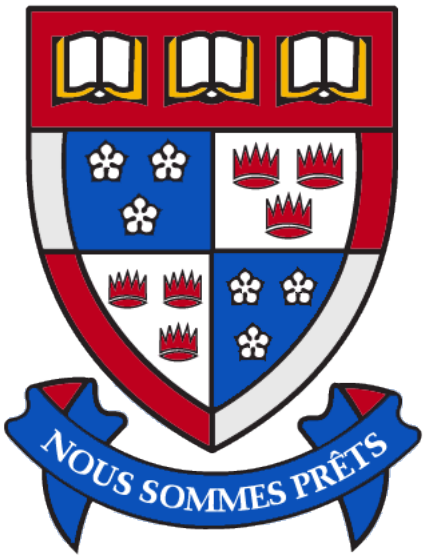


Simon Fraser University

Faculty of Sciences, Math Department

Here comes the title

Juan Gabriel García Osorio

Advisor: John Stockie

Commitee: Paul Tupper

Burnaby B.C - May 2017

Contents

Acknowlegments

Chapter 1

Introduction

Chapter 2

Theoretical and Computational Framework

The framework of Bayesian statistics is the foundation of our approach to estimate parameters and solve inverse problems. Unlike frequentist statistics, in the Bayesian approach, randomness is a measure of uncertainty or lack of information, not a matter of frequency. Consider a statement such as: the probability of having life in the universe is 0.01. In the frequentist perspective, this number is interpreted as: for every hundred planets, on average, one planet shelters life. In the Bayesian perspective the number 0.01 is interpreted as a measure of how certain we are about life in the universe given the current state of knowledge about the outer space. Clearly there is a big philosophical difference between these two approaches that has a direct impact in how far reaching is each point of view in terms of theoretical foundations and applications [8].

At this point we mention that when we talk about uncertainty we are

talking about every possible source of randomness or lack of information. That is, the use of the word uncertainty in this work is related to either [10]

- Epistemic: a phenomenon might not be random but the complete lack of understanding of it makes us see it as random.
- Aleatory: Inherent to the nature of the phenomenon. For example, this is the kind of randomness physicists believe is happening in quantum mechanics.

In real life the uncertainty associated with a measurement or quantity of interest is usually connected with the uncertainty of other variables involved in the problem under study. The Bayesian framework provides a rigorous framework to study these uncertainties, using whatever information is available for the underlying problem. The cornerstone of this framework in the mathematical language is known as Bayes' formula. Before we present it, let us introduce some important definitions taken from [6].

Definition 1. *A probability space is a triple* $(\Omega, \mathscr{F}, \mathbb{P})$*, where* Ω *is a set called sample space and* $\mathscr{F}$ *is a collection of subsets of* Ω *that satisfies*

1. $\emptyset, \Omega \in \mathscr{F}$.
2. *If* $A \in \mathscr{F}$ *then* $A^c \in \mathscr{F}$.
3. *If* $A_1, A_2, \ldots \in \mathscr{F}$ *then* $\bigcup_{i\in\mathbb{N}} A_i \in \mathscr{F}$.

A collection of sets that satisfies properties 1 to 3 is called a $\sigma-$ *algebra and its elements are called events.*
The map $\mathbb{P} : \mathscr{F} \to [0, 1]$ *is called a probability measure and satisfies*

1. $\mathbb{P}(\Omega) = 1$.

2. *If* $A_1, A_2, \ldots \in \mathscr{F}$ *are pairwise disjoint, then*

$$\mathbb{P}\left(\bigcup_{i\in\mathbb{N}} A_i\right) = \sum_{i\in\mathbb{N}} \mathbb{P}(A_i).$$

Definition 2. *Given a probability space* $(\Omega, \mathscr{F}, \mathbb{P})$ *and two events* $A, B \in \mathscr{F}$*, with* $\mathbb{P}(B) \neq 0$*, we define the conditional probability of* A *given* B *by*

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A \cap B)}{\mathbb{P}(B)}.$$

With the definitions above, we are now in position to state Bayes' fomula as

$$\mathbb{P}_{post}(A|B) = \frac{1}{Z}\mathbb{P}_{like}(B|A)\mathbb{P}_{prior}(A). \tag{2.1}$$

The sets A and B are subsets of the sample space Ω and are elements of the associated $\sigma-$algebra $\mathscr{F}$. The notation $\mathbb{P}_{like}(\cdot|\cdot)$ or $\mathbb{P}_{post}(\cdot|\cdot)$, denotes conditional probability. Let us introduce some terminology: the term $\mathbb{P}_{like}(B|A)$ is called the *likelihood* of B given A. The term $\mathbb{P}_{prior}(A)$ is called the *prior* probability for A. The prior probability expresses how much we believe the event A to happen without assuming anything about B. The reciprocal of Z is a *normalization constant* defined as

$$Z = \int_\Omega \mathbb{P}_{like}(B|A) d\mathbb{P}_{prior}. \tag{2.2}$$

The integral is understood in the Lebesgue sense as the integral with respect to the measure $\mathbb{P}_{prior}$ [12]. The term $\mathbb{P}_{post}(A|B)$ is the *posterior* probability of A given B. The posterior contains the information that we

gained by comparing our beliefs (decoded in the prior probability) with experimental data (decoded in the likelihood).

Now we look at the connection between Bayesian statistics and the field of inverse problems. Inverse problems are often concerned with finding the cause of an effect, whereas a forward problem is concerned with finding the effect of a cause. If we have information about the forward problem, then we can use it to obtain information about the inverse problem. Bayes' rule puts in a mathematical language the connection between the inverse and forward problems. If we consider the cause to be the event A and the effect the event B, then the information about the forward problem is encoded in $\mathbb{P}_{like}(B|A)$. The information of the inverse problem is contained in $\mathbb{P}_{post}(A|B)$. That is why in the Bayesian framework, the posterior probability is the *solution to an inverse problem.*

Often, inverse problems are ill-posed, which means that these problems might not satisfy one or more of the following properties [11]:

- Existence: There exists a solution for the problem.
- Uniqueness: The problem has a unique solution.
- Stability: Small changes in inputs result in small changes in outputs.

Any such lack of well-posedness is a serious issue. For example, if the problem under study has at least one solution but is unstable to small perturbations, how can we assess the accuracy of the solution to the problem? Therefore an statistical or non-deterministic approach is called for. As explained before,

the Bayesian framework is useful in this context. Bayes' rule connects the inverse problem of finding the cause of an effect through the posterior with the forward problem of finding the effect of a cause through the likelihood in a way that is possible to quantify the uncertainty about the solution of the problem. Let us clarify with an example of how the Bayesian framework can be used to solve inverse problems.

Consider the problem of finding the launch location of a rock that impacts (and cracks) a window. We can start by considering the following events

$$A = \text{Coordinates of the launching location.}$$
$$B = \text{Coordinates of the impact location on the window.}$$

Here we assume we know B, but A is unknown. We can use Bayes' rule to estimate A through the posterior $\mathbb{P}_{post}(A|B)$. In this case we need to find the connection between A and B via the forward problem, i.e. given the launch coordinates find the impact location. This connection is encoded in the likelihood $\mathbb{P}_{like}(B|A)$. In addition we also need to set the prior probability for A.

Let us explain how we could estimate the different probabilities mentioned in the previous paragraph. First, to find the likelihood we need to know how the rock's impact position in the window is related to the launch location. We can use the kinematic equations for parabolic trajectories to get [2]

$$\mathbf{r} = \mathbf{r}_0 + \mathbf{v}_0 t + \frac{1}{2}\mathbf{g}t^2, \tag{2.3}$$

where $\mathbf{r}$ and $\mathbf{r}_0$ are the final and initial position of the rock, $\mathbf{v}_0$ is the initial velocity, and $\mathbf{g}$ is a vector that points to the center of the earth and has a magnitude equal to the acceleration of gravity. The scalar t represents time. In a more physical language, to compute the likelihood it is necessary to estimate $\mathbf{r}$ (where the rock hit the window) assuming we know $\mathbf{r}_0$ (where it was thrown), and the initial velocity of the rock $\mathbf{v}_0$. Once all the other variables are identified the value of t can be computed in a straightforward manner.

Equations in physics are just models of reality and as such are just an approximation to it. For example, equation (2.3) does not consider air resistance or the Coriolis force. To take this into account we add an extra layer to the model by adding a random parameter that accounts for the discrepancy of our model with reality. We propose

$$\mathbf{r} = \mathbf{r}_0 + \mathbf{v}_0 t + \frac{1}{2}\mathbf{g}t^2 + \epsilon, \tag{2.4}$$

where ϵ is a *random vector* distributed as *multivariate Gaussian*. Before we proceed it is necessary to define more terminology and mathematical objects that are going to be used throughout the rest of the text.

Definition 3. *Given a set Ω, for any subset $T \subset \Omega$, we define the $\sigma-$algebra generated by T as the smallest $\sigma-$algebra in Ω that contains T.*

Definition 4. *Let O be the set of all open sets in $\mathbb{R}^n$. The $\sigma-$algebra generated by O is called the Borel $\sigma-$algebra and is denoted by $\mathcal{B}^n$. If $n = 1$ we denote $\mathcal{B}^1 := \mathcal{B}$.*

Definition 5. *Given a probability space $(\Omega, \mathscr{F}, \mathbb{P})$, a function $X : \Omega \to \mathbb{R}$ is*

called a random variable if $X^{-1}(C) \in \mathscr{F}$ *for all* $C \in \mathcal{B}$.

Definition 6. *An* n*-dimensional random vector* $\boldsymbol{X} = (X_1, \ldots, X_n)$ *in* $(\Omega, \mathscr{F}, \mathbb{P})$ *is a function* $\boldsymbol{X} : \Omega \to \mathbb{R}^n$ *such that each component is* X_i *is a random variable. Note that a single random variable can be considered as a one dimensional random vector.*

Definition 7. *Given a probability space* $(\Omega, \mathscr{F}, \mathbb{P})$ *and an* n*-dimensional random vector* $\mathbf{X} : \Omega \to \mathbb{R}^n$*, the distribution of* $\mathbf{X}$ *is the probability measure*

$$\mu : \mathcal{B}^n \to [0, 1],$$

where μ *is defined by*

$$\mu := \mathbb{P} \circ \boldsymbol{X}^{-1}.$$

Definition 8. *Given a random vector* $\boldsymbol{X} : \Omega \to \mathbb{R}^n$ *with probability distribution* μ*, we say that* $\mathbf{X}$ *is absolutely continuous with respect to the Lebesgue measure if there exists a real valued, integrable function* ρ *such that for all* $C \in \mathcal{B}^n$ *we have*

$$\mu(C) = \int_C \rho(x)dx.$$

We say that ρ *is the density function for* $\mathbf{X}$.

Definition 9. *Given an* n *dimensional random vector* $\mathbf{X}$ *such that for any* $C \in \mathcal{B}^n$ *we have*

$$\mu(C) = \int_C \frac{1}{2\pi det(\Sigma)^{-\frac{1}{2}}} \exp\left[(\boldsymbol{x} - \boldsymbol{x}^*)^T \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{x}^*)\right] d\boldsymbol{x}, \tag{2.5}$$

then we say that $\boldsymbol{X}$ *has a multivariate Gaussian distribution (or just Gaussian distribution) with mean* $\boldsymbol{x}^* \in \mathbb{R}^n$ *and covariance matrix* Σ*. The matrix* Σ *is*

symmetric and positive definite. We shall write

$$\boldsymbol{X} \sim \mathcal{N}(\boldsymbol{x}^*, \Sigma). \tag{2.6}$$

In this case the components of $\boldsymbol{X}$ *are said to be jointly Gaussian.*

We now return to equation (2.4), and assume $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. Here I represents the 3×3 identity matrix and $\sigma > 0$ parametrizes One belief in quantifying the accuracy of equation (2.3). By introducing a random variable into the model we make all variables involved in equation (2.3) to be random variables; that is, we now look at the associated stochastic equation. With this notation we can cast equation (2.1) as

$$\mathbb{P}_{post}(\mathbf{r}_0|\mathbf{r}, \mathbf{v}_0) = \frac{\mathbb{P}_{like}(\mathbf{r}|\mathbf{r}_0, \mathbf{v}_0)\mathbb{P}_{prior}(\mathbf{r}_0)}{Z}, \tag{2.7}$$

where we assumed independence between $\mathbf{r}_0$ and $\mathbf{v}_0$. Since ϵ is Gaussian we can readily obtain [26]

$$\mathbf{r}|\mathbf{r}_0, \mathbf{v}_0 \sim \mathcal{N}(\mathbf{r}_0 + \mathbf{v}_0 t + \frac{1}{2}\mathbf{g}t^2, \sigma^2 I).$$

This last equation gives an explicit density for the likelihood.

We now turn our attention to the prior. Suppose that we suspect the rock was thrown from the bedroom a neighbor. One way to model this suspicion is to assume a prior distribution on $\mathbf{r}_0$ as

$$\mathbf{r}_0 \sim \mathscr{N}(\mathbf{w}, \lambda^2 I),$$

where $\mathbf{w}$ is the coordinate vector of the center of mass of the neighbor's bedroom and λ represents One's belief the launch location is at the point $\mathbf{w}$. We note that this is only one way to model prior knowledge and other forms of the prior are also possible. Finally the normalization constant can be found as

$$\begin{aligned} Z &= \int_{\mathbb{R}^3} \mathbb{P}_{like}(\mathbf{r}|\mathbf{r}_0, \mathbf{v}_0)\mathbb{P}_{prior}(\mathbf{r}_0)d\mathbf{r}_0 \\ &= \frac{1}{(2\pi)^6(\sigma\lambda)^3}\int_{\mathbb{R}^3} \exp\left[-\frac{1}{2\sigma^2\lambda^2}\left(\|\mathbf{r} - (\mathbf{r}_0 + \mathbf{v}_0 t + \frac{1}{2}\mathbf{g}t^2)\|^2 + \|\mathbf{r}_0 - \mathbf{w}\|^2\right)\right] d\mathbf{r}_0. \end{aligned}$$

Having the likelihood, prior, and normalization constant allows us to compute the posterior using Bayes' rule. With these probabilities calculated we can obtain several different estimates for the value $\mathbf{r}_0$. Common choices of pointwise estimates include

$$\mathbf{r}_{MAP} = \underset{\mathbf{r}_0}{\operatorname{argmax}}\, \mathbb{P}_{post}(\mathbf{r}_0|\mathbf{r}, \mathbf{v}_0) \qquad \text{(Maximum a posteriori)}, \quad (2.8)$$

$$\mathbf{r}_{CM} = \int_{\mathbb{R}^3} \mathbf{r}_0 \mathbb{P}_{post}(\mathbf{r}_0|\mathbf{r}, \mathbf{v}_0)d\mathbf{r}_0 \qquad \text{(Conditional mean)}, \quad (2.9)$$

$$\mathbf{r}_{ML} = \underset{\mathbf{r}_0}{\operatorname{argmax}}\, \mathbb{P}_{post}(\mathbf{r}|\mathbf{r}_0, \mathbf{v}_0) \qquad \text{(Maximum likelihood)}. \quad (2.10)$$

Each of these estimates has its own strengths and weaknesses. If the posterior is bimodal, then the conditional mean might point at a value with very low probability, whereas the maximum a posteriori estimate might be more reliable. If the posterior has no critical points then the mean might be used as a point estimate. We can also assess how confident we are about the point estimate. For example, if $\mathbf{r}^*$ is our point estimate we can calculate a

number $\alpha > 0$ such that

$$\int_{B(\mathbf{r}^*,\alpha)} \mathbb{P}_{post}(\mathbf{r}_0|\mathbf{r},\mathbf{v}_0)d\mathbf{r}_0 = 0.95, \tag{2.11}$$

where $B(\mathbf{r}^*,\alpha)$ is the ball centered at $\mathbf{r}^*$ with radius α. This value of α can be thought of as the Bayesian version of the frequentist's 95% confidence interval. Another way to estimate uncertainty is by calculating the covariance matrix of $\mathbf{r}$ around $\mathbf{r}_0$ as

$$\int_{\mathbb{R}^3} (\mathbf{r}_0 - \mathbf{r}^*) \otimes (\mathbf{r}_0 - \mathbf{r}^*) d\mathbb{P}_{post}.$$

The diagonal of this matrix contains the variance of each coordinate of $\mathbf{r}^*$.

Note that the posterior is a probability density and does not necessarily have a closed form, which can make difficult to calculate the uncertainties we mentioned above. Hence we need a way of extracting information from $\mathbb{P}_{post}$. One approach is to generate independent samples from it and do a Monte Carlo integration to obtain the different uncertainty estimates. How to sample from a probability density and do a Monte Carlo integration will be explained in Chapter 3. For the moment, we assume it is possible to evaluate any of the point estimates and the uncertainty measures. With this, we can use a point estimate from equation (2.8) to obtain a method to infer the launch location of the rock and with equation (2.11) we can estimate how confident we are about that estimate.

Practical problems are often substantially more challenging than in the

above example. Often times we have to deal with further issues such as

1. Uncertainties in experimental measurements.
2. Lack of sufficient information about the physics of the problem and experimental data.
3. Computational complexity of physical models that are too expensive to evaluate.
4. Parameters that might belong to high dimensional spaces so the associated probability density is hard to sample from.
5. Evaluating any of the possible point estimates for the quantity of interest might computationally difficult.

In the problem that we outlined in Chapter 1, we have to deal with all of the above mentioned issues. In this chapter we are going to discuss our approach for dealling with issues 3, 4 and 5 above. We omit 1 and 2, since these are intrinsic to the physics of the problem and the methodology used to obtain the experimental data, and so, they are outside of our control.

