Chapter 3

Toy Problem: How Theory Works in Practice

In the previous chapter we reviewed some of the theoretical and computational tools needed obtain the solution to Bayesian inverse problems. In this chapter we are going to work on a toy problem to illustrate how the theory explained can be applied, so in Chapter 4 we can focus mainly on results for the solution of the problem described in Chapter 1.

To talk about an inverse problem we need to specify the forward problem. We define the forward problem as the solution of 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. The function u represents the mathematical approximation of a quantity with a physical interpretation $\tilde{u}$. The behavior of $\tilde{u}$ is assumed to be modeled by equation (3.1).

In Chapter 2 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 the context of this chapter, the function $M(\cdot)$ takes as input a point $(\mathbf{x}, b) \in \Omega \times [0,1]$. 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 $\tilde{u}$. This 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 best the experimental data $\mathbf{y}$. To estimate b is the inverse problem. The obvious approach to achieve this would be to solve equation (3.1) for a big number of values b in the interval $[0, L]$ where L is large enough to guarantee that there exists a $b^* \in [0, L]$ such that the set $\{u(\mathbf{x}_1; b^*), \ldots, u(\mathbf{x}_{10}; b^*)\}$ has 'small' discrepancy with the experimental data $\mathbf{y}$. For the sake of the example we assume that solving equation (3.1) is computationally expensive and solving it for a big range of different values of b is not feasible. Therefore the need to construct an emulator $\hat{M}(\cdot)$ that approximates $M(\cdot)$ as explained in Chapter 2. Then solve equation (3.1) for a number of different values of b that makes the cost of solving (3.1) acceptable. Finally use the emulator $\hat{M}(\cdot)$ to predict the output of $M(\cdot)$ for as many different values of b as possible. The value of the emulator at the point $(\mathbf{x}, b)$ is going to be denoted by $\hat{u}(\mathbf{x}; b)$. The following table summarizes the notation that is going to be used from now on in this 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.

We want to estimate the value of b that explains the best the experimental data $\mathbf{y}$. To incorporate $\mathbf{y}$ into the inference, we can use Bayes rule and estimate the posterior distribution for b as

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

Having the posterior we can estimate b using any of the point estimates given in equation (2.6) along with the uncertainty associated with the chosen estimate.

Before we proceed to find the posterior distribution for b, let us explain how we are going to generate the experimental measurements $\mathbf{y}$. Assume that the true value of b is $0,925$. Then we solve equation (3.1) using this value of b. Next we pick ten points at random and save the value of the numerical solution u at those location (see Figure 3.1). Finally we add noise from a normal distribution with mean 0 and $\sigma = 0.01$ to each of the ten values. The resulting numbers obtained are what we use as the experimental data $\mathbf{y} = (\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$. 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 $\tilde{u}$.

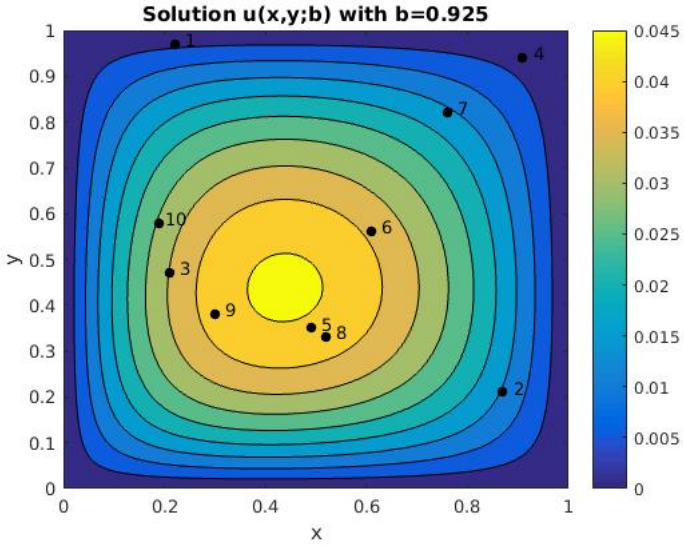

Figure 3.1: Numerical solution of the system (3.1) using a five points stencil finite difference approximation for the Laplacian. The mesh size used in x and y was 0.01. The value of the parameter b was set at 0.925. The black dots in the plot represent the points used to generate the experimental data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

