhou, D. (2007). *Learning Theory : An Approximation Theory Viewpoint*. en. Cambridge University Press. ISBN: 978-0-511-27551-7.

Damblin, G., Couplet, M., and Iooss, B. (July 2013). "Numerical Studies of Space Filling Designs: Optimization of Latin Hypercube Samples and Subprojection Properties". en. In: *arXiv:1307.6835 [math, stat]*. arXiv: 1307.6835 [math, stat].

Debnath, L. and Mikusinski, P. (1990). *Introduction to Hilbert Spaces with Applications*. English. Boston, Mass.: Boston, Mass. : Academic Press, Harcourt Brace Janovich. ISBN: 0-12-208435-7 978-0-12-208435-5.

Dekking, M., ed. (2005). *A Modern Introduction to Probability and Statistics: Understanding Why and How*. en. Springer Texts in Statistics. London: Springer. ISBN: 978-1-85233-896-1.

Demmel, J. W. (1997). *Applied Numerical Linear Algebra*. English. Philadelphia: SIAM, Soc. for Industrial and Applied Mathematics. ISBN: 0-89871-389-7 978-0-89871-389-3.

Dueck, G. and Scheuer, T. (1990). "Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing". In: *Journal of Computational Physics* 90.1, pp. 161–175. ISSN: 0021-9991. DOI: 10.1016/0021-9991(90)90201-B.

Dulikravich, G. S. and Tanaka, M. (Dec. 2000). *Inverse Problems in Engineering Mechanics II*. en. Elsevier. ISBN: 978-0-08-053515-9.

Dulikravich, G. S. and Tanaka, M. (Nov. 2001). *Inverse Problems in Engineering Mechanics III*. en. Elsevier. ISBN: 978-0-08-053514-2.

Forrester, A. I. J., Sóbester, A., and Keane, A. J. (July 2008). *Engineering Design via Surrogate Modelling: A Practical Guide*. en. First. Wiley. ISBN: 978-0-470-06068-1 978-0-470-77080-1. DOI: 10.1002/9780470770801.

Franz, A., Hoffmann, K., and Salamon, P. (July 2001). "Best Possible Strategy for Finding Ground States". In: *Physical Review Letters* 86. DOI: 10.1103/PhysRevLett.86.5219.

Gal, Y., van der Wilk, M., and Rasmussen, C. E. (Sept. 2014). "Distributed Variational Inference in Sparse Gaussian Process Regression and Latent Variable Models". en. In: *arXiv:1402.1389 [cs, stat]*. arXiv: 1402.1389 [cs, stat].

Geman, S., Bienenstock, E., and Doursat, R. (1992). "Neural Networks and the Bias/Variance Dilemma". In: *Neural Computation* 4.1, pp. 1–58.

Golub, G., Heath, M., and Wahba, G. (May 1979). "Generalized Cross-Validation as a Method for Choosing a Good Ridge Parameter". In: *Technometrics* 21, pp. 215–223. DOI: 10.1080/00401706.1979.10489751.

Golub, G. H. and Van Loan, C. F. (2013). *Matrix Computations*. en. Fourth edition. Johns Hopkins Studies in the Mathematical Sciences. Baltimore: The Johns Hopkins University Press. ISBN: 978-1-4214-0794-4.

Haario, H., Saksman, E., and Tamminen, J. (Apr. 2001). "An Adaptive Metropolis Algorithm". en. In: *Bernoulli* 7.2, p. 223. ISSN: 13507265. DOI: 10.2307/3318737.

Henderson, D., Jacobson, S. H., and Johnson, A. W. (2003). "The Theory and Practice of Simulated Annealing". en. In: *Handbook of Metaheuristics*. Ed. by F. Glover and G. A. Kochenberger. Vol. 57. Boston: Kluwer Academic Publishers, pp. 287–319. ISBN: 978-1-4020-7263-5. DOI: 10.1007/0-306-48056-5_10.

Hensman, J., Fusi, N., and Lawrence, N. (Sept. 2013). "Gaussian Processes for Big Data". In: *Uncertainty in Artificial Intelligence - Proceedings of the 29th Conference, UAI 2013*.

Hochstadt, H. (1989). *Integral Equations*. English. New York: Wiley.

Hoerl, A. E. and Kennard, R. W. (Feb. 1970). “Ridge Regression: Biased Estimation for Nonorthogonal Problems”. en. In: *Technometrics* 12.1, pp. 55–67. ISSN: 0040-1706, 1537-2723. DOI: 10.1080/00401706.1970.10488634.

Hofmann, T., Schölkopf, B., and Smola, A. J. (June 2008). “Kernel Methods in Machine Learning”. en. In: *The Annals of Statistics* 36.3, pp. 1171–1220. ISSN: 0090-5364. DOI: 10.1214/009053607000000677. arXiv: math/0701907.