2.1 Dealing with the computational complexity of the physical model

Models of physical processes can be represented in different ways. Following O'Hagan [18], we represent the mathematical model of the physical process as a function $M(\cdot)$ so that $y = M(\mathbf{x})$ where $\mathbf{x} \in \mathbb{R}^n$ and $y \in \mathbb{R}$. Mathematical

models are approximations to complex physical processes. Often times these mathematical models are expensive to compute. It is of great advantage if the complexity of the model can be reduced. One way to do this is by performing a sensitivity analysis on the parameters the model depends on. Roughly, we choose a combination of different values of the parameters and then we assess the importance of each parameter in the output. This means that we need to run the model $M(\cdot)$ for a large set of different combination of its parameters. Since realistic mathematical models are typically expensive, this implies that the use of classical methods such as correlation ratios, FAST method, Method of Sobol', etc. are not feasible (see [23] for details).

Here the concept of emulator as defined in [18] comes into play. We approximate the function $M(\cdot)$, which is expensive to evaluate, with a function $\widehat{M}(\cdot)$ that is cheap to evaluate. To construct such an approximation, we associate a probability distribution with each possible value of $M(\mathbf{x})$ and for example take $\widehat{M}(\mathbf{x})$ to be the mean of this distribution. We will refer to $\widehat{M}(\cdot)$ as an emulator. Following [18] we expect the emulator to satisfy the conditions in the following definition

Definition 10. *An emulator $\widehat{M}(\cdot)$ of a function $M(\cdot)$, is a map that:*

- *At points $\{\boldsymbol{x}\}_{k=1}^{N}$ were we know the output of the mathematical model (i.e. we know $M(\boldsymbol{x}_k)$ for $k = 1, 2, \ldots, N$) the emulator should satisfy $\widehat{M}(\boldsymbol{x}_k) = M(\boldsymbol{x}_k)$.*

- *For points $\{\boldsymbol{x}_k^*\}_{k=1}^{T}$ where we don't know the output $M(\boldsymbol{x}_k^*)$, the emulator should return an estimate $\widehat{M}(\boldsymbol{x}_k^*)$ based on the distribution for $M(\boldsymbol{x}_k^*)$. That estimate should reflect the uncertainty associated with the*

interpolation/extrapolation done at that point.

From now on in this work we are going to refer to the mathematical model or the computationally expensive function to calculate as $M(\cdot)$, and the emulator that approximates this function by $\widehat{M}(\cdot)$.

A popular method to construct an emulator with the desired extrapolation/interpolation properties is known as a Gaussian process regression.

2.1.1 Gaussian Processes

The conditions on the emulator $\widehat{M}(\cdot)$ imply that we need specify a probability distribution for each point $\mathbf{x}$ in the domain of the model $M(\cdot)$. This means that we need to work with a set with high cardinality of random variables. When dealing with several random variables there is one probability density that is computationally tractable and easy to work with: the multivariate Gaussian distribution (see Definition 9). The computational tractability of the multivariate Gaussian distribution can be used as a justification to define a Gaussian process.

Definition 11. *A Gaussian process (GP) is a collection of random variables* $\{g(\boldsymbol{x})\}_{\boldsymbol{x}\in A}$ *for some set A, possibly uncountable, such that any finite subset* $\{g(\boldsymbol{x}_k)\}_{k=1}^N \subset \{g(x)\}_{\boldsymbol{x}\in A}$ *for* $\{\boldsymbol{x}_k\}_{k=1}^N \subset A$ *is jointly Gaussian [21].*

A GP is specified by a mean function and a covariance operator or covariance kernel. Following Rasmussen [21] we define

$$m(\mathbf{x}) := \mathbb{E}(g(\mathbf{x})), \qquad \text{(Mean)}$$

$$k(\mathbf{x}, \mathbf{x}') := \mathbb{E}((g(\mathbf{x}) - m(\mathbf{x}))(g(\mathbf{x}') - m(\mathbf{x}'))) \qquad \text{(Kernel)}.$$

If $\{g(\mathbf{x})\}_{\mathbf{x}\in A}$ is a GP with mean $m(\cdot)$ and covariance $k(\cdot,\cdot)$ we will write

$$g(\mathbf{x}) \sim \mathbf{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')).$$

To understand why the notion of a GP is useful for us, recall that our goal is to create an emulator $\widehat{M}(\cdot)$ that approximates a function $M(\cdot)$. For a fixed $\mathbf{x} \in A$, a realization of the random variable $g(\mathbf{x})$ represents a possible value of $M(\mathbf{x})$. The mean function at that point $\mathbf{x}$, i.e. $m(\mathbf{x})$ represents the best prediction of the true value of $M(\mathbf{x})$, so we may set $\widehat{M}(\mathbf{x}) = m(\mathbf{x})$. Later we will show that one way to measure the uncertainty associated with that prediction is given by the quantity $k(\mathbf{x}, \mathbf{x})$.

We will use GPs to fit functions in high dimensional euclidean spaces, so that we may think of the index set A of Definition 11 as a subset of $\mathbb{R}^n$ for some $n \geq 1$.

The reason why Gaussian processes are useful in practice is that they are completely characterized by the mean $m(\cdot)$ and choosing a covariance kernel $k(\cdot,\cdot)$ [13]. For example a common covariance or kernel is the squared exponential (SE) function given by

$$k(\mathbf{x}, \mathbf{x}') = e^{-\frac{1}{2}\|\mathbf{x}-\mathbf{x}'\|_2^2}. \tag{2.12}$$

We choose to use the name "squared exponential" instead of Gaussian to avoid confusion with the probability distribution. This covariance function tells us that if $\mathbf{x}$ and $\mathbf{x}'$ are close in the Euclidean metric then they are highly correlated, whereas far away points have a correlation that decays

exponentially. How to choose the covariance function depends on the kind of regularity we want for the realizations of the GP. We will discuss this topic in more detail later in this chapter. For reference purposes, we list some of the most common kernels used in practice [21], setting $r = \|x - x'\|_2$:

- Squared-Exponential: $k(r;\theta) = \exp\left[-\frac{1}{2}(\frac{r}{\theta})^2\right]$
- Exponential: $k(r;\theta) = \exp\left[-\frac{r}{\theta}\right]$
- Matern $\frac{3}{2} : k(r;\theta) = (1 + \frac{\sqrt{3}r}{\theta})exp\left[-\frac{\sqrt{3}r}{\theta}\right]$.
- Matern $\frac{5}{2} : k(r;\theta) = (1 + \frac{\sqrt{5}r}{\theta} + \frac{5}{3}(\frac{r}{\theta})^2)\exp\left[-\frac{\sqrt{5}r}{\theta}\right]$.
- Power-Exponential: $k(r;\theta,p) = \exp\left[-(\frac{r}{\theta})^p\right]$.

Gaussian Processes as Distributions Over Function Spaces

Alternatively GPs can be viewed as measures on function spaces, and so we now discuss them in this context following the approach of [13]. Relevant function spaces (e.g. L^p spaces, Sobolev spaces, etc...) are normed vector spaces with a topology inherited from the metric induced by the norm. So the function spaces of interest here, are topological vector spaces (TVS).

Let $\mathscr{T}$ be a TVS and let $\mathscr{T}^*$ be its topological dual. We will denote the action of an element $h \in \mathscr{T}^*$ over an element $z \in \mathscr{T}$ with $\langle h, z\rangle$. Moreover we define a random variable taking values in $\mathscr{T}$ as a map

$$X : (\Omega, \mathscr{F}, P) \longrightarrow \mathscr{T},$$

that is measurable with respect to the σ-algebra generated by the open sets of $\mathscr{T}$. This σ-algebra is known as the Borel σ-algebra for $\mathscr{T}$. The triple $(\Omega, \mathscr{F}, P)$ is a probability space as in Definition 1. We use the shorthand notation $X \in \mathscr{T}$ whenever the random variable X takes values in $\mathscr{T}$. For example if $\mathscr{T} = L^2(\mathbb{R})$, then $X \in L^2(\mathbb{R})$ means that X is a measurable map from the probability space $(\Omega, \mathscr{F}, P)$ into $L^2(\mathbb{R})$.

We say that a random variable $X \in \mathscr{T}$ is called Gaussian if $\langle h, X \rangle$ is a Gaussian random variable on the real line for all $h \in \mathscr{T}^*$. An element $a \in \mathscr{T}$ is the expectation of $X \in \mathscr{T}$ if

$$\mathbb{E}(f, X) = \langle f, a \rangle, \qquad \text{for all } f \in \mathscr{T}^*.$$

Also a linear and positive definite operator $K : \mathscr{T}^* \longrightarrow \mathscr{T}$ is called the covariance operator (the covariance matrix in the finite dimensional case) if

$$\operatorname{cov}(\langle f_1, X \rangle, \langle f_2, X \rangle) = \langle f_1, K f_2 \rangle,$$

for all $f_1, f_2 \in \mathscr{T}^*$. Then we say that X is distributed as $\mathcal{N}(a, K)$. It is worth mentioning that given a covariance operator L and an element $b \in \mathscr{T}$ the distribution $\mathcal{N}(b, L)$ does not always exist [14]. But if it does exist, the Gaussian measure $\mathcal{N}(a, K)$, is completely identified with a and K.

As an example consider the $\mathscr{T} = C(T)$ where T is compact subset of $\mathbb{R}^n$. This is the space of real valued continuous functions on T. This is a Banach space with the norm [4]

$$\|h\| = \max_{x \in T} |h(x)|.$$

The dual space of $\mathscr{T}$ is given by $\mathscr{T}^* = \mathbb{M}(T)$ the set of signed measures defined on the Borel $\sigma-$ algebra of T. In this case the duality pairing is given by

$$\langle \mu, g \rangle = \int_T g d\mu.$$

A GP, $\{g(t)\}_{t\in T}$ (see definition 11) with mean function $m(t)$ and covariance kernel $k(t,t')$, can be thought of as a Gaussian measure $\mathcal{N}(m,K)$ where [13]

$$\begin{aligned} \mathbb{E}(f) &= m \in \mathbb{C}(T), \\ (K\nu)(t) &= \int_T k(t,t') d\nu(t'), \qquad \text{for } \nu \in \mathbb{M}(T). \end{aligned}$$

The above example illustrates the connection between GPs and distributions over function spaces, and more precisely how is connected to Gaussian measures on function spaces. Next we will explain how to use GPs in practice.

Assume we have data $\{(\mathbf{x}_i, y_i)\}_{i=1}^m \subset \mathbb{R}^n \times \mathbb{R}$ from an expensive function $M(\cdot)$, where $M(\mathbf{x}_i) = y_i$. The set $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$ is called the *training set*. For simplicity we do not presume any trend in the *training outputs* $\{y_i\}_{i=1}^m$. Given the training set we would like to infer possible values of $M(\cdot)$ on another set of points $\{\mathbf{x}_j^*\}_{j=1}^k$. This set of points is known as *test set*. For this purpose we construct an emulator $\widehat{M}(\cdot)$ (see introduction to section 2.1) by considering the GP denoted by $\{f(\mathbf{x})\}_{\mathbf{x}\in \mathrm{dom}(M)}$ where $\mathrm{dom}(M)$ is the domain of $M(\cdot)$.

By Definition 11, the random vectors

$$\mathbf{f} = \begin{bmatrix} f(\mathbf{x}_1) & \dots & f(\mathbf{x}_m) \end{bmatrix}^T,$$
$$\mathbf{f}^* = \begin{bmatrix} f(\mathbf{x}_1^*) & \dots & f(\mathbf{x}_l^*) \end{bmatrix}^T,$$

are jointly Gaussian with

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X,X) & K(X,X^*) \\ K(X^*,X) & K(X^*,X^*) \end{bmatrix}\right), \tag{2.13}$$

where the zero mean reflects the assumption of no trend in the training output $\{y_i\}_{i=1}^m$. The element of the submatrices in the covariance matrix are given by

$$(K(X,X))_{ij} = \text{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)),$$
$$(K(X,X^*))_{ij} = \text{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j^*)),$$

and so on. By the requirements of Defintion 10, the realization of the random vector $\mathbf{f}$ is known and is equal to $[y_1, \dots, y_m]^T$. Given this vector, we want to infer the vector $\mathbf{f}_*$. This can be achieved by obtaining the distribution of $\mathbf{f}_*|\mathbf{f}$. By well known properties of the multivariate Gaussian distribution we obtain [14]

$$\mathbf{f}^*|\mathbf{f} \sim \mathcal{N}(\langle \mathbf{f}^* \rangle, \Sigma), \tag{2.14}$$

where

$$\langle \mathbf{f}^* \rangle = K(X^*, X)K(X, X)^{-1}\mathbf{f},$$
$$\Sigma = K(X^*, X^*) - K(X^*, X)K(X, X)^{-1}K(X, X^*).$$

Note that if in the above equations, we only consider one test point, $\mathbf{x}^*$, and we take the limit as $\mathbf{x}^*$ approaches to the training input $\mathbf{x}_j$, the matrix $K(\mathbf{x}^*, X)$ reduces to a vector and converges to $K(\mathbf{x}_j, X)$. In this case, it is not hard to see that the mean would be given by

$$K(\mathbf{x}_j, X)K(X, X)^{-1}\mathbf{f} = y_j, \tag{2.15}$$

and the covariance matrix would reduce to an scalar that tends to zero as $\mathbf{x}^* \to \mathbf{x}$. The interpretation is that at a training input $\mathbf{x}_j$ the prediction is exactly equal to the corresponding training output y_j. For a point $\mathbf{x}^*$ that is not part of the training set, with 95% confidence we have

$$M(\mathbf{x}^*) \in [\langle f \rangle(\mathbf{x}^*) - 2\sigma, \langle f \rangle(\mathbf{x}^*) + 2\sigma], \tag{2.16}$$

where

$$\langle f \rangle(\mathbf{x}^*) = K(\mathbf{x}^*, X)K(X, X)^{-1}\mathbf{f} \qquad \text{(Mean at point } \mathbf{x}^*)$$
$$\sigma^2 = K(\mathbf{x}^*, \mathbf{x}^*) - K(\mathbf{x}^*, X)K(X, X)^{-1}K(X, \mathbf{x}^*) \quad \text{(Variance at point } \mathbf{x}^*).$$

Equations (2.15) and (2.16) show that if we define

$$\widehat{M}(\mathbf{x}^*) := \langle f \rangle(\mathbf{x}^*), \tag{2.17}$$

then $\widehat{M}(\cdot)$ satisfies the conditions for an emulator laidout in Definition 10.

In Figure 2.1, it is shown an example that summarizes the discussion above. We consider the problem of emulating the model $M(x) = \cos(2\pi x)$ having five training points. The 95% confidence region shows that in the training input $\mathbf{x}_j$ the variance is zero and $\widehat{M}(\mathbf{x}_j) = y_j$, as predicted by equation (2.15).

The covariance kernel is the quantity that defines the mean and covariance for the Gaussian distribution obtained when we look at finitely many random variables in a Gaussian Process. Therefore choosing it is a crucial step in the fitting process. We next discuss the properties of kernels and how to choose them depending on the data and the smooth properties we are looking for in the emulation process.

Covariance Kernels

The covariance kernel cannot be any arbitrary function $k(\mathbf{x}, \mathbf{x}^*)$. To see why, consider the matrix in equation (2.13) given by

$$C := \begin{bmatrix} K(X, X) & K(X, X^*) \\ K(X^*, X) & K(X^*, X^*) \end{bmatrix}.$$

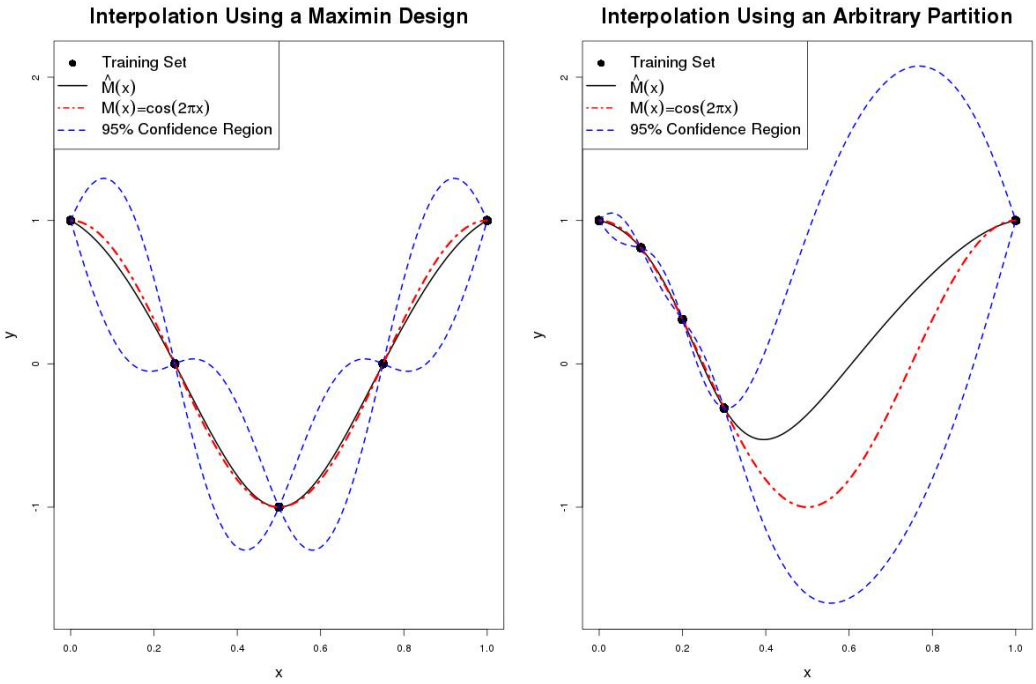


Figure 2.1: Comparison between the approximation quality of the emulator $\widehat{M}$ (solid line) for for the model $M(x) := \cos(2\pi x)$ (dashed-dotted line) in the interval $[0, 1]$ for two different partitions. On the left, the emulation is performed in the partition $\{0, 0.25, 0.5, 0.75, 1\}$. On the right in the partition $\{0, 0.1, 0.2, 0.3, 1\}$. The dashed line represents the 95% confidence region. The black points are the training set.