Now that we explained how to generate the data $\mathbf{y}$, we continue with how to estimate b using Bayes formula (3.2). First we need to choose a prior distribution for b. For the sake of the example, let us assume that equation (3.1) describes a well known physical process and 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*. With this distribution, given a Borel measurable set $A \subset (0, 2]$, the probability that b belongs to A is given by

$$\frac{1}{2}\int_A dx.$$

In this case we have

$$\mathbb{P}_{prior}(b) = \frac{1}{2}\mathbf{1}_{(0,2]}(b), \tag{3.3}$$

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

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

To calculate the likelihood we need to know how b is connected with the data $\mathbf{y}$. The connection is given by equation (3.1). By solving explicitly this equation we can find a functional relation between u and b for each one of the ten locations depicted in Figure 3.1. It is possible to explicitly solve equation (3.1). However the relation between u and b is given by an infinite Fourier series. Having a functional relation given by an infinite series is often not very useful. Hence, the approach we are going to use is the following: by assumption, solving equation (3.1) is computationally expensive. Assume that given time and computational budget we can solve the PDE for no more than 10 different values of b (Having ten values of b and ten points where we measured $\tilde{u}$ is just coincidence). The idea is to use GPs as described in Chapter 2 to interpolate for the values of $b \in (0, 2]$ where we did not solve equation (3.1). The value of the interpolation done by the GP at a point is going to play the role of the output of the emulator $\hat{M}(\cdot)$ at that point.

We need to choose ten values of b to solve the PDE (3.1) in a way that the interpolation error for the other values of b in the set $(0, 2]$ is small compared to the error in the interpolation if we were to choose a different set of ten values for b. We shall denote the points as $\{b_1, \ldots, b_{10}\}$. To choose the ten points we use a maximin design as explained in Chapter 2. In this case it is straightforward to check that the way to choose the points that maximizes the minimum distance among the points is by locating them in an equidistant manner i.e.

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

This set gives the maximin design for our problem.
Having the design, it is now possible to create the set of training points for the GP as $\{b_i, u(\mathbf{x}_k, b_i)\}_{i=1}^{10}$ for each of the $k = 10$ sites where we obtained the experimental measurements $(\tilde{u}(\mathbf{x}_1; b), \ldots, \tilde{u}(\mathbf{x}_{10}; b))$. Using the training points, we create the GP for each of the ten sites. For the interpolation we use as a test set

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

In Figure 3.2 are plotted the training points, the GP regression over the test points, along with the true value of b and the experimental measure $\tilde{u}(\mathbf{x}_k; b)$ for each of the ten sites.

