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.

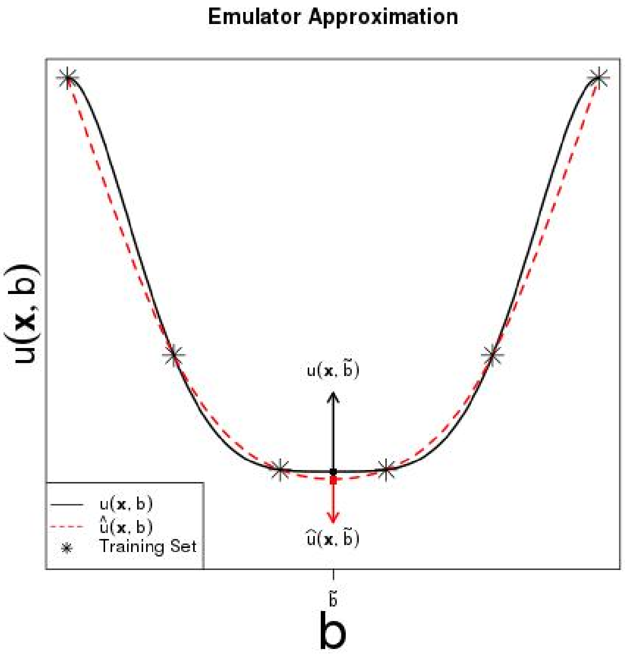

Figure 3.1: Approximation of a model $u(\mathbf{x}; \cdot)$ by the mean of a Gaussian process trained with six different outputs from the model. The mean of the Gaussian process at a point $\tilde{b}$ is taken as the value $\widehat{u}(\mathbf{x}; \tilde{b})$ of the emulator.

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 $\tilde{u}$.

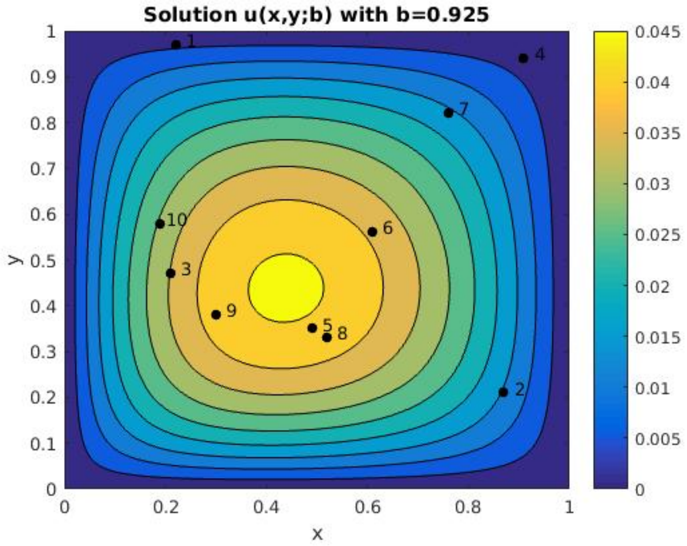

Figure 3.2: Numerical solution of the system (3.1) using a five points stencil finite difference approximation for the Laplacian. The mesh size used in x and y was 0.01. The value of the parameter b was set at 0.925. The black dots in the plot represent the points used to generate the experimental data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

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 $\mathbf{y} = (\tilde{u}(\mathbf{x}_1; b), \dots, \tilde{u}(\mathbf{x}_{10}; b))$ 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 [14]. 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 [23]

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

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

$$\mathbf{y} = \mathbf{G}(b) + \epsilon, \qquad \text{with } \epsilon \sim \mathcal{N}(0, \lambda^2 I_{10\times10}). \tag{3.6}$$

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

$$\mathbf{y}|b \sim \mathcal{N}(\mathbf{G}(b), \lambda^2 I_{10\times10}),$$

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.