This is the covariance matrix of a multivariate Gaussian distribution and is obtained by evaluating the covariance kernel at different points. The matrix C must be symmetric and positive definite for any set of training and test inputs. This implies that the covariance kernel has to be symmetric, or

in other words, for all $\mathbf{x}$ and $\mathbf{x}'$ in the domain of $k(\cdot,\cdot)$ we must have

$$k(\mathbf{x},\mathbf{x}') = k(\mathbf{x}',\mathbf{x}).$$

We also need that for any set of inputs $\{\mathbf{x}_i\}_{i=1}^n$ the Gram matrix defined by $K_{jk} := k(\mathbf{x}_j,\mathbf{x}_k)$, must be positive definite. If k is just a function of $\mathbf{x}-\mathbf{x}'$, which is common for many kernels of practical interest, then $k(\cdot,\cdot)$ is said to be *stationary*.

To understand the role of the covariance kernel in the continuity and differentiability of the mean function, let us define some concepts first.

Definition 12. *Let $\boldsymbol{y}, \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$ be a sequence of points in $\mathbb{R}^n$, such that*

$$\|\boldsymbol{x}_n - \boldsymbol{y}\|_2 \to 0 \qquad \text{as } n \to \infty.$$

Then the collection of real valued random variables $\{f(\boldsymbol{x})\}$ defined in a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ are said to be continuous at y in the mean sense if

$$\mathbb{E}(|f(\boldsymbol{x}_n) - f(\boldsymbol{y})|^2) \to 0 \qquad \text{as } n \to \infty,$$

where

$$\mathbb{E}(f(x)) := \int_\Omega f(x) d\mathbb{P}.$$

We also have a definition for differentiability

Definition 13. *The mean square derivative of the collection $\{f(\boldsymbol{x})\}$ in the*

i-th direction at a point $\boldsymbol{y}$ *is*

$$\frac{\partial f(\boldsymbol{y})}{\partial x_i} = \lim_{h\to 0} \mathbb{E}\left(\left|\frac{f(\boldsymbol{y}+h\boldsymbol{e}_i) - f(\boldsymbol{y})}{h}\right|^2\right),$$

where $\boldsymbol{e}_i$ *is the i-th canonical vector of the standard basis in* $\mathbb{R}^n$*. The mean square* $n-th$ *derivative is given by*

$$\frac{\partial^n f(\boldsymbol{y})}{\partial x_i^n} = \lim_{h\to 0} \mathbb{E}\left(\left|\frac{\frac{\partial^{n-1} f(\boldsymbol{y}+h\boldsymbol{e}_i)}{\partial x_i^{n-1}} - \frac{\partial^{n-1} f(\boldsymbol{y})}{\partial x_i^{n-1}}}{h}\right|^2\right),$$

whenever the limit exists.

For Gaussian processes $\{f(\mathbf{x})\}$ with stationary covariance kernel, it can be shown that the process is continuous in the mean at a point $\mathbf{y}$ if and only if k is continuous at $(\mathbf{y}, \mathbf{y})$. Also the kernel function for the n-th derivative is given by [1]

$$\frac{\partial^{2n} k(\mathbf{x}, \mathbf{x}')}{\partial^2 x_1 \dots \partial^2 x'_m}.$$

Therefore the continuity and differentiability properties of the mean function in a Gaussian process depends exclusively in the continuity and differentiability properties of the covariance kernel.

Another important aspect of covariance kernels is that they are defined in terms of parameters. The way we choose the values of these parameters in practice, is based on the data we are analyzing. To see how this works, let us return to the problem of of approximating $M(\cdot)$ by $\widehat{M}(\cdot)$ using Gaussian processes. Let $k(x, x'; \theta)$ be the covariance kernel for the GP that depends on

the parameter θ, where θ could be a scalar, vector, etc. In this case to predict the output $\mathbf{y}^* = \{M(\mathbf{x}_1^*), \dots M(\mathbf{x}_m^*)\}$ given the training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$, we can try different approaches. One of the most common is maximum likelihood optimization (MLE), where we pick a parameter $\hat{\theta}$ such that

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}}\, \mathbb{P}(\mathbf{y}^* | \{(\mathbf{x}_i, y_i)\}_{i=1}^m, \theta).$$

By Definition 11 we know that the conditional probability for $\mathbf{y}^*$ has to be distributed as a multivariate Gaussian distribution. More precisely

$$p(\mathbf{y}^* | \{(\mathbf{x}_i, y_i)\}_{i=1}^m, \theta) = \frac{1}{(2\pi)^{\frac{m}{2}} \det(K_{\mathbf{y}^*}(\theta))^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\mathbf{y}^{*T} K_{\mathbf{y}^*}(\theta)^{-1} \mathbf{y}^*)\right], \tag{2.18}$$

where $K_{\mathbf{y}^*}(\theta)$ is the matrix $K(X, X)$ in equation (2.13). To find the value of $\hat{\theta}$ we have to maximize (2.18) with respect to θ. This goal is unchanged if we take the logarithm of both sides and minimize the following function instead[1]

$$L(\theta) = -\log(p(\mathbf{y}^* | \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^m, \theta)) = \frac{1}{2}\mathbf{y}^{*T} K_{\mathbf{y}^*}(\theta)^{-1} \mathbf{y}^* + \frac{1}{2}\log |K_{\mathbf{y}^*}(\theta)|. \tag{2.19}$$

A minimizer of $L(\theta)$ gives a possible value for $\hat{\theta}$ that explains the best the data $\mathbf{y}^*$ given the training set $\{\mathbf{x}_i, y_i\}_{i=1}^m$. Another common way to tune the parameters, is using K-fold cross validation, but will not use this approach

[1]The reason for taking the logarithm is because most software packages for optimization search for the minimum, not the maximum.

here (the interested reader is referred to [17] for details).

So far we have not discussed how to choose the training inputs $\{\mathbf{x}_i\}_{i=1}^m$. Clearly this choice has a profound impact on the accuracy of the emulator. To see this, let us assume that the function $M(\cdot)$ is supported in $[0, 1]$ and we have computational resources to calculate the output of only five training points. If we pick the points $\{0, 0.1, 0.2, 0.3, 1\}$ the interpolation error of the emulator $\widehat{M}(\cdot)$ for points between 0.3 and 1, will be large, compared to the error associated with the partition $\{0, 0.25, 0.5, 0.75, 1\}$ as shown in Figure 2.1.

Ideally we would like to pick as many training points as possible to improve the fit, but picking too many points to create the training set, can result in a very high computational cost. On the other hand, if we pick just few points to create the training set, then it is possible to end up with unreliable predictions. Thus we need a systematic way to choose the number and distribution of the training points. One strategy is to fill as much of the space as possible given a fixed number (possibly small) of training points. This can be accomplished through space-filling designs which we discuss next.

2.1.2 Design of Experiments

We assume there is a fixed computational budget. In this case, we need to decide how to choose the training inputs $\{\mathbf{x}_j\}_{j=1}^m$ to obtain reliable predictions of the emulator for points different than the training points. As shown in Figure 2.1, the quality of the emulation depends heavily on the distribution

of the training inputs. Intuitively we want to spread the training inputs as much as possible in the parameter space while covering as much space as possible. Distributions of points that achieve this are called *space filling designs*.

Given an set $T \subset \mathbb{R}^n$, there are several ways to create space filling designs. In this work we focus on maximin designs [9]. We note that there are other ways to obtain space filling designs and we refer to the reader to [19]. Consider a metric space (T, d) (e.g. $T \subset \mathbb{R}^n$, compact and d the Euclidean distance) and a subset S of T, with finite (fixed) cardinality, say $|S| = n$. A maximin distance design S^o is a collection of points of T such that

$$\max_{S \subset T,\ |S|=n} \min_{s,s' \in S} d(s, s') = \min_{s,s' \in S^o} d(s, s') = d^o.$$

That is, we are looking for a set S^o of cardinality n that maximizes the minimum distance among its elements. As an example consider $T = [0, 1]^3$, the unit cube in $\mathbb{R}^3$ and $n = 8$. In this case the design that maximizes the minimum distance among its elements is given by choosing the 8 vertices of the cube. Or as shown if Figure 2.1 (right), if $T = [0, 1]$ and $n = 5$, the maximin design is given by a uniform partition of the set T.

The problem of finding the optimal maximin design is difficult to solve in general. In practice we use computational tools to find a design that is close to optimal. Different algorithms can be used for the optimization of the design, such as genetic algorithms, simulated annealing, particle swarm, etc. A survey on the subject can be found in [27]. In Chapter 4 we will see how the particle swarm algorithm can be used to create a maximin design for a five dimensional parameter space.

To conclude this section, we note that there is a conection between maximin designs and Gaussian processes. Consider a GP $\{f(x)\}_{x\in T}$, fix $S = \{s_1, \ldots, s_n\} \subset T$, and consider the random vector

$$\mathbf{f} = [f(s_1), \ldots, f(s_n)],$$

where $\mathbf{f}$ is assumed to be jointly Gaussian. Let K_s be the correlation matrix for the probability distribution of $\mathbf{f}$. Then it can be shown that the minimax design minimizes the quantity

$$D(S) = -\det(K_s),$$

where the matrix K_s is the same as the covariance matrix in equation (2.13). A survey of the theory behind maximin distance designs can be found in [9].

2.1.3 Sensitivity Analysis

Having an space filling design for the training input permits us to create an emulator $\widehat{M}(\cdot)$ that closely approximates $M(\cdot)$ over its whole domain. By "closely" we mean within a tolerable uncertainty in the output of the emulator for all points in the domain (see Figure 2.1). If we have a reliable fitting, then we can confidently assess what parameters in the model $M(\cdot)$ are relevant and which ones are not. This ultimately allows to approximate the model with a simpler one. For example, if our model is given by

$$M(x_1, x_2, x_3) = x_1 + x_2 + 10^{-8}x_3, \qquad (x_1, x_2, x_3) \in T = [0, 1]^3,$$

then clearly the variable x_3 is not as relevant as x_1 or x_2. We need to formalize in what sense x_3 is irrelevant. One way to achieve this is by doing a sensitivity analysis. In summary, the goal of a sensitivity analysis is to assess how the output of a function $M(\cdot)$ depends on variations of its arguments. There is a great number of methods to perform a sensitivity analysis, such as adjoint methods, local methods, and variance based methods, to name a few. The primary difference between each of these methods is how they measure the importance of each variable. For example, in local methods the sensitivity at a point in a given direction is the slope of the function, whereas in variance based methods what matters is the magnitude of the area under the curve when fixing all parameters but one. For a survey of techniques in sensitivity analysis, the reader is referred to [23].

In this work we use variance-based Monte Carlo methods (VBMCM) as described in [25]. The idea of VBMCMs is to use the variance produced by the inputs of a function as an indicator of their importance. More precisely we will use the method of Sobol', which we outline next

The functions of interest in this work have compact support. This implies that without loss of generality we may assume that the domain of these functions is the n-dimensional unit cube Ω^n. Let us consider a generic square integrable function

$$\varphi : \Omega^n \to \mathbb{R},$$

and start by decomposing φ as

$$\varphi(x_1, \ldots, x_n) = \varphi_0 + \sum_{k=1}^{n} \varphi_k(x_k) + \sum_{1 \le k < l \le n} \varphi_{kl}(x_k, x_l) + \ldots + \varphi_{1,2,\ldots,n}(x_1, \ldots, x_n).$$

This decomposition is not unique, but it can be shown that if each term $\varphi_{i_1,\ldots,i_j}$ in the expansion satisfies

$$\int_{[0,1]} \varphi_{i_1,\ldots,i_j} dx_{i_k} = 0 \qquad \text{if } i_k \in \{i_1, \ldots, i_j\}, \tag{2.20}$$

then the decomposition is unique and all terms in the expansion are orthogonal in $L^2(\Omega^k)$. To demonstrate the orthogonality property, we may consider the functions $g = \varphi_{i_1,\ldots,i_j}$ and $h = \varphi_{ell_1,\ldots,ell_k}$ with arbitrary indices $(i_1, \ldots, i_j) \neq (\ell_i, \ldots, \ell_k)$. Without loss of generality we may assume $i_1 \neq \ell_1$. In this case we have

$$\langle g, h \rangle = \int_{[0,1]} \cdots \int_{[0,1]} \underbrace{\left(\int_{[0,1]} \varphi_{i_1,\ldots,i_j} dx_{i_1} \right)}_{= 0 \text{ by } (2.20)} \left(\int_{[0,1]} \varphi_{l_1,\ldots,l_k} dx_{l_1} \right) dx_{\sim i_1, l_1} = 0,$$

where we used Fubinni's theorem to split the integrals [12]. The symbols to the right of $\sim$ represent the variables omitted in the integration. Another consequence of (2.20) is

$$\int_{\Omega^n} \varphi dx = \varphi_0.$$

This allows us to find the other functions in the decomposition recursively, given φ_0. For example, for $i \in \{1, \ldots, n\}$ we have

$$\varphi_i(x_i) = -\varphi_0 + \int_{[0,1]^{n-1}} \varphi(x) dx_{\sim i}.$$

Having $\varphi_i(x_i)$ we can then proceed to find $\varphi_{ij}(x_i, x_j)$ using

$$\varphi_{ij}(x_i, x_j) = -\varphi_0 - \varphi_i(x_i) - \varphi_j(x_j) + \int_{\Omega^{n-2}} \varphi(x) dx_{\sim ij}.$$

By knowing all of the functions in the decomposition of φ we are able to assess how each variable affects the output of φ in the following way. The total variance D of φ is defined as

$$D = \int_{\Omega^n} \varphi^2(x) dx - \varphi_0^2,$$

and similarly we can compute the partial variances as

$$D_{i_1,\ldots,i_s} = \int_{[0,1]^{n-1}} \varphi^2_{i_1,\ldots,i_s} dx_{i_1} \ldots dx_{i_s}.$$

With these variances we define the s-th order Sobol index

$$S_{i_1,\ldots,i_s} = \frac{D_{i_1,\ldots,i_s}}{D},$$

which is a measure of the contribution of the variables $x_{i_1}, \ldots, x_{i_s}$ to the total variance D. If we want to know the separate effect of each variable $x_1, \ldots, x_n$ in the total variance D, we look at the first order Sobol indices $S_1, \ldots, S_n$ given by

$$S_i = \frac{D_i}{D}, \qquad \text{for } i = 1, \ldots, n.$$

Finally if we want to assess the full effect of a variable in the total variance D, we calculate a quantity known as the total effect index. For example if we want to calculate the total effect index for the variable x_i we would do so

by calculating

$$S_i + S_{i1} + S_{i2} + \ldots + S_{i12} + S_{i13} + \ldots + S_{12\ldots,i,\ldots,n}.$$

Note that to calculate each Sobol' index, it is necessary to perform high dimensional integrals. Therefore integration using quadratures is not feasible. It is necessary to resort to other numerical integration techniques. A common tool to perform high dimensional integrals is known as Monte Carlo integration. We will not go into details of Monte Carlo integration in this chapter, but rather postpone them for Chapter 3. What is important at this time is that to apply Monte Carlo integration, it is necessary to evaluate the integrand a large number of times at different points in its domain. If the integrand is the expensive model $M(\cdot)$, then the computational cost of estimating the Sobol' indices is prohibitive. If instead we calculate the Sobol' indices of the emulator $\widehat{M}(\cdot)$, we can use them as an approximation for the Sobol' indices of $M(\cdot)$. In this way we can estimate what arguments of the model are relevant and what arguments are not. This will allow us to reduce the complexity of the model.

In the next chapter we will show how the theory explained in this chapter can be applied in a practical setting using a toy problem. In Chapter 4 we proceed to apply the tools developed here to the atmospheric dispersion problem explained in Chapter 1.

Chapter 3

Toy Problem: How Theory Works in Practice

In the previous chapter we reviewed some of the theoretical and computational tools needed to solve a Bayesian inverse problem. In this chapter we are going to present a toy problem to illustrate how the theory can be applied in practice. We begin by considering the forward problem, given by the following partial differential equation (PDE).

$$\begin{cases} \Delta u = e^{-b\|\mathbf{x}\|_2}, & \text{for } x \in \Omega = [0,1] \times [0,1] \subset \mathbb{R}^2, \\ u = 0, & \text{for } x \in \partial\Omega, \end{cases} \tag{3.1}$$

where b is some real positive parameter. For us, the function u represents the mathematical approximation of a quantity $\tilde{u}$ that has a physical realization. For example we may think of $\tilde{u}$ as the actual difference in electric potential in Ω relative to a reference point and u as the mathematical approximation to it. Since mathematical models of the physical world are simplification of

reality, it is convenient to make a clear distinction between physics (e.g. $\tilde{u}$) and mathematics (e.g. u).