Ingber, L. (1989). “Very Fast Simulated Re-Annealing”. en. In: *Mathematical and Computer Modelling* 12.8, pp. 967–973. ISSN: 08957177. DOI: 10.1016/0895-7177(89)90202-1.

Ingber, L. (Jan. 2000). “Adaptive Simulated Annealing (ASA): Lessons Learned”. en. In: *arXiv:cs.MS/0001018*. arXiv: cs.MS/0001018.

Kimeldorf, G. and Wahba, G. (Jan. 1971). “Some Results on Tchebycheffian Spline Functions”. en. In: *Journal of Mathematical Analysis and Applications* 33.1, pp. 82–95. ISSN: 0022-247X. DOI: 10.1016/0022-247X(71)90184-3.

Knuth, D. E. (1969). *The Art of Computer Programming. Volume 2*. English. Reading, Mass: Addison-Wesley.

König, H. (1986). *Eigenvalue Distribution of Compact Operators*. English. Basel; Boston: Birkhäuser Verlag. ISBN: 3-7643-1755-8 978-3-7643-1755-3.

Krauth, W. (2006). *Statistical Mechanics: Algorithms and Computations*. en. Oxford Master Series in Physics Statistical, Computational, and Theoretical Physics 13. Oxford: Oxford Univ. Press. ISBN: 978-0-19-851535-7 978-0-19-851536-4.

Lázaro-Gredilla, M. and Titsias, M. (Jan. 2011). “Variational Heteroscedastic Gaussian Process Regression.” In: Proceedings of the 28th International Conference on Machine Learning, ICML 2011, pp. 841–848.

Li, W. et al. (Sept. 2017). “A Novel Extension Algorithm for Optimized Latin Hypercube Sampling”. en. In: *Journal of Statistical Computation and Simulation* 87.13, pp. 2549–2559. ISSN: 0094-9655, 1563-5163. DOI: 10.1080/00949655.2017.1340475.

Liu, H. et al. (2020). “When Gaussian Process Meets Big Data: A Review of Scalable GPs”. en. In: *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–19. ISSN: 2162-237X, 2162-2388. DOI: 10.1109/TNNLS.2019.2957109.

Marsaglia, G. and Bray, T. (July 1964). "A Convenient Method for Generating Normal Variables". In: *Siam Review - SIAM REV* 6, pp. 260–264. DOI: 10.1137/1006063.

Mercer, J. (1909). "Functions of Positive and Negative Type, and Their Connection with the Theory of Integral Equations". In: *Philosophical Transactions of the Royal Society, London* 209, pp. 415–446.

Metropolis, N. et al. (June 1953). "Equation of State Calculations by Fast Computing Machines". In: *The Journal of Chemical Physics* 21.6, pp. 1087–1092. ISSN: 0021-9606. DOI: 10.1063/1.1699114.

Micchelli, C. A. and Pontil, M. (Jan. 2005). "On Learning Vector-Valued Functions". en. In: *Neural Computation* 17.1, pp. 177–204. ISSN: 0899-7667, 1530-888X. DOI: 10.1162/0899766052530802.

Mises, R. von. and Geiringer, H. (1964). *Mathematical Theory of Probability and Statistics*. English. New York: Academic Press.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of Machine Learning*. en. Second edition. Adaptive Computation and Machine Learning. Cambridge, Massachusetts: The MIT Press. ISBN: 978-0-262-03940-6.

Muandet, K. et al. (2017). "Kernel Mean Embedding of Distributions: A Review and Beyond". en. In: *Foundations and Trends® in Machine Learning* 10.1-2, pp. 1–141. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000060.

Neto, F. D. M. and Neto, A. J. d. S. (Sept. 2012). *An Introduction to Inverse Problems with Applications*. en. Springer Science & Business Media. ISBN: 978-3-642-32556-4.

O'sullivan, F., Yandell, B. S., and William, J. R. (Mar. 1986). "Automatic Smoothing of Regression Functions in Generalized Linear Models". In: *Journal of the American Statistical Association* 81.393, pp. 96–103. ISSN: 0162-1459. DOI: 10.1080/01621459.1986.10478243.

Press, W. H. et al. (1992). *Numerical Recipes in C : The Art of Scientific Computing*. English. Cambridge [etc.]: Cambridge University Press. ISBN: 0-521-43108-5 978-0-521-43108-8.

Rao, S. S. (2009). *Engineering Optimization: Theory and Practice*. en. 4th ed. Hoboken, N.J: John Wiley & Sons. ISBN: 978-0-470-18352-6.

Rasmussen, C. E. and Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. en. Adaptive Computation and Machine Learning. Cambridge, Mass: MIT Press. ISBN: 978-0-262-18253-9.