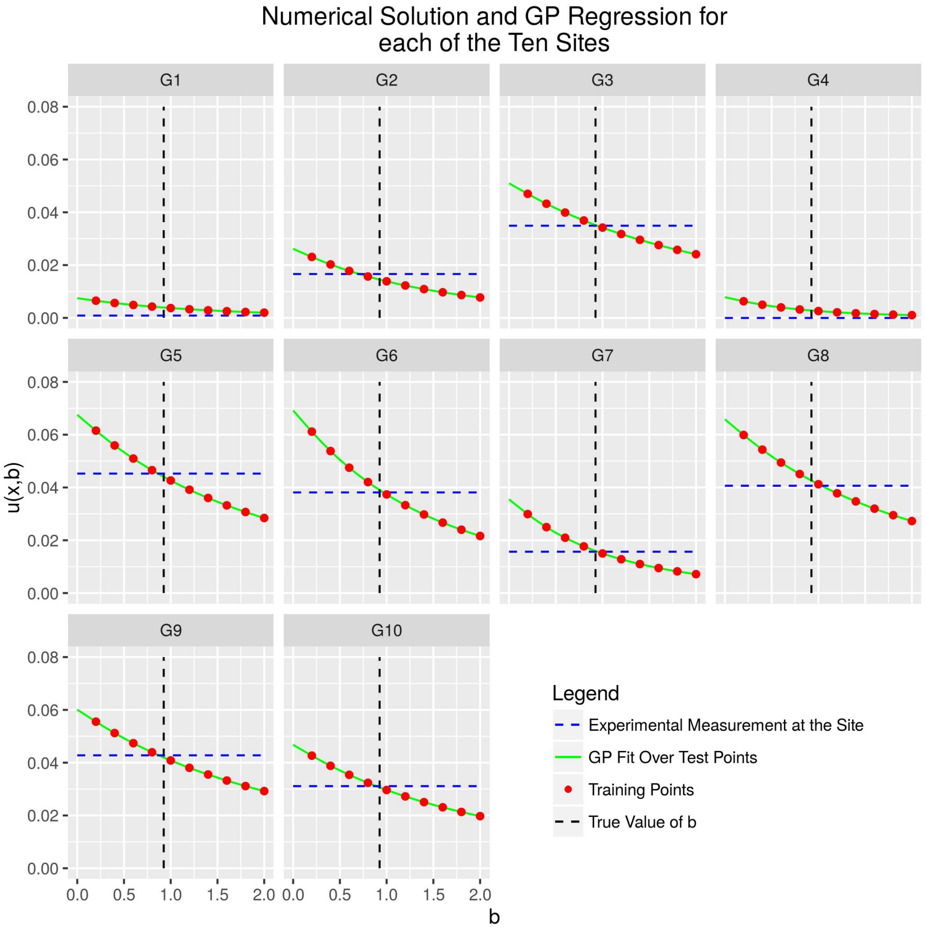

Figure 3.3: Training points, GP regression, true value of b and experimental measures for each one of the ten sites labeled from 1 to 10 in Figure 3.2

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.

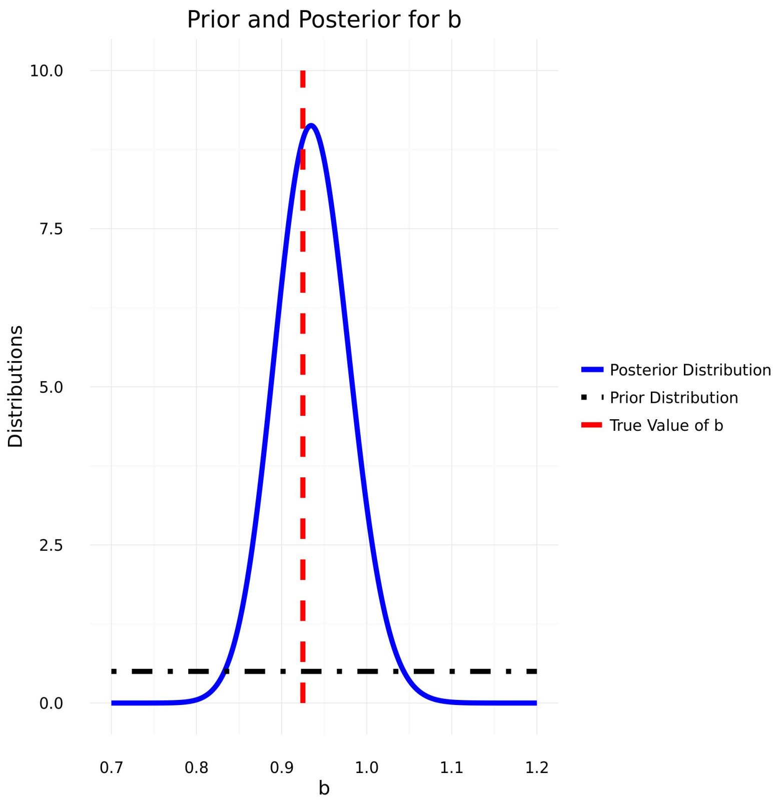

Figure 3.4: Plots of the prior distribution, posterior distribution and true value of the parameter b.

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 [23]