In Chapter 2 Section 2.1, we explained how to build an emulator $\hat{M}(\cdot)$ that approximates the output y of a computationally expensive function $M(\cdot)$ at a point in its domain. In this chapter, the function $M(\cdot)$ takes as input a point $(\mathbf{x}, b) \in \Omega \times (0, \infty)$. The output is the value of the solution u at that point, i.e. $u(\mathbf{x}; b) = M(\mathbf{x}, b)$. Now we proceed to explain the associated inverse problem and how we are going to construct $\hat{M}(\cdot)$.

Assume that we have ten experimental measurements of $\tilde{u}$. These measurements were taken at the points $P := \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{10}\} \subset \Omega$. That is, we know the vector of measurements $\mathbf{y} = (\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$. We want to estimate the value of b that explains the experimental data $\mathbf{y}$ the best. This is our inverse problem. A simple approach to estimate b would be to solve equation (3.1) for a big number of values b in the interval $(0, L]$ where L is chosen in a manner that there exists a $b^* \in (0, L]$ such that the vector $(u(\mathbf{x}_1; b^*), \ldots, u(\mathbf{x}_{10}; b^*))$ has 'small' discrepancy with the experimental data $\mathbf{y}$. This approach is not feasible if solving the forward model is computationally expensive.

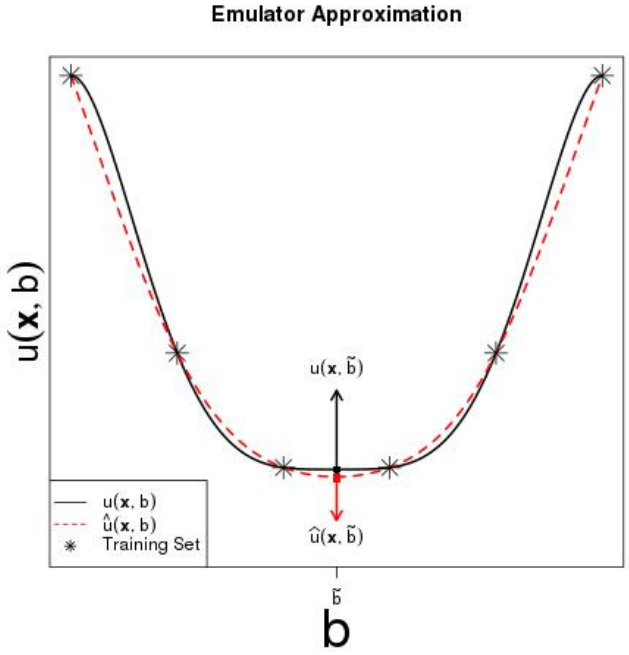


Figure 3.1: Approximation of a model $u(\mathbf{x};\cdot)$ by the mean of a Gaussian process trained with six different outputs from the model. The mean of the Gaussian process at a point $\tilde{b}$ is taken as the value $\widehat{u}(\mathbf{x};\tilde{b})$ of the emulator.

Let us assume that solving equation (3.1) is computationally expensive and repeating the calculation for a big range of different values of b is not feasible. One way to get around that is by constructing an emulator $\widehat{u}(\cdot)$ that approximates $u(\cdot)$ and is cheap to compute. The way we are going to construct $\widehat{u}(\cdot)$ is as follows: for a fixed $\mathbf{x} \in \mathbb{R}^2$ we solve equation (3.1) for n different values of b. We pick the value of n in a way that the computational cost of computing (3.1) n times, does not exceed our computational and time budget. Then use the data $\{b_j, u(\mathbf{x}, b_j)\}_{j=1}^{n}$ as a training set to create a

Gaussian process, as explained in Chapter 2, Section 2.1.1. Finally for any value $\tilde{b}$ we use the mean of the Gaussian process at that point as $\widehat{u}(\mathbf{x}, \tilde{b})$. An sketch from the result for approximating an arbitrary model $u(\mathbf{x}; \cdot)$ with an emulator $\widehat{u}(\mathbf{x}; \cdot)$ is shown in Figure 3.1.

For clarity in the exposition, the table below summarizes the notation we are going to use throughout the rest of the chapter.

Symbol	Meaning
$\tilde{u}(\mathbf{x}; b)$	Value of the physical variable at the point $\mathbf{x}$ with parameter b.
$u(\mathbf{x}; b)$	Numerical solution of equation (3.1) at $\mathbf{x}$ with parameter b.
$\hat{u}(\mathbf{x}; b)$	Value of the interpolation of the emulator $\hat{M}(\cdot)$ at the point $\mathbf{x}$ with parameter b .
$P := \{\mathbf{x}_1, \ldots, \mathbf{x}_{10}\}$	Points where the experimental measurements were taken .
$\mathbf{y} := (\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$	Values of the experimental measurements for the variable $\tilde{u}$.

Table 3.1: Summary of symbols used in Chapter 3.

Let us return to our original goal: to estimate the value of b that explains the experimental data $\mathbf{y}$ as best as possible. To create the experimental data

$\mathbf{y}$ we assume that the true value of b is $0,925$. Then, for this value of b, we solve equation (3.1) using a finite difference five point stencil approximation for the Laplacian. Next we pick ten points at random in Ω and save the value of the numerical solution u at those location (see Figure 3.2). Finally we add noise from a normal distribution with mean zero and standard deviation 0.01 to each of the ten values. The resulting numbers are what we use as the experimental data $\mathbf{y} = (\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$. Note that the noise added to the data obtained from the numerical solution of equation (3.1) plays the role of possible errors in the experimental measurements plus inaccuracies of the model to describe the true behavior of the physical variable $\tilde{u}$.

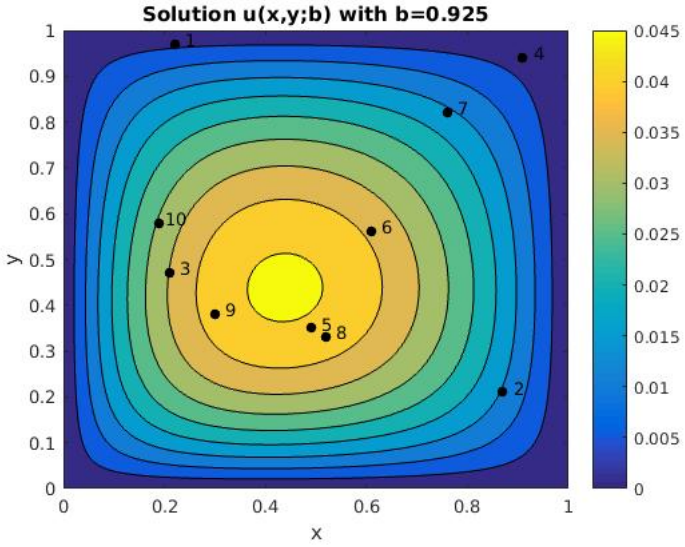


Figure 3.2: Numerical solution of the system (3.1) using a five points stencil finite difference approximation for the Laplacian. The mesh size used in x and y was 0.01. The value of the parameter b was set at 0.925. The black dots in the plot represent the points used to generate the experimental data $\mathbf{y} = (\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$

With the experimental data $\mathbf{y}$ created, we now proceed to obtain a point estimate value of b that produced that data. To that end we first compute the posterior distribution. We are going to explain step by step how to obtain such distribution.

3.1 Computing the Posterior

To calculate the posterior we use Bayes' rule to get

$$\mathbb{P}_{post}(b|\mathbf{y}) = \frac{\mathbb{P}_{like}(\mathbf{y}|b)\mathbb{P}_{prior}(b)}{Z(\mathbf{y})}. \tag{3.2}$$

Note that finding the posterior enables us to obtain any of point estimate from equation (2.8) and the uncertainty associated with that estimate. To compute $\mathbb{P}_{post}(b|\mathbf{y})$ we need to choose a prior distribution and the likelihood for b. Let us start with the prior.

3.1.1 Choosing the Prior

For the sake of the example let us assume that it is known that the parameter b cannot be greater than 2. In this case one way to choose a prior distribution for b that does not assume any other knowledge than $b \in (0, 2]$, is the *uniform distribution*. In this case we have

$$\mathbb{P}_{prior}(b) = \frac{1}{2}\mathbf{1}_{(0,2]}(b), \qquad \text{for all } b \in \mathbb{R}, \tag{3.3}$$

where $\mathbf{1}_{(0,2]}$ is the indicator function of the set $(0, 2]$. The indicator function for a Borel measurable set C is defined as

$$\mathbf{1}_C(y) = \begin{cases} 1 & \text{if } y \in C \\ 0 & \text{if } y \in C^c. \end{cases}$$

3.1.2 Finding the Likelihood

To calculate the likelihood,first we need to know how the set of possible measurements $\mathbf{y} = (\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$ is related to b when we let b to vary. Since we don't know the experimental values of the physical variable $\tilde{u}$

for different values of b, it is necessary to approximate the relation between $\tilde{u}$ and b by the relation between u and b. To obtain such relation we need to solve equation (3.1). By solving this equation explicitly we can find a functional relation between u and b for each one of the ten locations depicted in Figure 3.2. It is possible to solve analytically equation (3.1). However the relation between u and b is given by an infinite series. Indeed equation (3.1) is Poisson's equation with homogeneous boundary conditions. This equation can be solve using an eigenfunction expansion [15]. The eigenfunctions of the Laplacian in the unit square are given by

$$\phi_{mn} = \sin(n\pi x)\sin(m\pi y), \qquad \text{for } m, n \in \mathbb{N},$$

with eigenvalues

$$\lambda_{mn} = (n\pi)^2 + (m\pi)^2.$$

The eigenfunction expansion for u in equation (3.1) is

$$u = \sum_{n=1}^{\infty}\sum_{m=1}^{\infty} a_{mn}\phi_{mn}$$

where

$$a_{mn}\lambda_{nm} = -\frac{\int_\Omega e^{-b\|x\|^2}\phi_{mn}d\mathbf{x}}{\int_\Omega \phi_{mn}^2 d\mathbf{x}} = -\frac{\langle e^{-b\|x\|^2}, \phi_{mn}\rangle}{\|\phi_{mn}\|^2_{L^2(\Omega)}}. \tag{3.4}$$

The symbol $\langle\cdot,\cdot\rangle$ represents the standard inner product in $L^2(\Omega)$.

Having a functional relation given by an infinite series is often not very useful. For example in equation (3.4) the integral on the numerator does not have a closed form. Hence we need a different approach to gain insight into the relation between $\mathbf{y}$ and b. The approach we will use is the same as the

one that allowed us to obtain Figure 3.1. First we solve equation (3.1) for n different values of b. For the sake of the example assume $n = 10$. Then for each $\mathbf{x}$ in $P = \{\mathbf{x}_1, \ldots, \mathbf{x}_{10}\}$ we use the set $\{b_j, u(\mathbf{x}_k; b_j)\}_{j=1}^{10}$ to train a Gaussian process for each $k = 1, 2, \ldots, 10$. Finally for any $\tilde{b} \in (0, 2]$ we use the mean of the Gaussian process at that point as the value $\widehat{u}(\mathbf{x}_k; \tilde{b})$. By proceeding in this manner we obtain a cheap method to approximate the behavior of $\mathbf{y} = (\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$ when we let b to vary.

The next step is to choose the values of b for which the PDE (3.1) is solved in a way the uncertainty associated with to the emulator is as small as possible. We shall denote the points we choose as $\{b_1, \ldots, b_{10}\}$. To choose the points we use a maximin design as explained in Chapter 2 Section 2.1.2. In this case it is straightforward to check that the maximin design is the set of equidistant points

$$\{b_1 = 0.2, b_2 = 0.4, \ldots, b_{10} = 2\}.$$

By solving equation (3.1) for these values of b and for each $\mathbf{x}$ in P, we know the values in the set $\{u(b_j, \mathbf{x}_k)\}_{j,k=1}^{10}$. We use this set to train ten Gaussian Process. With these processes we define the functions

$$G_k : (0, 2] \to \mathbb{R} \qquad \text{for } k = 1, 2, \ldots 10.$$

such that for each k and b, the value of the mean of the k-th GP is going to

be given by $G_k(b)$. That is, $G_k(\cdot)$ is the emulator for $u(\mathbf{x}_k, \cdot)$. More precisely

$$G_k(b) = \hat{u}(\mathbf{x}_k; b).$$

The functions $G_k(\cdot)$ are cheap to evaluate and are a good approximation of $\tilde{u}(\mathbf{x}_k, \cdot)$. Now it is possible to approximate the value of b that explains $\mathbf{y} = (\tilde{u}(\mathbf{x}_1, b), \ldots, \tilde{u}(\mathbf{x}_{10}, b))$ by trying a big number of different values of b and then compare with the experimental data, to see what choice of b gives the smallest discrepancy. To this end, we calculate the values of $G_k(\cdot)$, for $k = 1, \ldots, 10$ in the set

$$\{0.01, 0.02, \ldots, 1.99, 2\}.$$

In Figure 3.3 are plotted the emulator at these points, the true value of b, the experimental measurement $\tilde{u}(\mathbf{x}_k; b)$ and the training data $\{u(\mathbf{x}_k; b_j)\}_{j=1}^{10}$ for each of the ten sites.

We are now ready to make the mathematical connection between $\tilde{u}, u$ and $\hat{u}$. Recall that u is the mathematical approximation of the physical variable $\tilde{u}$ and $\hat{u}$ is an emulator for u. Hence if $\hat{u}$ approximates well u, we would expect that $\hat{u}$ approximates $\tilde{u}$. For any point $\mathbf{x}_k \in P$ we do not know exactly how $\hat{u}(\mathbf{x}_k, \cdot) = G_k(\cdot)$ differs from $\tilde{u}(\mathbf{x}_k, \cdot)$. If we define $y_k(b) := \tilde{u}(\mathbf{x}_k, b)$, then, a possible relation that connects these quantities is given by the following Gaussian additive model [26]

$$y_k(b) = G_k(b) + \epsilon_k, \qquad \text{with } \epsilon_k \sim \mathcal{N}(0, \lambda^2), \tag{3.5}$$

where λ is a positive number that models how much we believe the emulator prediction differs from $\tilde{u}$. We chose the value $\lambda = 5.4 \times 10^{-3}$ to get a signal to noise ratio of 1:10. By defining the vector $\mathbf{G}(b) = (\hat{u}(\mathbf{x}_1; b), \ldots, \hat{u}(\mathbf{x}_{10}; b))$ and the definition of $\mathbf{y}$ (see table 3.1). Then equation (3.5) can we written more compactly as

$$\mathbf{y} = \mathbf{G}(b) + \epsilon, \qquad \text{with } \epsilon \sim \mathcal{N}(0, \lambda^2 I_{10\times 10}). \tag{3.6}$$

Since the random vector ϵ has a Gaussian distribution, we can use equation (3.6) to conclude

$$\mathbf{y}|b \sim \mathcal{N}(\mathbf{G}(b), \lambda^2 I_{10\times 10}),$$

i.e.

$$\mathbb{P}_{like}(\mathbf{y}|b) \propto e^{-\frac{1}{2\lambda^2}\|\mathbf{G}(b)-\mathbf{y}\|_2^2}, \tag{3.7}$$

where the proportionality constant normalizes the distribution on the right hand side to one.

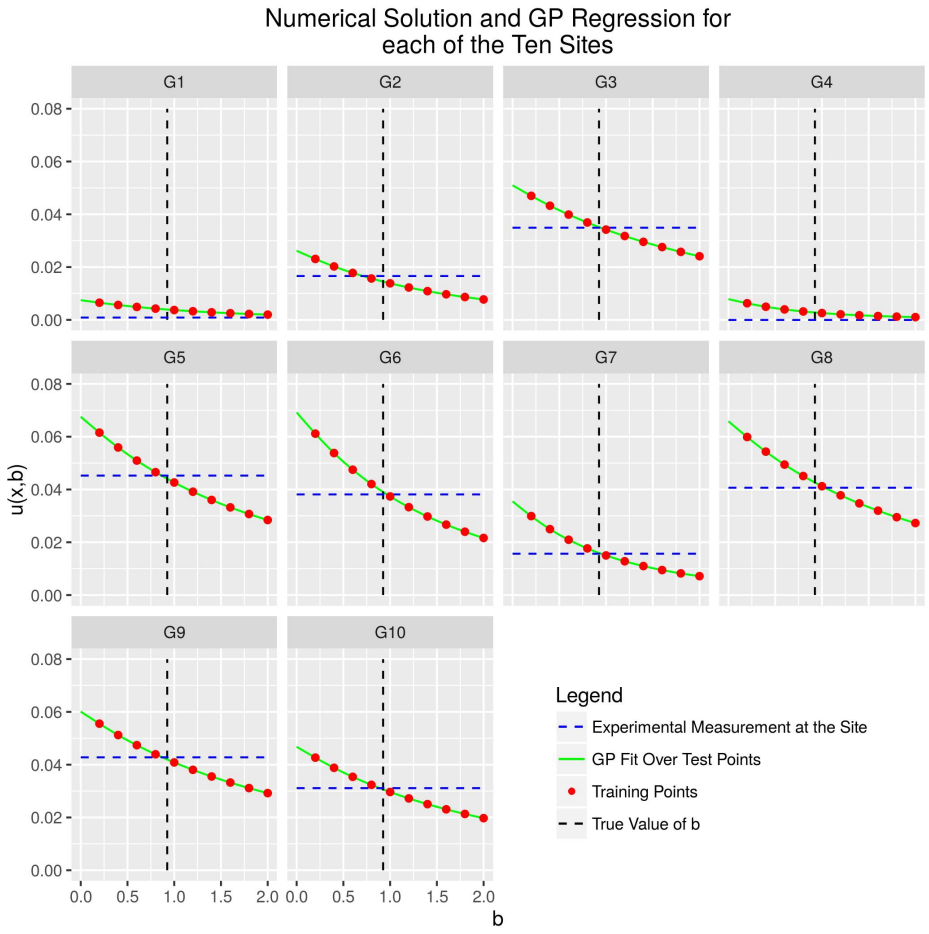


Figure 3.3: Training points, GP regression, true value of b and experimental measures for each one of the ten sites labeled from 1 to 10 in Figure 3.2

Now that we have explicit expressions for the prior and likelihood distributions, we can compute the posterior probability for b. Since the denominator in Bayes' rule (3.2) is independent of b, we can use equations (3.3) and (3.7) to write

$$\mathbb{P}_{post}(b|\mathbf{y}) \propto \mathbb{P}_{like}(\mathbf{y}|b)\mathbb{P}_{prior}(b) \propto \mathbf{1}_{(0,2]}(b)e^{-\frac{1}{2\lambda^2}\|\mathbf{G}(b)-\mathbf{y}\|_2^2}. \tag{3.8}$$

An interpretation of this result is that before taking experimental measurements we only knew that $b \in (0, 2]$. After weighting this prior belief with the data $\mathbf{y}$, our current state of knowledge about the parameter b is encoded in the posterior distribution. Figure 3.4 shows this updated distribution.