Rescorla, M. (Mar. 2015). "Some Epistemological Ramifications of the Borel–Kolmogorov Paradox". en. In: *Synthese* 192.3, pp. 735–767. ISSN: 0039-7857, 1573-0964. DOI: 10.1007/s11229-014-0586-z.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*. en. Springer Texts in Statistics. New York, NY: Springer New York. ISBN: 978-1-4419-1939-7 978-1-4757-4145-2. DOI: 10.1007/978-1-4757-4145-2.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003). *The Design and Analysis of Computer Experiments*. en. Springer Series in Statistics. New York, NY: Springer New York. ISBN: 978-1-4419-2992-1 978-1-4757-3799-8. DOI: 10.1007/978-1-4757-3799-8.

Schoenberg, I. J. (Oct. 1964). "Spline Functions and the Problem of Graduation". In: *Proceedings of the National Academy of Sciences of the United States of America* 52.4, pp. 947–950. ISSN: 0027-8424.

Schoenberg, I. J. (1988). "Metric Spaces and Completely Monotone Functions". English. In: pp. 115–145.

Schölkopf, B., Herbrich, R., and Smola, A. J. (2001). "A Generalized Representer Theorem". en. In: *Computational Learning Theory*. Ed. by G. Goos et al. Vol. 2111. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 416–426. ISBN: 978-3-540-42343-0 978-3-540-44581-4. DOI: 10.1007/3-540-44581-1_27.

Simpson, T. W. et al. (2001). "Kriging Models for Global Approximation in Simulation-Based Multidisciplinary Design Optimization". In: *AIAA Journal* 39.12, pp. 2233–2241. ISSN: 0001-1452. DOI: 10.2514/2.1234.

Smith, A. F. M. and Roberts, G. O. (1993). "Bayesian Computation Via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 55.1, pp. 3–23. ISSN: 0035-9246.

Song, L. (2008). "Learning via Hilbert Space Embedding of Distributions". PhD thesis. University of Sydney.

Spall, J. C. (2003). *Introduction to Stochastic Search and Optimization : Estimation, Simulation, and Control*. English. Hoboken: Wiley. ISBN: 0-471-33052-3 978-0-471-33052-3.

Tanaka, M. (Nov. 2003). *Inverse Problems in Engineering Mechanics IV*. en. Elsevier. ISBN: 978-0-08-053517-3.

Tanaka, M. and Dulikravich, G. S. (Nov. 1998). *Inverse Problems in Engineering Mechanics*. en. Elsevier. ISBN: 978-0-08-053516-6.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985). *Statistical Analysis of Finite Mixture Distributions*. English. Chichester: J. Wiley. ISBN: 0-471-90763-4 978-0-471-90763-3.

Trefethen, L. N. and Bau, D. (1997). *Numerical Linear Algebra*. English. Philadelphia, Pa.: SIAM. ISBN: 0-89871-361-7 978-0-89871-361-9.

von Luxburg, U. and Schoelkopf, B. (Oct. 2008). "Statistical Learning Theory: Models, Concepts, and Results". en. In: *arXiv:0810.4752 [math, stat]*. arXiv: `0810.4752 [math, stat]`.

Wang, L. et al. (Jan. 2005). "Gaussian Process Meta-Models for Efficient Probabilistic Design in Complex Engineering Design Spaces". en. In: *Volume 2: 31st Design Automation Conference, Parts A and B*. Long Beach, California, USA: ASMEDC, pp. 785–798. ISBN: 978-0-7918-4739-8. DOI: `10.1115/DETC2005-85406`.

Woodbury, M. A. (1950). *Inverting Modified Matrices*. English. Ed. by Princeton University. SRG Memorandum Report ; 42. Princeton, NJ: Department of Statistics, Princeton University.

Xiong, F. et al. (Aug. 2009). "Optimizing Latin Hypercube Design for Sequential Sampling of Computer Experiments". en. In: *Engineering Optimization* 41.8, pp. 793–810. ISSN: 0305-215X, 1029-0273. DOI: `10.1080/03052150902852999`.

Yang, X.-S. (July 2010). *Engineering Optimization: An Introduction with Metaheuristic Applications*. en. John Wiley & Sons. ISBN: 978-0-470-64041-8.

Yao, K. and Gao, J. (2016). "Law of Large Numbers for Uncertain Random Variables". In: *IEEE T. Fuzzy Systems* 24.3, pp. 615–621. DOI: `10.1109/TFUZZ.2015.2466080`.

Žilinskas, A. and Zhigljavsky, A. (Jan. 2016). "Stochastic Global Optimization: A Review on the Occasion of 25 Years of Informatica". en. In: *Informatica* 27.2, pp. 229–256. ISSN: 0868-4952, 1822-8844. DOI: `10.15388/Informatica.2016.83`.