Algorithm 1 Metropolis-Hastings Algorithm

```
pick a point $q_1$ in the support of the distribution
for j=2:N do
    Draw $u \sim U([0, \alpha])$
    $q_j \leftarrow q_{j-1} + u$
    $\beta \leftarrow \min(1, \frac{\mathbb{P}_{post}(q_j|D)}{\mathbb{P}_{post}(q_{j-1}|\mathbf{y})})$
    Draw $w \sim U([0, 1])$
    if $w < \beta$ 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 [4]. 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 [4]. 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.

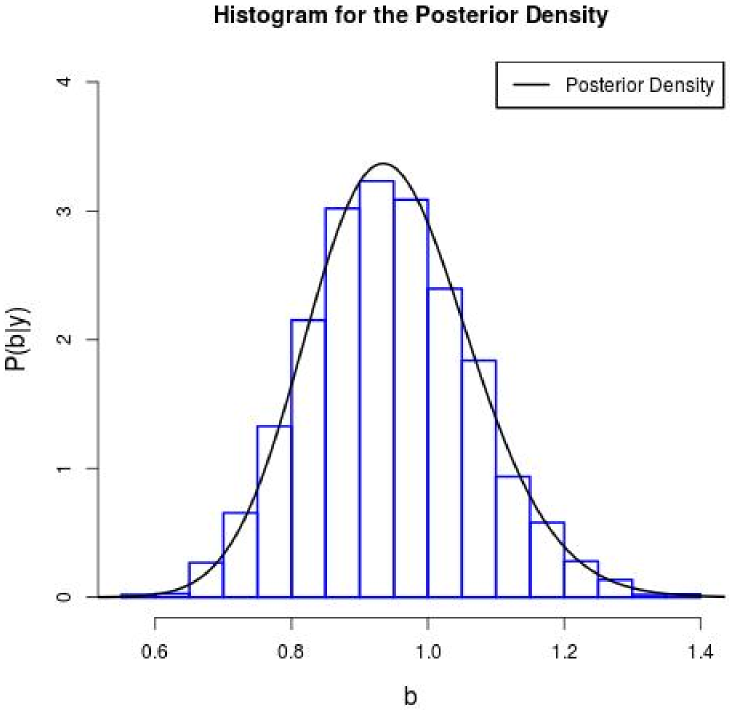

Figure 3.5: Histogram obtained for the posterior distribution (3.2) from 5000 samples from MH algorithm with step size $\alpha = 0.23$. The solid line is the graph for the posterior $\mathbb{P}_{post}(b|D)$.

With the samples obtained we readily obtain useful statistics for b. For example, we can estimate the conditional mean using [4]

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

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 $\mathbb{E}[h]$ [5]. 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

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

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.

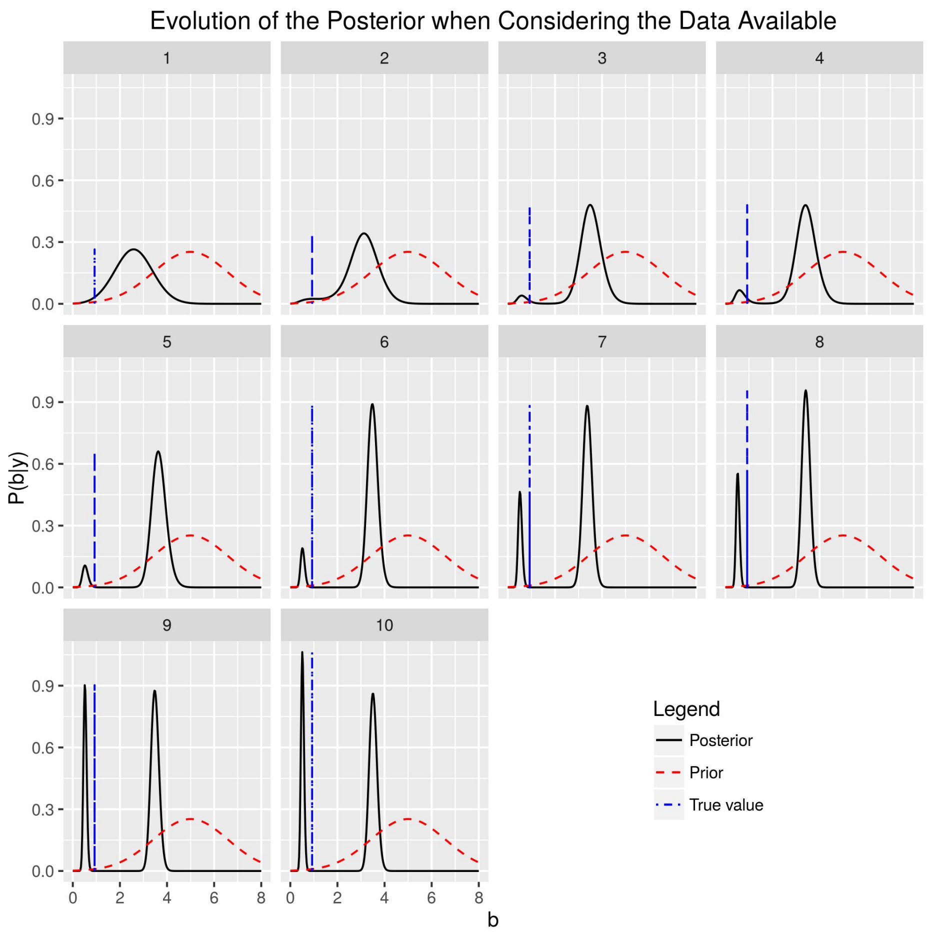

Figure 3.6: Evolution of the posterior distribution when more experimental data is taken into account

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.

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

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

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

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

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

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

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

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

[15] 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.

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

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

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

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

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

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

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

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

[24] 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.