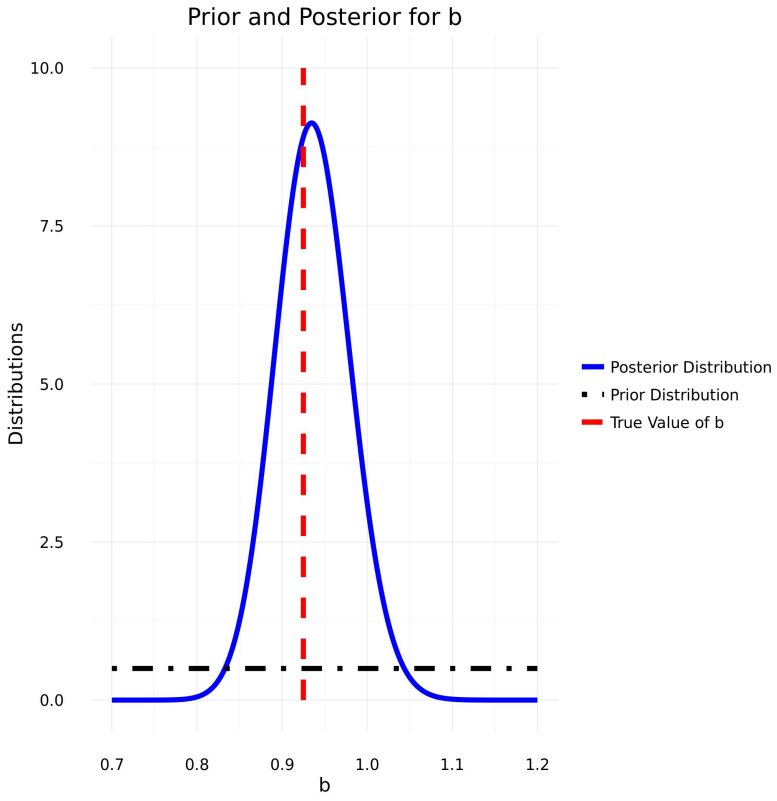


Figure 3.4: Plots of the prior distribution, posterior distribution and true value of the parameter b.

It is not always possible to visualize a probability density so it is necessary to sample from it in order to obtain statistics about the parameters of interest. A family of methods for this purpose are known as Markov Chain Monte Carlo (MCMC). In this work we focus on a particular algorithm known as

Metropolis-Hastings (MH). We now proceed to explain how MH works in practice using the posterior for b in equation (3.8) as an example.

Consider the posterior density $\mathbb{P}_{post}(b|\mathbf{y})$. The idea is to construct a Markov chain that wanders around the support of the posterior in a way that the chain spends more time in regions with high probability. One way to achive that is as follows: if we are at a point q_1 and we want to move to a point q_2 we will accept that move with probability one if $\mathbb{P}_{post}(q_1|\mathbf{y}) \leq \mathbb{P}_{post}(q_2|\mathbf{y})$ and with probability $\frac{\mathbb{P}_{post}(q_2|\mathbf{y})}{\mathbb{P}_{post}(q_1|\mathbf{y})}$. We choose in what direction to move, randomly, using some probability distribution that is easy to sample from. For simplicity, In this and the next Chapter we chose the uniform distribution to decide in what direction to move. The pseudocode for the MH algorithm as described above is [26]

Algorithm 1 Metropolis-Hastings Algorithm

```
pick a point q_1 in the support of the distribution
for j=2:N do
    Draw u ~ U([0, α])
    q_j ← q_{j-1} + u
    β ← min(1, P_post(q_j|D) / P_post(q_{j-1}|y))
    Draw w ~ U([0, 1])
    if w < β then
        q_{j-1} = q_j        (Accept the move)
    else
        q_{j-1} = q_{j-1}    (Reject the move)
    end if
end for
```

The rule of thumb for choosing the parameter α in the scheme above is that the proportion of times we accept a move is about 0.25 [5]. It can

be shown that the sequence $q_1, q_2, \ldots, q_N$ are realizations of a Markov chain that in the limit as $N \to \infty$ are distributed according to the distribution $\mathbb{P}_{post}(b|\mathbf{y})$. This convergence result works under mild conditions over the distribution that is being sampled. For more details about the theory behind MCMC methods we refer the reader to [5]. Since we do not have the computational power to let $N \to \infty$. We let the chain run for a large number of steps until it converges. Then, we throw away the *burn-in* portion of the chain and compute statistics using the remaining samples. The burn-in portion of the chain are the samples obtained before the chain is close to converge. A common choice is to discard the first $\frac{N}{2}$ samples.

Using Algorithm 1, we sample from the posterior distribution $\mathbb{P}_{post}(b|\mathbf{y})$ setting the values $\alpha = 0.23$ and $N = 10000$. The burn-in period is set to be the first 5000 samples. An histogram of the last 5000 is shown below.

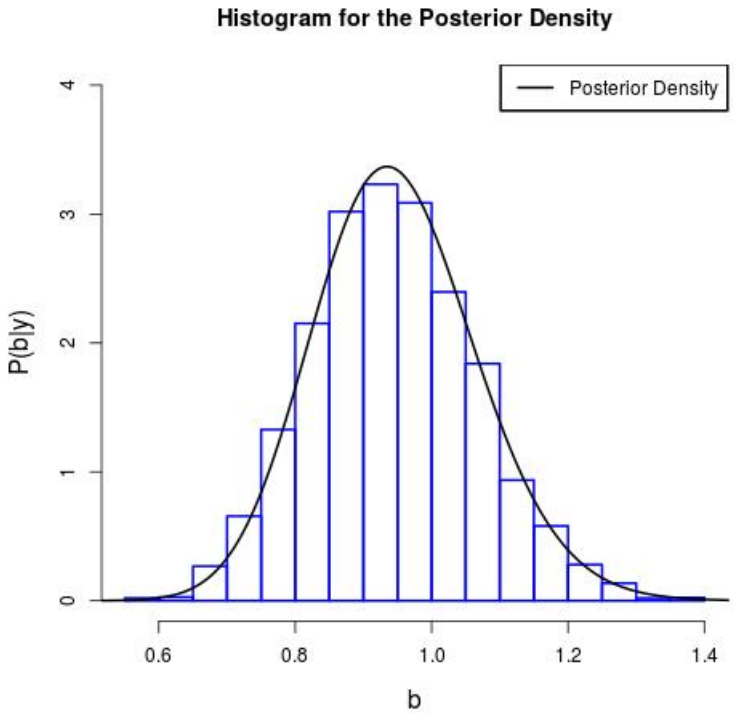


Figure 3.5: Histogram obtained for the posterior distribution (3.2) from 5000 samples from MH algorithm with step size $\alpha = 0.23$. The solid line is the graph for the posterior $\mathbb{P}_{post}(b|D)$.

With the samples obtained we readily obtain useful statistics for b. For example, we can estimate the conditional mean using [5]

$$b_{cm} = \int_{(0,2]} b\mathbb{P}_{post}(b|D)db \approx \frac{1}{5000}\sum_{j=1}^{5000} b_j = 0.9247042, \tag{3.9}$$

where the summands b_j are the samples obtained after the burn-in period of

5000 samples. We can also estimate the variance of the samples as

$$\int_{(0,2]} (b - b_{cm})^2 \mathbb{P}_{post}(b|D)db \approx \frac{1}{5000} \sum_{j=1}^{5000} (b_j - b_{cm})^2 = 0.01427.$$

With these values we can compute a 95% confidence interval for b. In this case the interval is given by

$$[0.9247042 - 2\sqrt{0.01427}, 0.92470422 + 2\sqrt{0.01427}] = [0.68579, 1.163618].$$

Let us do a short digression about the idea behind Monte Carlo integration. Consider the generic problem of evaluating the n-dimensional integral

$$\int_{\mathbb{R}^n} h(x)\rho(x)dx, \tag{3.10}$$

where ρ is the Lebesgue density of some probability measure $\mathbb{P}$. This means that calculating (3.10) is equivalent to calculating the expected value of h, i.e.

$$\mathbb{E}[h] = \int_{\mathbb{R}^n} h(x)\rho(x)dx.$$

If we have $X_1, \ldots, X_n$ random variables independent with density ρ, then by the strong law of large numbers, the sequence of random variables

$$h_n = \frac{1}{n} \sum_{k=1}^{n} h(X_k),$$

converges to $\mathbb{E}[h]$ [6]. Furthermore if $\mathbb{E}[h^2] < \infty$ we can assess the speed of convergence and the quality of the approximation h_n for $\mathbb{E}[h]$. By the central

limit theorem the sequence of random variables h_n

$$\frac{h_n - \mathbb{E}[h]}{\sqrt{\sigma_n}} \to \mathcal{N}(0,1),$$

where

$$\sigma_n = \frac{1}{n}\sum_{k=1}^{n}(h(X_k) - h_n)^2.$$

This means that the uncertainty in the approximation h_n for $\mathbb{E}[h]$ goes to 0 as $\mathcal{O}(\frac{1}{\sqrt{n}})$. Note that the convergence rate is independent of the dimension of the problem. That is the reason why Monte Carlo integration is used in high dimensional problems, where quadrature methods are prohibitively expensive to implement. In Chapter 4 we are going to apply this method to calculate integrals of real valued functions supported in a seven dimensional space.

The estimate for b in equation (3.9), depends on the choice of the prior. At this point it is unclear how choosing a different prior would give a different estimate for b. To close this chapter we discuss the role that the prior has in inference in the Bayesian Framework.

3.2 The Importance of the Prior

Once again consider problem of estimating the value of the parameter b, whose real value is, as before, 0.925. This time we assume the parameter b can be any real number (not just $0 < b \leq 2$ as before) and the prior distribution for b to be

$$b \sim \mathscr{N}(b^*, \sigma_b^2),$$

where b^* and σ_b are parameters to be set later. With this new prior the formula for the posterior is

$$\mathbb{P}_{post}(b|\mathbf{y}) \propto \underbrace{\exp\left(-\frac{\|\mathbf{y}-\mathbf{G}(b)\|_2^2}{2\sigma^2}\right)}_{\text{Likelihood}} \underbrace{\exp\left(-\frac{(b-b^*)^2}{2\sigma_b^2}\right)}_{\text{Prior}}.$$

To illustrate the role that the prior has in the inference of the value of the parameter given the data $\mathbf{y}$, suppose that

$$b \sim \mathcal{N}(4, 2.5).$$

This prior assumes that, with 95% of confidence, the value of b is in the interval $[1.8, 8.2]$. Clearly, there is a mismatch between the true value of b and the range of values that the prior distribution assigns high probability. Let us evaluate how the posterior distribution for b evolves as we consider more and more experimental data from the measurements of $\tilde{u}$. Figure 3.6 shows how the posterior evolves when we calculate the likelihood with more and more data. The first frame shows the result when only the measurement $\tilde{u}(\mathbf{x}_1; b)$ is taken into account. The second frame when the measurements $\tilde{u}(\mathbf{x}_1; b), \tilde{u}(\mathbf{x}_2; b)$ are taken into account. In each new frame we proceed adding one more measurement to calculate the likelihood.

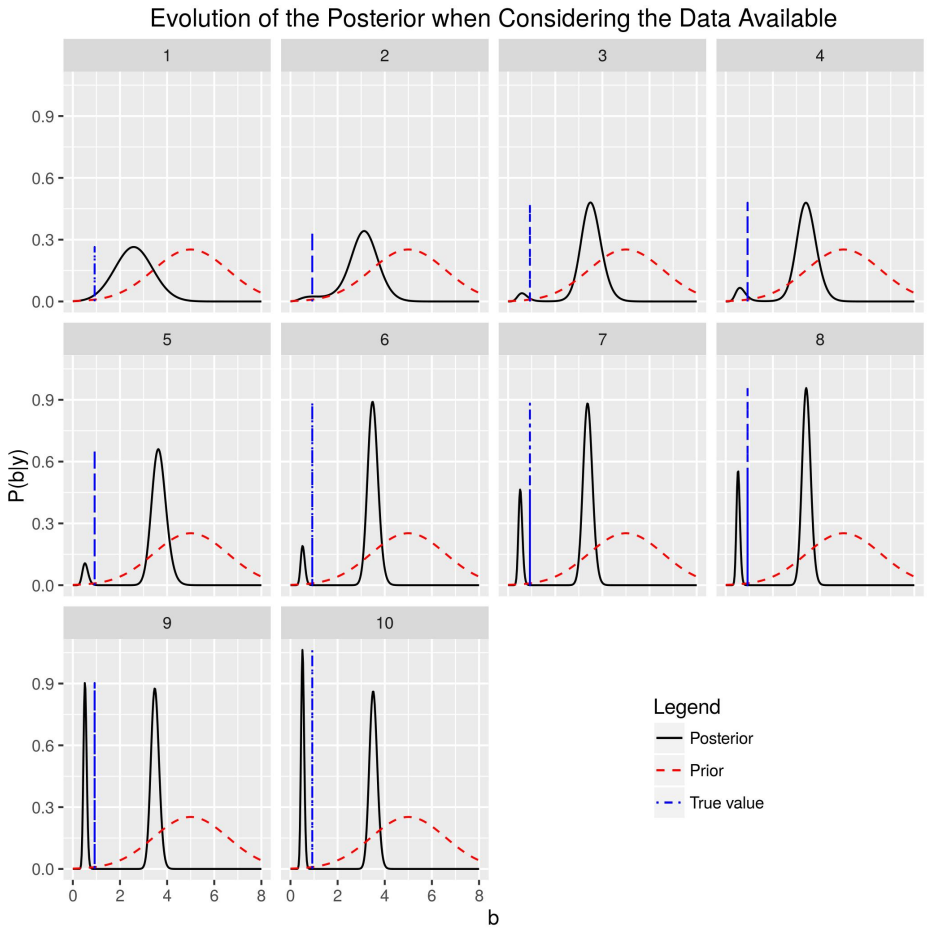


Figure 3.6: Evolution of the posterior distribution when more experimental data is taken into account

The sequence of plots in Figure 3.6 shows that the experimental data creates a new mode in the posterior distribution that is close to the true value of b. In the end of the sequence where we consider all 10 experimental measurements, the mode that is close to the true value of b is bigger than the mode originated by the prior at the point $b = 4$. The explanation for this behavior is that the prior has a high value near $b = 4$, but it is close to zero

for values around $b = 0.925$. Then, when the experimental data is used, the likelihood distribution has a higher value for points close to $b = 0.925$ than points close to $b = 4$. The more data, the higher the value of the likelihood around $b = 0.925$ and closer to zero away from it. However since the prior distribution gives negligible probability to values close to the true value of b, when all data are used the product $\mathbb{P}_{prior}(\mathbf{y}|b)\mathbb{P}(b)$ will be non-negligible only in regions close to $b = 4$ or $b = 0.925$.

The above example is a warning example. If we know how to choose the prior distribution in a way that is meaningful to the problem, reliable inference could be done even with small amount of data. On the contrary if the prior distribution is not realistic, inference could not be done or may not be reliable even with a large amount of data.

Chapter 4

Industrial Case Problem

As mentioned in the introduction, our interested is to study the dispersion of zinc from a lead-zinc smelter in Trail, British Columbia, Canada, operated by Teck Industries Ltd. The smelter has four smokestacks. From now on the smokestacks are going to be referred as source. Our goal is to estimate how much each source is contributing to the total amount of zinc that is being released by the smelter. To this end, we count on experimental measurements of the zinc deposition at nine different locations and wind velocity data of the surrounding area. We also know prior estimates of the zinc release by the smelter. These estimates where calculated by the engineers that work there. An aerial photograph of the region of interest with the location of the sources and the measurement devices is shown in Figure 4.1. The sources are represent by the letters Q_1 to Q_4. The zinc deposition measurement sites are represented by the letter R_1 to R_9.

To obtain the estimates mentioned in the previous paragraph, we are going to proceed in the same manner as in estimating the parameter b in

equation (3.1) in Chapter 3. In that Chapter we were given a mathematical model that approximates the physics of interest. For this chapter, it is necessary to develop, from scratch, a mathematical model that approximates the physics behind pollutant dispersion in the atmosphere. To propose this mathematical model in detail is the topic of the next section.

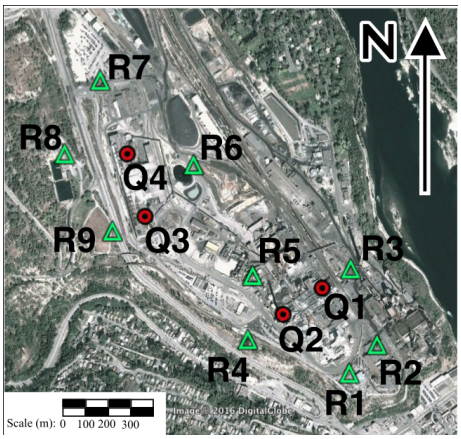


Figure 4.1: Aereal photograph of the lead-zinc smelter in Trail, British Columbia, Canada. The points Q_1 to Q_4 represent the sources of zinc. The green triangles R_1 to R_9 represent the location of the measurement devices.

4.1 A Mathematical Model for Pollutant Dispersion

Our starting point is the conservation of mass. In particular conservation of mass for particulate zinc in the atmosphere. Consider a region $\Lambda \subset \mathbb{R}^3$ with m units of mass of zinc within it. Assume that in the interior of Λ there is a source of zinc. Furthermore assume that zinc is flowing throughout the

boundary $\partial\Lambda$ of Λ due to a wind velocity field (see Figure 4.2). If $\mathbf{v}(\mathbf{x},t)$ represents the wind velocity at a point $\mathbf{x}$ at time t, it can be shown that net mass per unit time, of zinc that is flowing through the boundary is given by the expression [24]

$$\oint_{\partial\Lambda} c(\mathbf{x},t)\mathbf{v}(\mathbf{x},t)\cdot\hat{\mathbf{n}}dA,$$

where $c(\mathbf{x},t)$ is the concentration of zinc at a point $\mathbf{x}$ at time t. The units of c are given in units of mass per units of volume. On the other hand the rate of change total mass m at a time t inside Λ can be calculated as

$$\frac{dm(t)}{dt} = \frac{d}{dt}\int_{\Lambda} c(\mathbf{x},t)dV.$$

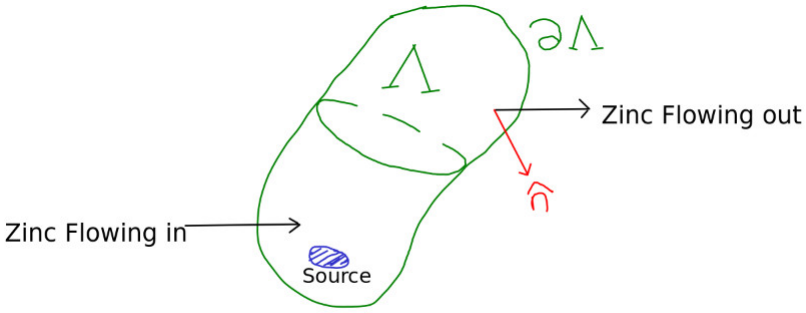


Figure 4.2: Schematic representation of a volume region in space containing a source of Zinc.

Finally, the amount of zinc that comes from the source at a time t is given by

$$\int_{\Lambda} s(\mathbf{x},t)dV,$$

where $s(\mathbf{x},t)$ is the source density. Its units are mass per unit volume per

unit time. Conservation of mass states that the total mass inside Λ should be conserved. Therefore for all times the rate of change of the mass inside m should equal all source of variation of it. Mathematically we can write

$$\frac{d}{dt}\int_{\Lambda} c(\mathbf{x},t)dV = -\oint_{\partial\Lambda} c(\mathbf{x},t)\mathbf{v}\cdot\mathbf{n}dA + \int_{\Lambda} s(\mathbf{x},t)dV.$$

Since we picked the orientation of $\partial\Lambda$ with the normal pointing outwards, it is necessary to put a minus in front of the surface integral for consistency. Assuming the concentration and the velocity field are continuous functions of time and space, an straightforward application of the divergence theorem and Leibniz rule for integral gives

$$\int_{\Lambda}\left(\frac{\partial c(\mathbf{x},t)}{\partial t} + \nabla\cdot(c\mathbf{v}) - s(\mathbf{x},t)\right)dV = 0.$$

Since the region Λ was arbitrary, the previous equality holds if and only if

$$\frac{\partial c(\mathbf{x},t)}{\partial t} + \nabla\cdot(c\mathbf{v}) = s(\mathbf{x},t). \tag{4.1}$$

If we apply this equation to estimate the concentration of zinc using real wind data, we will find that the prediction for the concentration is not completely accurate. The reason is that at small scales, there are random fluctuations in the wind velocity. To model this we follow [24] and write the wind velocity field as

$$\mathbf{v} = \bar{\mathbf{v}} + \mathbf{v}', \tag{4.2}$$

where $\bar{\mathbf{v}}$ is the measured wind velocity and $\mathbf{v}'$ is a random variable with zero mean. If we replace $\mathbf{v}$ in equation (4.1) with the expression for velocity in

equation (4.2) we get

$$\frac{\partial c(\mathbf{x},t)}{\partial t} + \nabla \cdot (c(\bar{\mathbf{v}} + \mathbf{v}')) = s(\mathbf{x},t). \tag{4.3}$$

The presence of the random variable $\mathbf{v}'$ transforms the solution c into a random variable as well. In this case we describe c as the contribution of two terms as

$$c(\mathbf{x},t) := \mathbb{E}(c)(\mathbf{x},t) + c(\mathbf{x},t)', \tag{4.4}$$

where c' satisfies $\mathbb{E}(c(\mathbf{x},t)') = 0$. The intuition behind this definition is the following: if we repeat a large number of times the experiment of measuring the concentration at a point $\mathbf{x}$ and at time t under identical initial and boundary condition, then, we expect the measurements to have an underlying average behavior $\mathbb{E}(c)(\mathbf{x},t)$ plus some noise $c(\mathbf{x},t)'$. Plugging in equation (4.4) into equation (4.3) we obtain

$$\frac{\partial \mathbb{E}(c)}{\partial t} + \nabla \cdot (\bar{\mathbf{v}}\mathbb{E}(c)) + \nabla \cdot (\mathbb{E}(c'\mathbf{v}')) = s(\mathbf{x},t). \tag{4.5}$$

This equation includes the extra variable $\mathbb{E}(c'\mathbf{v}')$. In this case we have two unknowns and one equation. One way to overcome this issue is given by the so-called mixing-length theory. The theory uses the constitutive equation

$$\mathbb{E}(c'\mathbf{v}') = \mathbf{D}\nabla(\mathbb{E}(c)).$$

The term $\mathbf{D}$ is a rank two tensor called the eddy diffusivity tensor. This tensor is assumed to be symmetric, hence is always diagonalizable. We may

assume that we are working on the principal axis of $\mathbf{D}$ [24], hence

$$\mathbf{D} = \begin{bmatrix} D_{xx} & 0 & 0 \\ 0 & D_{yy} & 0 \\ 0 & 0 & D_{zz} \end{bmatrix}.$$

Under the mixing-length theory, equation (4.5) reads as

$$\frac{\partial \mathbb{E}(c)}{\partial t} + \nabla \cdot (\bar{\mathbf{v}}\mathbb{E}(c) + \mathbf{D}\nabla\mathbb{E}(c)) = s(\mathbf{x}, t). \tag{4.6}$$

The variable $\mathbb{E}(c)$ is a deterministic function of space and time. So to make notation lighter, we set $C(\mathbf{x}, t) := \mathbb{E}(c)$. Note that C has the same units as c. We interpret $C(\mathbf{x}, t)$ as the expected concentration we would measure at the point $\mathbf{x}$ at time t.

For the source density we assume point-wise sources. If there are n sources at the points $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, located inside the domain of definition of equation (4.6), then we assume a source density of the form

$$s(\mathbf{x}, t) = \sum_{j=1}^{n} q_j(t)\delta(\mathbf{x} - \mathbf{x}_j). \tag{4.7}$$

Here q_j is referred as the strength of the j-th source and has units of mass per unit time. The function $\delta(\cdot)$ is the Dirac delta function. For our model we have $n = 4$ sources (see Figure 4.1). Taking into account the observations made in the previous paragraph, we finally state the mathematical model for

zinc disperssion as

$$\frac{\partial C(\mathbf{x},t)}{\partial t} + \nabla \cdot (\bar{\mathbf{v}} C(\mathbf{x},t) + \mathbf{D}\nabla C(\mathbf{x},t)) = \sum_{j=1}^{4} q_j(t)\delta(\mathbf{x}-\mathbf{x}_j). \tag{4.8}$$

Due to the presence of the diffusivity tensor, it is necessary to make assumptions about the behavior of the diagonal elements of $\mathbf{D}$. Also the measurements of the wind velocity are given by one 2-axis anemometer. This means we count with just one wind velocity measurement at one point in space. This means it is also necessary to make assumption about the wind velocity distribution.

4.1.1 Assumptions on the Diffusivity Tensor

Following [16], the vertical diffusivity coefficient D_{zz} is approximated by

$$D_{zz} = \frac{\kappa v_* z}{\phi(z/L)}, \tag{4.9}$$

Where κ is the *Von-Karman* constant whose value in practice is set as 0.4. The denominator is defined as the piece-wise continuous function

$$\phi\left(\frac{z}{L}\right) = \begin{cases} 1 + 4.7\frac{z}{L} & \text{for } \frac{z}{L} > 0 \\ 1 & \text{for } \frac{z}{L} = 0 \\ (1 - 15\frac{z}{L})^{-\frac{1}{2}} & \text{for } \frac{z}{L} < 0 \end{cases}.$$

Here L is referred as the *Monin-Obukhov length*. Finally the parameter v_* is known as the *friction velocity*. This parameter is calculated as

$$v_*(t) = \frac{\kappa v_r(t)}{\ln(\frac{z_r}{z_0})},$$

where $v_r(t)$ is a reference velocity at a reference height h_r. The variable z_0 is called the *roughness length*.

For the elements D_{xx} and D_{yy}, we assume $D_{xx} = D_{yy}$ and independence of height [16]. A common used equation to approximate these variables is given by

$$D_{xx} = D_{yy} \approx \frac{v_* z_i^{\frac{3}{4}} (-\kappa L)^{-\frac{1}{3}}}{10}.$$

The variable z_i is known as the *mixing layer height*.

4.1.2 Assumptions on the Wind Velocity Distribution

In practice, the wind velocity measurements are done using anemometers that measure a two dimensional projection of the velocity vector field. Therefore it is necessary to make assumptions on the three dimensional and global behavior of this vector field. Following [7], we assume a velocity vector field of the form

$$\mathbf{v} = (v_x(z,t), v_y(z,t), v_{set}). \tag{4.10}$$

Observe it is assumed the wind velocity field is independent of x and y. In equation 4.10, v_{set} is a constant given by the settling velocity of the zinc particles. By Stoke's law, this velocity is given by

$$v_{set} = \frac{\rho g d^2}{18\mu},$$

where ρ is the particle density, g the acceleration of gravity, d its diameter, and μ is the air viscosity. For the x, y components of $\mathbf{v}$ we assume a power law relation of the form

$$\|(v_x(z,t), v_y(z,t))\|_2 = v_r(t)\left(\frac{z}{z_r}\right)^p, \tag{4.11}$$

where $v_r(t)$ is the wind speed at a reference height z_r. The exponent p depends on factors such as the surface roughness and atmosphere stability. For more details in the power law model for the wind velocity the reader is referred to [24].

Observe that the diffusivity coefficient in equation (4.9) vanishes at the ground level. This is inconsistent with the boundary condition in equation (4.12). Hence, we assume the diffusivity coefficient to be non-zero below a *cutting height* z_{cut}.

Assuming an specific form for the wind velocity and the diffusivity tensor, introduces new parameters whose values need to be set. In our model there are five parameters we need to assign a value before we can use equation (4.8). These parameters are

- p: the fitting parameter for the wind velocity power law.
- z_0: roughness length
- z_i: mixing layer height.
- L: Monin-Obukhov length.
- z_{cut} cutting height.

In practice we set a value for these parameters from a given interval of numbers that empirically has shown to work [7,24]. The caveat with this approach is that there is no good reason to choose one value over a different one. In this work we will use the Bayesian framework to decide the values of these parameters and the sources as well.

To completely specify the model for the zinc pollutant dispersion, it is necessary to specify the domain of interest, the boundary and initial conditions in equation (4.8). The spatial domain of interest is given by

$$\Pi := \{(x, y, z) \in \mathbb{R}^3 | z \geq 0\}.$$

Following [7], the boundary conditions are given by the far-field condition

$$C(\mathbf{x}, t) \to 0 \qquad \text{as } \|x\| \to \infty,$$

and the Robin boundary condition

$$\left(v_{set} C + D_{zz} \frac{\partial C}{\partial z} \right) \Big|_{z=0} = v_{dep} C|_{z=0} \,. \tag{4.12}$$

To estimate the parameters p, z_0, z_i, L, z_{cut} and the four sources sources, we count on experimental measurements of the zinc deposition at the sites $R_1, \ldots, R_9$ (see Figure 4.1) over one month period. The mathematical model described so far, approximates the physics of the concentration, not the deposition. Thus is necessary to make the connection between the solution C

of equation (4.8) and the deposition of zinc at $z = 0$. If v_{set} is the settling velocity of zinc particles, it is not hard to see that the deposition per unit area at a point $(x, y, 0) \in \Pi$ during the interval of time $(0, T]$ is given by

$$w(x, y, T) = \int_0^T C(x, y, 0, t) v_{set} dt. \tag{4.13}$$

Since the nine measurements $R_1, \ldots, R_9$ were obtained by the placement of identical dust-fall jar collectors with non-zero, but small cross-sectional area, we can readily approximate the total deposition during the interval $(0, T]$ at the i-th site as

$$W(x_i, y_i, T) = \int_{dust-fall} w(x, y, T) dx dy \approx w(x_i, y_i, T) \Delta A,$$

where ΔA is the cross-sectional area of the dust-fall jar. To make notation lighter we define

$$M_i = W(x_i, y_i, T) \qquad \text{for } i = 1, \ldots, 9. \tag{4.14}$$

The scalar M_i is the zinc deposition at the site R_i. From now on we assume T to be one month.

From equations (4.13) and (4.14), we infer that the map that takes concentration into deposition is linear. Also from equation (4.8) it is straightforward to check that the concentration C is related to the source $s(\mathbf{x}, t)$ by a linear partial differential operator. Since composition of linear maps is linear, we conclude that the map that takes sources to deposition is linear. Since we have four different sources, which we denote by q_1, q_2, q_3, q_4, we may

write

$$M_i = \sum_{j=1}^{4} a_{ij}(p, z_0, z_i, L, z_{cut})q_j \qquad \text{for } i = 1, \ldots, 9. \tag{4.15}$$

The coefficients a_{ij} capture the dependence of the deposition with respect to the parameters p, z_0, z_i, L, z_{cut}, which in general is non-linear. If we define the vectors

$$\begin{aligned} \mathbf{y} &= \begin{bmatrix} M_1 & \ldots & M_9 \end{bmatrix}^T, \\ \mathbf{q} &= \begin{bmatrix} q1 & q_2 & q_2 & q_4 \end{bmatrix}^T, \end{aligned}$$

we can write equation (4.15) more compactly as

$$\mathbf{y} = A(p, z_0, z_i, L, z_{cut})\mathbf{q}. \tag{4.16}$$

Here A is a 9×4 matrix whose coefficients are $a_{ij}(p, z_0, z_i, L, z_{cut})$.

Equation (4.16) models the relation between deposition and all parameters involved in equation (4.8). However we do not know an expression for the 36 coefficients a_{ij} as a function of $(p, z_0, z_i, L, z_{cut})$. To find such expression it is necessary to solve equation (4.8). There is no known closed form of this equation, hence a numerical approximation is necessary. If we had unlimited computational or time budget we could solve equation (4.8) for as many different configurations of the parameters $(p, z_0, z_i, L, z_{cut})$ as we want and use these results to approximate the coefficients a_{ij}. Clearly this is not feasible. A more realistic approach is to solve equation (4.8) for a number of different configurations of $(p, z_0, z_i, L, z_{cut})$ that does not exceed our compu-

tational and time budget. Then use Gaussian processes to interpolate at the configurations we did not solve. Before we proceed to use Gaussian processes for the interpolation let us see if is possible to reduce the dimensionality of the parameter space. To this end, we do sensitivity analysis, as explained in Chapter 2, Section 2.1.3.

4.2 Sensitivity Analysis

Our starting point is to consider the set of maps $\{\varphi_{x,y}\}_{x,y\in\mathbb{R}^2}$. The input for each map is a point $(p, z_0, z_i, L, z_{cut})$ in the parameter space and the output is the deposition value at the point (x, y). The domain of each map is the set of allowed values for the parameters. The range of allowed values for the set of parameters is shown in Table 4.1 [24].

Parameter (units)	Symbol	Range
Velocity exponent	p	$[0.1, 0.4]$
Roughness length (m)	z_0	$[10^{-3}, 2]$
Height of mixing layer (m)	z_i	$[10^2, 3 \times 10^3]$
Monin-Obukhov length (m)	L	$[-500, -1]$
Cut-off length (m)	z_{cut}	$[1, 5]$

Table 4.1: Parameters under study and their accepted ranges

We perform a sensitivity analysis as explained in Chapter 2, Section 2.1.3, on the set of maps $\{\varphi_{x_i,y_i}\}_{i=1}^{9}$, where (x_i, y_i) represent the location R_i in Figure 4.1. To make computations simpler we map bijectively the set of ranges of the parameters into the five-dimensional unit hypercube $[0, 1]^5$.

Thus, without lost of generality we may assume the deposition maps are of the form

$$\varphi_{x_i,y_i} : [0,1]^5 \subset \mathbb{R}^5 \to \mathbb{R} \qquad \text{for } i = 1, 2, \ldots, 9.$$

Recall from Chapter 2, Section 2.1.3, that in order to estimate Sobol indices is necessary to perform numerical integrations over the integrands $\{\varphi_{x_i,y_i}\}_{i=1}^{9}$. These integrals are five-dimensional, hence a quadrature method is not feasible. It is necessary to use Monte Carlo integration. In order to implement it, is necessary to know the output of $\{\varphi_{x_i,y_i}\}_{i=1}^{9}$ at any point in their domain. In general we do not know the output of the function of interest beyond a very limited number of points. To extrapolate any function from the set $\{\varphi_{x_i,y_i}\}_{i=1}^{9}$ at points in its domain where we do not know the output, we use an emulator using Gaussian processes as explained in Chapter 2, Section 2.1.1.

Implementing the routines to emulate and estimate Sobol indices is a time consuming task. In order to optimize our time budget, we used the R packages DiceKriging and Sensitivity [20,22] for the emulator and to estimate the total effect Sobol index for each of the maps $\{\varphi_{x_i,y_i}\}_{i=1}^{9}$. The DiceKriging package allows us to use five different kernels for the emulation. In order to consider the influence in using different kernels for the estimation the Sobol indices, we calculate the indices five times, one for each different kernel.

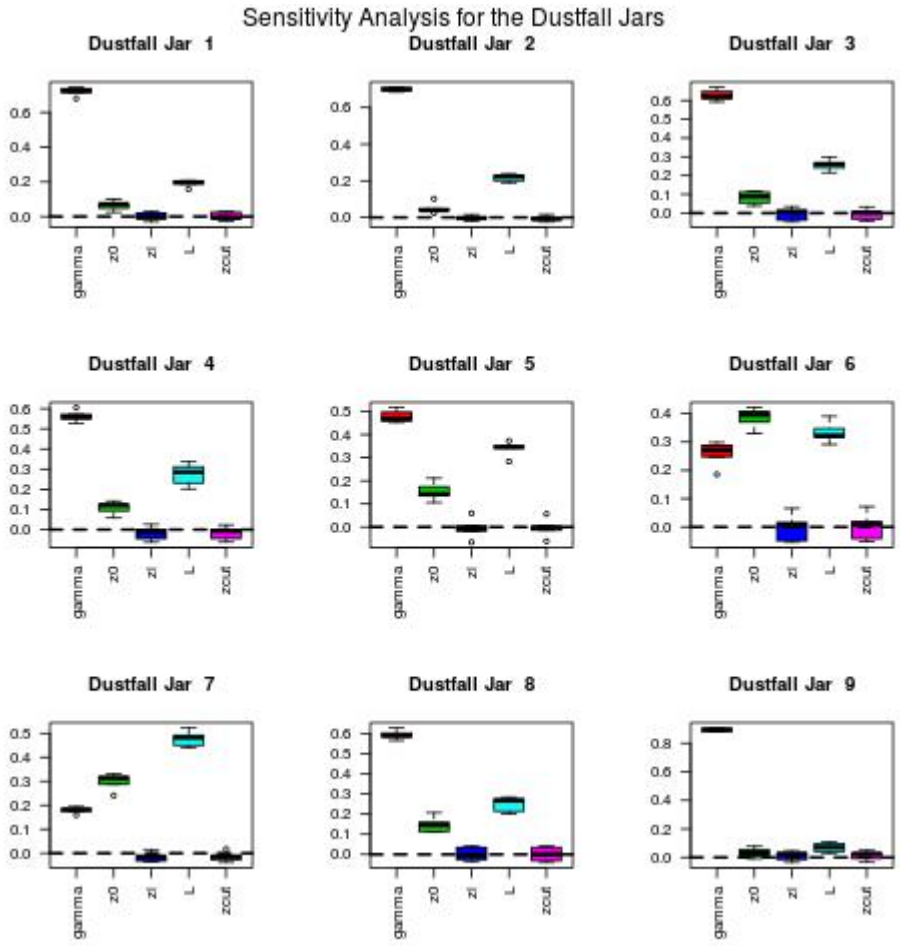


Figure 4.3: Boxplots containing the result for the total effect Sobol index performed on each of the nine sensors. The dashed line represents the value of zero.

Figure 4.3 shows that two of the five variables do not account for the variance in the deposition, namely, z_i and z_{cut}. Therefore we can reduce the dimensionality of the parameter space from five to three. In this case we can

write equation (4.16) as

$$\mathbf{y} = A(p, L, z_0)\mathbf{q}.$$

From now on we use the convention that the parameters z_i and z_{cut} are fixed at the values 100 and 2 respectively (see [7]).

In order to obtain a closed expression for the matrix A it is necessary to solve equation (4.8) along with its boundary condition. There is no known close form to this equation, hence is necessary to use numerical methods to solve it. Then, use these results to obtain information about A. Solving equation (4.8) is a computationally intensive task. Is not feasible to solve it for a wide range of different values of (p, L, z_0). Instead, to learn about A we choose an small number of different values of the parameters (p, L, z_0), and for the rest of the points we will use an emulator.

4.3 Building an Emulator for $A(p, L, z_0)$

To solve numerically equation (4.8), we used a finite volume code. The details of the implementation are given in [7]. Running the volume code on an $30 \times 30 \times 30$ grid takes about 30 minutes. Hence the necessity to use Gaussian process regression to create an emulator as explained in Chapter 2, Section 3.1.2.

To construct the emulator we need a training set. For that end we create an space filling design for the parameter space (p, L, z_0), as explained in Chapter 2, Section 2.1.2. In particular we will use a maximin design. Finding a maximin design is an optimization problem, that is analytically challeng-

ing to solve, hence is necessary to use numerical approximations. For the optimization we used the particle swarm algorithm. The reader interested in this algorithm is referred to [3].

Considering our time and computational budget, we chose to do the experimental design with $n = 64$ points. The maximin design obtained by using a particle swarm for the optimization is shown in Figure 4.4.

With the experimental design completed, the next step is to run the finite volume code on each of the different 64 configurations of parameters. We repeat this step four times. The first time we run the 64 simulations by fixing the first source q_1 to 1 and the other three sources to zero. The second time we fix the second source q_2 to 1 and the other three sources to zero. For the third and fourth step we fix the third and fourth source to one respectively, and the rest of the sources to zero. The reason for this implementation is that we need to connect the output of the finite volume code, which is the deposition vector $\mathbf{y}$ at the nine sites $R_1, R_2, \ldots, R_9$ with the components of the matrix $A(p, L, z_0)$. Recall that the relation between these two quantities is given by equation (4.16), explicitly

$$\mathbf{y} = A(p, L, z_0) \underbrace{\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix}}_{\mathbf{q}}, \tag{4.17}$$

For example if we run the finite volume code with the parameters set at the values p^*, L^*, z_0^* with the i-th source set to one and the other three to

zero, obtaining the output

$$\mathbf{y}^* = \begin{bmatrix} y_1^* \\ \vdots \\ y_9^* \end{bmatrix}.$$

Then from equation (4.17) it is easy to see that the following equality holds

$$\begin{bmatrix} y_1^* \\ \vdots \\ y_9^* \end{bmatrix} = \begin{bmatrix} a_{1i}(p^*, L^*, z_0^*) \\ \vdots \\ a_{9i}(p^*, L^*, z_0^*) \end{bmatrix}$$

where the RHS is the i-th column of A. This result shows why it is necessary to run four times each of the 64 simulations to obtain the training set for the emulator of A. The emulator for A will be represented by $\widehat{A}$. The construction of $\widehat{A}$ is obtained by using Gaussian processes as explained in Chapter 2, Section 2.1.1, and exemplified in Chapter 3 Section 3.1.2. By construction, at the points in the maximin design in Figure 4.4, A and $\widehat{A}$ coincide. To account for the discrepancies outside this set of points we assume an additive Gaussian noise model between deposition, parameters and sources, that is (cf. Chapter 3, Section 3.1.2)

$$\mathbf{y} = \widehat{A}(p, L, z_0)\mathbf{q} + \epsilon, \tag{4.18}$$

with $\epsilon \sim \mathcal{N}(0, \lambda I_{9\times 9})$. The value of the parameter λ is fixed in a way to get a signal to noise ratio with the experimental data of 1 to 10. The value that fulfills this condition is $\lambda = 5.68 \times 10^{-5}$. In equation (4.18) the vector $\mathbf{y}$ could represent either the output of the finite volume code at the dustfall

jars positions or the experimental deposition measures. This interpretation of $\mathbf{y}$ implies that the random variable ϵ, also accounts for the discrepancy between simulation of equation (4.8), the physics and experimental measurement errors.

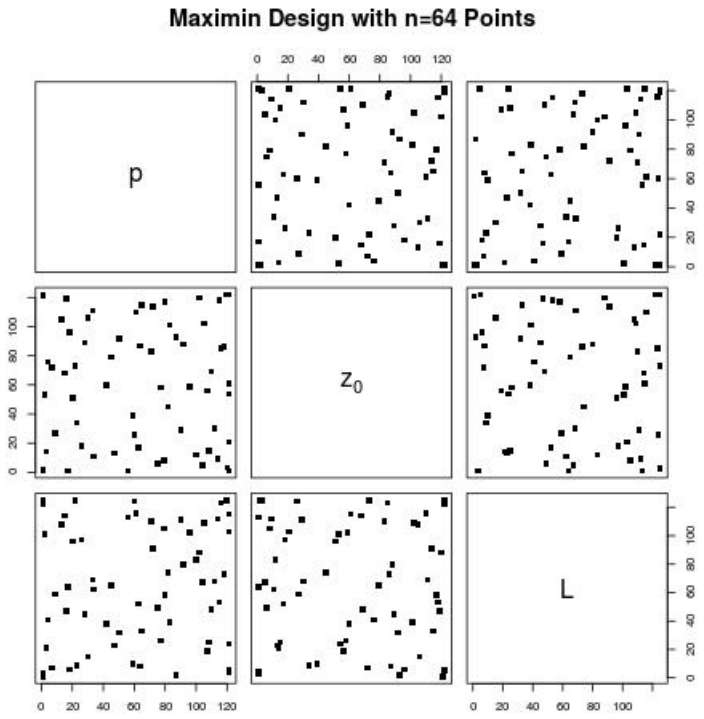


Figure 4.4: Maximin design with 64 points in the parameter space (p, L, z_0).

After constructing the emulator, we are now in position to apply the Bayesian framework to estimate the values for the parameters and the sources. This is going to be done in the next section.

4.4 Bayesian Framework for the Inverse Problem

Recall that in the Bayesian framework, to solve an inverse problem is equivalent to find the posterior distribution of the parameters of interest. In our case we want to find the posterior of the parameters p, z_0, L and sources $\mathbf{q}$ given the experimental data $\mathbf{y}$. By Bayes' rule the posterior distribution is given by

$$\mathbb{P}_{post}(p, z_0, L, \mathbf{q}|\mathbf{y}) = \frac{\mathbb{P}_{like}(\mathbf{y}|p, z_0, L, \mathbf{q})\mathbb{P}_{prior}(p, z_0, L, \mathbf{q})}{Z(\mathbf{y})}. \tag{4.19}$$

Given the Gaussian additive noise model in equation (4.18), the likelihood can be calculated as

$$\mathbf{y}|p, z_0, L, \mathbf{q} \sim \mathcal{N}(\widehat{A}\mathbf{q}, \lambda I).$$

or explicitly

$$\mathbb{P}_{like}(\mathbf{y}|p, z_0, L, \mathbf{q}) = \frac{1}{(2\pi\lambda^2)^{\frac{9}{2}}} \exp\left(-\frac{1}{2\lambda^2}\|\widehat{A}q - \mathbf{y}\|^2\right). \tag{4.20}$$

The constant of proportionality is given by

$$Z(\mathbf{y}) = \int \mathbb{P}_{like}(\mathbf{y}|p, z_0, L, \mathbf{q})\mathbb{P}_{prior}(p, z_0, L, \mathbf{q}) dp dz_0 dL d\mathbf{q}. \tag{4.21}$$

To find it, is necessary to choose a prior distribution. We now turn our attention into choosing a prior that captures the better the information we have about the problem.

4.4.1 Choosing a Prior

The values of the parameters $p, z_0.L$ depend on the environmental conditions of the region where the lead-zinc smelter is, whereas the values of the sources q_1, q_2, q_3, q_4 do not. Thus, it is reasonable to assume that the possible values between these two sets of variables are independent of each other. This can be captured by the assumption

$$\mathbb{P}_{prior}(p, z_0, L, \mathbf{q}) = \mathbb{P}_{prior}(p, z_0, L)\mathbb{P}_{prior}(\mathbf{q}).$$

By writing the prior distribution as the product of two distributions, we simplify the problem by working with two different subsets of variables. Let us first work with the prior for the parameters $p, z_0.L$. In [24] it is propose a function relation of the form

$$L = a + b\ln(z_0),$$

where a and b are constants. This relation is empiric and its validity is yet to be confirmed. Instead we assume that z_0 and L are independent. Finally there is no known connection between the parameter p and the other two parameters. Thus we assume all three parameters are independent of each other. This assumption is modeled by

$$\mathbb{P}_{prior}(p, z_0, L) = \mathbb{P}_{prior}(p)\mathbb{P}_{prior}(z_0)\mathbb{P}_{prior}(L).$$

As mentioned in Chapter 1, there is no strong reason to pick a value over the other for these parameters. Since there is no preference, we assume

an uniform distribution for each parameter. The allowed range for each parameter is given in Table (4.1). Explicitly we have the distribution density for each parameter as

$$\begin{aligned} \mathbb{P}_{prior}(p) &= \frac{1}{0.3}\mathbf{1}_{[0.1,0.4]} \\ \mathbb{P}_{prior}(z_0) &= \frac{1}{2-10^{-3}}\mathbf{1}_{[10^{-3},2]}, \\ \mathbb{P}_{prior}(L) &= \frac{1}{499}\mathbf{1}_{[-500,-1]}. \end{aligned} \tag{4.22}$$

Choosing the prior for the sources q_1, q_2, q_3, q_4 requires more analysis since we are not completely ignorant about their possible values. Let us summarize our knowledge about these variables.

1. Each source is unrelated to the other three.

2. $q_i > 0$ and $q_i < \infty$ for $i = 1, 2, 3, 4$.

3. According to the engineers working at the smelter, the estimated output for the sources are

Source	Estimated Emission Rate [ton/yr]
q_1	35
q_2	80
q_3	5
q_4	5

Table 4.2: Estimated parameters for the four sources.

4. The engineers estimates are reliable.

Mathematically the first condition can be written as

$$\mathbb{P}_{prior}(q_1, q_2, q_3, q_4) = \mathbb{P}_{prior}(q_1)\mathbb{P}_{prior}(q_2)\mathbb{P}_{prior}(q_3)\mathbb{P}_{prior}(q_4).$$

The second condition requires that the probability density for each source has to be supported in the set $[0, \infty)$. The third and fourth conditions can be interpreted as follows: the mode of the prior for each sources has to be at the engineers estimated value and 99% of the mass should be contained between 0 and 3 times that value. A probability distribution that satisfies the above conditions is the gamma distribution. The gamma distribution is a parametric distribution. It will be denoted by $\mathscr{G}(\alpha, \beta)$. The Lebesgue density for the gamma distribution is given by

$$f(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x} \mathbf{1}_{[0,\infty)},$$

where α and β are constants. We are going to assume that each source has a gamma distribution. In this case we write

$$q_i \sim \mathscr{G}(\alpha_i, \beta_i) \text{ for } i = 1, 2, 3, 4,$$

or equivalently

$$\mathbb{P}_{prior}(q_i) = \frac{\beta_i^{\alpha_i}}{\Gamma(\alpha_i)} q_i^{\alpha_i - 1} e^{-\beta_i q_i} \mathbf{1}_{[0,\infty)}. \tag{4.23}$$

We choose the values of α_i and β_i in terms of the engineer estimate $q_{eng,i}$ for the i-th source. More precisely we choose the values of α_i and β_i as the

solution of the following system of equations

$$\frac{\alpha_i - 1}{\beta_i} = q_{eng,i},$$
$$qgamma(0.99, \alpha_i, \beta_i) = 3q_{eng,i}.$$

Here *qgamma* is the quantile function for the gamma distribution. By choosing the values of the parameters for the gamma distribution in this manner, we satisfy

$$\max_{q\in[0,\infty)} f(q|\alpha_i, \beta_i) = q_{eng,i} \qquad \text{for } i = 1, 2, 3, 4.$$

and 99% of the mass of the density is concentrated between 0 and 3 times the engineers estimate. Combining the results from equations (4.22) and (4.23) we conclude that the prior distribution satisfies

$$\mathbb{P}_{prior}(p, z_0, L, \mathbf{q}) \propto \mathbf{1}_{[0.1,0.4]}\mathbf{1}_{[10^{-3},2]}\mathbf{1}_{[-500,-1]} \prod_{i=1}^{4} q_i^{\alpha_i - 1} e^{-\beta_i q_i} \mathbf{1}_{[0,\infty)}. \qquad (4.24)$$

With the prior and the likelihood, we finally obtain the expression for the posterior by combining equations (4.20),(4.21) and (4.24) into Bayes' rule in equation (4.19). Explicitly, the posterior $\mathbb{P}_{post}(p, z_0, L, \mathbf{q}|\mathbf{y})$ is proportional to

$$\exp\left(-\frac{1}{2\lambda^2}\|\widehat{A}\mathbf{q} - \mathbf{y}\|^2 - \sum_{i=1}^{4} \beta_i q_i\right) \prod_{j=1}^{4} q_i^{\alpha_i - 1} \mathbf{1}_{[0.1,0.4]\times[10^{-3},2]\times[-500,-1]}.$$

The indicator function represents the ranges of the allowed values of the parameters in Table (4.1). These values are widely accepted in the literature, however there is not a technically sounded reason of why these ranges are

acceptable. We will expand the possible values for these parameters in order to test the validity of the values in Table (4.1). The new set of ranges we picked are shown in Table 4.3.

Parameter (units)	Symbol	Range
Velocity exponent	p	$[0, 0.6]$
Roughness length (m)	z_0	$[0, 3]$
Monin-Obukhov length (m)	L	$[-600, 0]$

Table 4.3: Parameters under study and their new allowed ranges.

Using these new ranges for the parameters the posterior $\mathbb{P}_{post}(p, z_0, L, \mathbf{q}|\mathbf{y})$ is now proportional to

$$\exp\left(-\frac{1}{2\lambda^2}\|\widehat{A}\mathbf{q}-\mathbf{y}\|^2-\sum_{i=1}^{4}\beta_i q_i\right)\prod_{j=1}^{4} q_i^{\alpha_i-1}\mathbf{1}_{[0,0.6]\times[0,3]\times[-600,0]}. \tag{4.25}$$

This equation is the solution for the inverse problem. However the formula for the posterior is not of much use. It is necessary to extract the information contained in it. In order to extract such information we can obtain point estimates of the parameters and the uncertainty associated to those estimates. Finding these estimates is the topic of the next section.

4.5 Inferring the Parameters and the sources

To find point estimates such as those in equation (2.8) and its uncertainty associated, is necessary to perform high dimensional integrals that are not an-

alytically tractable. Thus we resort to use numerical integration techniques. In particular we will use use the Metropolis-Hastings(MH) algorithm (see Chapter 3, Algorithm 1) to sample from the posterior and then use Monte Carlo integration to obtain the quantities of interest. For the MH implementation we use one million samples. As a burn in period we discarded the first 500.000 samples. The size of the random step (value of α in step 3 in Algorithm 1) in each dimension was chosen in a way the acceptance rate was around 25%. Using MH to sample from highly dimensional supported probability distributions is challenging. Specially assessing precisely when the Markov chain has converged and if the samples obtained after the convergence are indeed from the probability distribution of interest. To overcome these difficulties, several heuristics on convergence criteria have been develop, such as, graphical methods and non-parametric tests of stationarity. The reader interested in this topic is referred to [5] and the references within. In this work we are going to use the traces of the Markov chain to assess the convergence of it. A *trace plot* is a graphic display of the motion of the Markov chain in each of the dimensions in the support of the probability density we are sampling.

The probability density we are interested in sampling from, is the posterior in equation (4.25). This posterior is supported in a compact subset of $\mathbb{R}^7$, thus to obtain the trace plots of it, we need to plot the motion of the Markov chain in each dimension. These plots are shown in Figure 4.5. The trace plots in Figure 4.5 show that the Markov chain is moving around the whole support of the posterior density. Another observation is every region in the support of $\mathbb{P}_{post}(p, z_0, L, \mathbf{q}|\mathbf{y})$ is visited by the chain every so often.

This means that the chain is not getting stuck in a local mode of the distribution. In conclusion the trace plots have the behaviour One would expect of a Markov chain that has converged and whose realizations are taken from the probability density that is being sampled from.

One of the advantages of using the Metropolis-Hastings algorithm is that the samples obtained in each dimension are distributed as the marginal distribution for that variable. If X and Y are random variables jointly distributed as $\mathbb{P}(X, Y)$, then, the *marginal distribution* of the random variable X is given by

$$\mathbb{P}(X) = \int \mathbb{P}(X, Y) dY.$$

For example the marginal distribution for the parameter p is given by

$$\mathbb{P}(p|\mathbf{y}) = \int \mathbb{P}_{post}(p, z_0, L, \mathbf{q}|\mathbf{y}) dz_0 dL d\mathbf{q}.$$

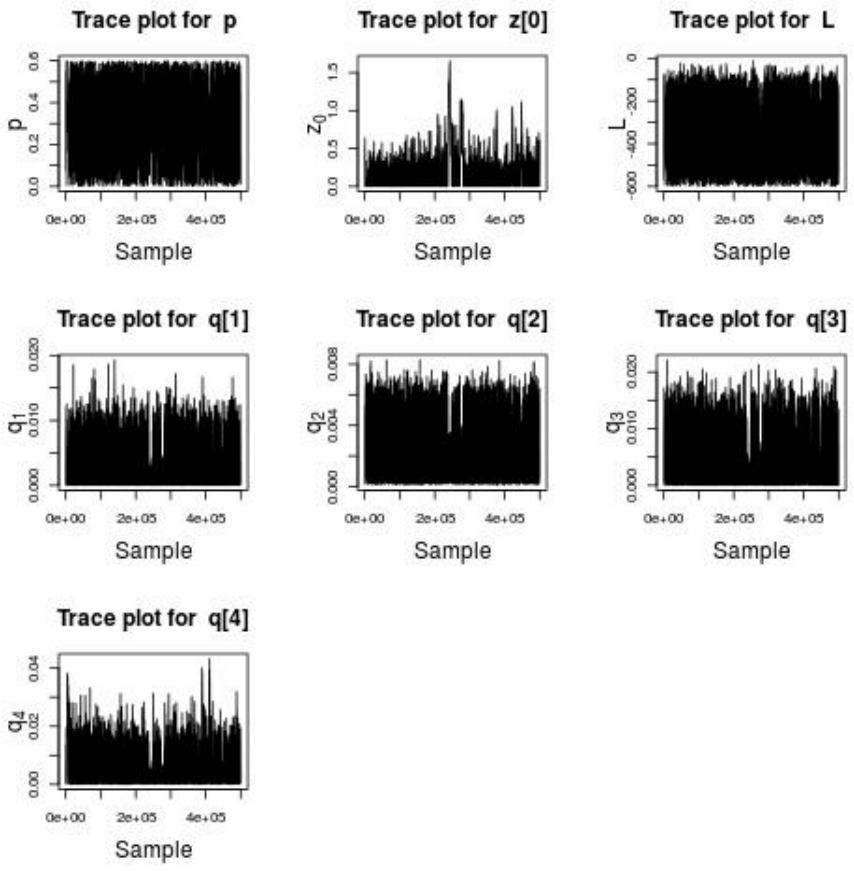


Figure 4.5: Trace plots for each of the variables p, z_0, L and $\mathbf{q}$.

The marginal distribution of a random variable contains all the information about that variable regardless of the value of other parameters. We are going to use the marginals for each of the variables in the posterior distribution in equation (4.25) to obtain useful statistics about the parameters. The histograms for the marginals of each of the variables of interest in the posterior are shown in Figure 4.6.

The next step for estimating the parameters is to decide what of the point

estimates from equation (2.8) we will choose. By looking at Figure 4.6, it is clear that the marginal posterior for the four sources is skewed and with a well-defined mode. In this case, finding the maximum marginal posterior estimate is a reasonable choice. To find the marginal maximum a posteriori estimate for each of the four sources, it is necessary to find

$$\max_{q_j \in [0,\infty)} f(q_j) \qquad \text{for } j = 1, 2, 3, 4,$$

where

$$f(q_j) = q_j^{\alpha_j - 1} \exp(-\beta q_j) \int \prod_{k \neq j} q_k^{\alpha_k - 1} \exp(-\beta_k q_k) \exp\left(-\frac{1}{2\lambda^2} \|\widehat{A}\mathbf{q} - \mathbf{y}\|^2 \right) dp dz_0 dL d\tilde{\mathbf{q}}_j,$$

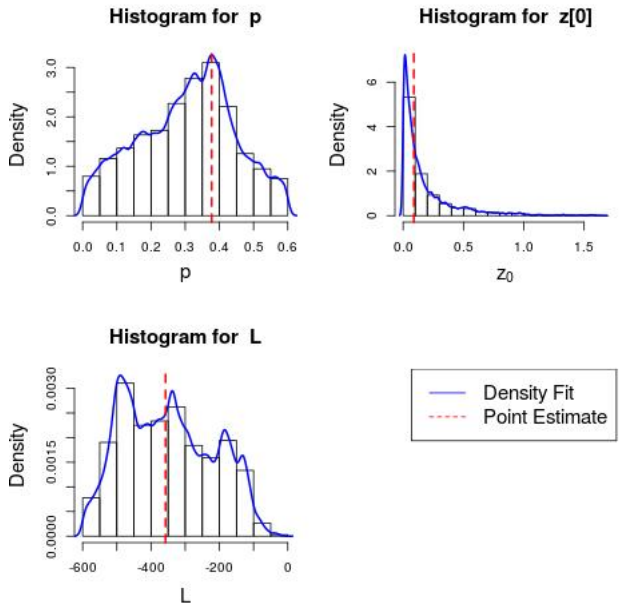
where $d\tilde{\mathbf{q}}_j$ means the j-th term in the volume element $\prod_{k=1}^{4} dq_k$ has been suppressed. Optimizing numerically f is a challenging problem. First, the integral involved is high dimensional, hence is necessary to use Monte Carlo integration to estimate it. The exponent in the integrand is a number close to zero unless the term $\|\widehat{A}\mathbf{q} - \mathbf{y}\|^2 = \mathcal{O}(10^{-12})$. Since Monte Carlo integration requires to evaluate the integrand at a large and random number of different points in the domain of integration, we are going to end up adding numbers that are, in general, smaller than the machine epsilon. Hence, numerically the integrand will behave like a very small constant. The problem with this is that all the information from the experimental data is contained in the integrand. So, if the integrand behaves like a constant due to rounding error, the optimization routine will not return an accurate value. Instead of using an optimization scheme, we will use the fact that the marginal posterior for the sources behaves similarly to a gamma distribution. In this case, we fit

the gamma distribution that is closer to the marginal posterior and report the statistics associated to that distribution.

For the parameters p, z_0, L the situation is more subtle. For the parameter p the distribution is unimodal. In this case we can choose the mode as a point estimate. The way to find the mode is by fitting a density over the histogram of p and pick the maximum of the fitting curve. For the parameters z_0 and L, there is no distinctive point in their distribution. In this case we pick a non-committal value like the median. The median of a one-dimensional probability density ρ is defined as the number m^* such that

$$\int_{-\infty}^{m*} \rho(x)dx = \int_{m^*}^{\infty} \rho(x)dx = 0.5.$$

In Figure 4.6 it is shown the values of the point estimates for each distribution and the fitting curves for the parametes p, z_0, L and the sources q_1 to q_4.


Histogram for p
Density
p
Histogram for z[0]
Density
z_0
Histogram for L
Density
L
Density Fit
Point Estimate

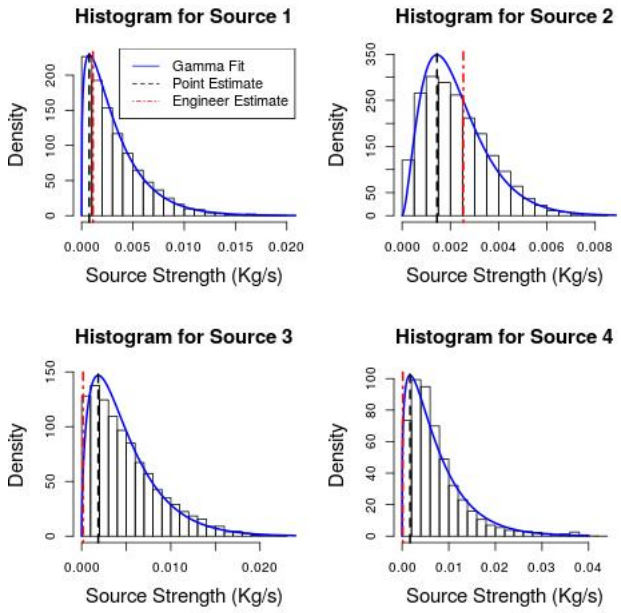


Figure 4.6: Histograms for the marginal distribution for each of the seven variables in the support of the posterior density in equation (4.25).

For the uncertainty estimate, we pick the 68% Bayesian confidence interval for each of the parameters. Given a point estimate x^* for a probability density ρ, a 68% Bayesian confidence interval of that estimate, is defined as the ball of radius r centered at x^* such that

$$\int_{B(x^*,r)} \rho(x)dx = 0.68,$$

The estimates for the parameters and the uncertainties are shown in Table 4.4.

Parameter	Point Estimate	Value	68% Confidence Interval
p	Mode	0.3775	$[0.2197, 0.5352]$
z_0	Median	0.0886	$[0.0024, 0.1748]$
L	Median	-357.60	$[-651, 11, -64.090]$
$q_1[ton/year]$	Mode	23.405	$[0, 120.27]$
$q_2[ton/year]$	Mode	45.544	$[4.438, 86.649]$
$q_3[ton/year]$	Mode	58.669	$[0, 185.74]$
$q_4[ton/year]$	Mode	53.789	$[0, 250]$

Table 4.4: Parameters and their estimates and uncertainties. The sources are given in Tons per year.

The results in Table 4.4, show that sources 3 and 4 are understimated by the engineers, whereas sources 1 and 2 are overstimated. According to the engineer estimates, the smelter produces 125 tons per year. According to our results, this number is small compared to the most likely value of 181.407 tons per year. Another interesting result is that according to the literature, the accepted ranges for the parameters p and L are $[0, 0.4]$ and $[-500, 1]$, respectively. Our results show that these allowed ranges can be stretched in order for the model the explain the experimental data available.

Bibliography

[1] Robert J Adler. The geometry of random fields. 1981. *John Wiley&Sons, Chichester*.

[2] Vladimir Igorevich Arnol'd. *Mathematical methods of classical mechanics*, volume 60. Springer Science & Business Media, 2013.

[3] Rajesh Kumar Arora. *Optimization: algorithms and applications*. CRC Press, 2015.

[4] Alberto Bressan. *Lecture Notes on Functional Analysis*. American Mathematical Society, 1900.

[5] George Casella. Monte carlo statistical methods. 2008.

[6] Richard M Dudley. *Real analysis and probability*, volume 74. Cambridge University Press, 2002.

[7] Bamdad Hosseini and John M Stockie. Airborne contaminant source estimation using a finite-volume forward solver coupled with a bayesian inversion approach. *arXiv preprint arXiv:1607.03518*, 2016.

[8] Edwin T Jaynes. *Probability theory: The logic of science*. Cambridge university press, 2003.

[9] Mark E Johnson, Leslie M Moore, and Donald Ylvisaker. Minimax and maximin distance designs. *Journal of statistical planning and inference*, 26(2):131–148, 1990.

[10] Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

[11] Leonid P Lebedev, Iosif I Vorovich, and Graham Maurice Leslie Gladwell. *Functional analysis: applications in mechanics and inverse problems*, volume 41. Springer Science & Business Media, 2012.

[12] Nicolas Lerner et al. *A Course on Integration Theory*. Springer, 2014.

[13] Mikhail Lifshits. *Lectures on Gaussian processes*. Springer, 2012.

[14] Mikhail Anatolevich Lifshits. *Gaussian random functions*, volume 322. Springer Science & Business Media, 2013.

[15] J David Logan. *Applied partial differential equations*. Springer, 2014.

[16] AS Monin and AMF Obukhov. Basic laws of turbulent mixing in the surface layer of the atmosphere. *Contrib. Geophys. Inst. Acad. Sci. USSR*, 151(163):e187, 1954.

[17] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[18] Anthony OHagan. Bayesian analysis of computer code outputs: a tutorial. *Reliability Engineering & System Safety*, 91(10):1290–1300, 2006.

[19] Luc Pronzato and Werner G Müller. Design of computer experiments: space filling and beyond. *Statistics and Computing*, 22(3):681–701, 2012.

[20] Gilles Pujol, Bertrand Iooss, Alexandre Janon with contributions from Khalid Boumhaout, Sebastien Da Veiga, Jana Fruth, Laurent Gilquin, Joseph Guillaume, Loic Le Gratiet, Paul Lemaitre, Bernardo Ramos, Taieb Touati, and Frank Weber. *sensitivity: Global Sensitivity Analysis of Model Outputs*, 2016. R package version 1.12.1.

[21] Carl Edward Rasmussen. Gaussian processes for machine learning. 2006.

[22] Olivier Roustant, David Ginsbourger, and Yves Deville. DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization. *Journal of Statistical Software*, 51(1):1–55, 2012.

[23] Andrea Saltelli, Karen Chan, E Marian Scott, et al. *Sensitivity analysis*, volume 1. Wiley New York, 2000.

[24] John H Seinfeld, Spyros N Pandis, and Kevin Noone. Atmospheric chemistry and physics: from air pollution to climate change, 1998.

[25] Ilya M Sobol. Sensitivity estimates for nonlinear mathematical models. *Mathematical Modelling and Computational Experiments*, 1(4):407–414, 1993.

[26] Somersalo and Kaipio. *Statistical and computational inverse problems*. Springer-Verlag, 2005.

[27] Felipe AC Viana, Gerhard Venter, and Vladimir Balabanov. An algorithm for fast optimal latin hypercube design of experiments. *International journal for numerical methods in engineering*, 82(2):135–156, 2010.