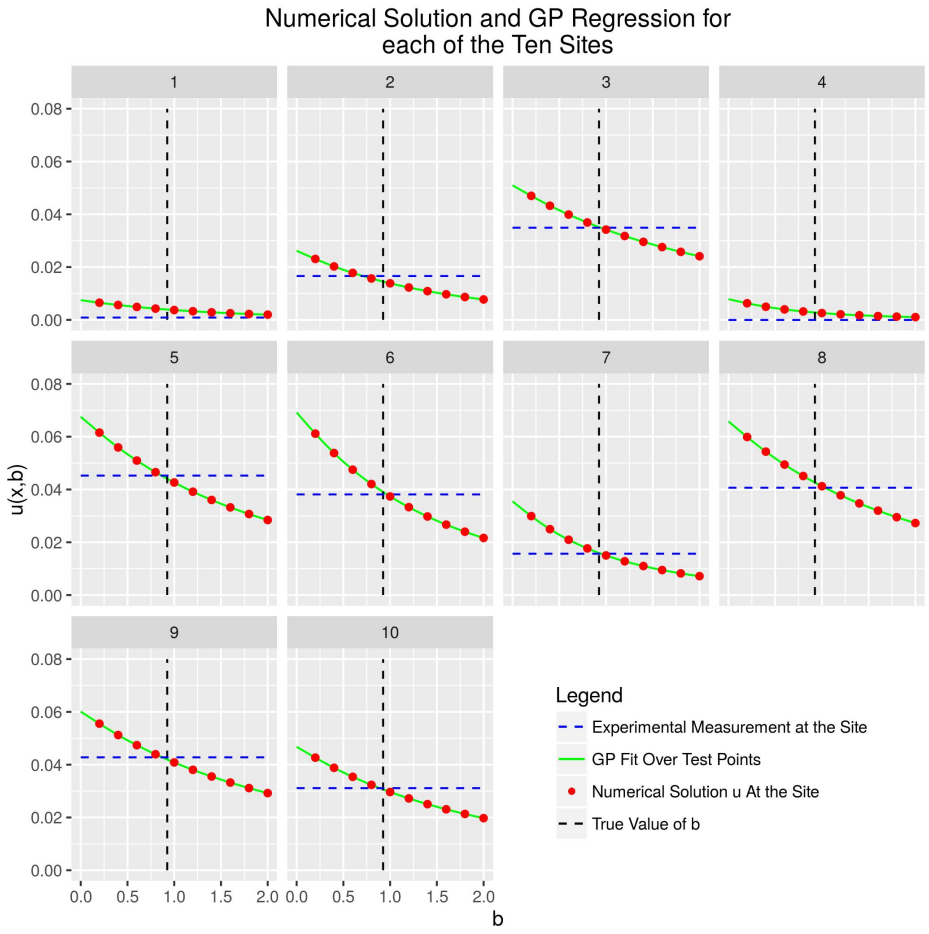

Figure 3.2: Training points, GP regression, true value of b and experimental measures for each one of the ten sites labeled from 1 to 10 in Figure 3.1

The intersection of the blue dotted line with the black dotted line in Figure 3.2 represents, the value the numerical solution u would attain if it were to perfectly model the physical variable $\tilde{u}$. Since we added noise to the values $\{u(\mathbf{x}_1; 0.925), \ldots, u(\mathbf{x}_{10}; 0.925)\}$ we know that the value of $\tilde{u}$ has to be different from the value of u.

The solid line in Figure 3.2 gives a functional relation between $\hat{u}$ and b for each of the ten sites. Let us denote $G_k(b) := \hat{u}(\mathbf{x}_k; b)$ and $y_k = \tilde{u}(\mathbf{x}_k; b)$ for $k = 1, \ldots, 10$. A possible functional relation that connects these quantities

is

$$y_k = G_k(b) + \epsilon_k, \qquad \text{where } \epsilon_k \sim \mathcal{N}(0, \lambda^2), \tag{3.4}$$

with λ a positive number that models how much we believe the value of $\tilde{u}$ differs from the GP prediction $\hat{u}$. We chose the value $\lambda = 5.4 \times 10^{-3}$ to get a signal to noise ratio with $\mathbf{y}$ of 1:10. If we define the vector $\mathbf{G}(b) = (\hat{u}(\mathbf{x}_1; b), \ldots, \hat{u}(\mathbf{x}_{10}; b))$ and recalling the definition of $\mathbf{y}$ (see table 3.1), equation (3.4) can we written more compactly as

$$\mathbf{y} = \mathbf{G}(b) + \epsilon, \qquad \text{where } \epsilon \sim \mathcal{N}(0, \lambda^2 I_{10\times 10}). \tag{3.5}$$

Since the random vector ϵ has multivariate Gaussian distribution, we can use equation (3.5) to conclude [19]

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

Explicitly

$$\mathbb{P}(\mathbf{y}|b) \propto e^{-\frac{1}{2\lambda^2}\|\mathbf{G}(b)-\mathbf{y}\|_2^2}, \tag{3.6}$$

where the proportionality constant normalizes the distribution on the right hand side to one. Since the denominator in Bayes rule in equation (3.2) is independent of b and serves only as a normalization constant we can use equations (3.3) and (3.6) to write

$$\mathbb{P}_{post}(b|\mathbf{y}) \propto \mathbb{P}_{like}(\mathbf{y}|b)\mathbb{P}_{prior}(b) \propto \mathbf{1}_{(0,2]}(b) e^{-\frac{1}{2\lambda^2}\|\mathbf{G}(b)-\mathbf{y}\|_2^2}. \tag{3.7}$$

An interpretation of this result is that before taking experimental measurements all we knew about the parameter b was that $b \in (0, 2]$ after weighting this prior believe with the data $\mathbf{y}$ our current state of knowledge about the parameter b is encoded in the posterior distribution. Figure 3.3 plots this update in our knowledge about b.

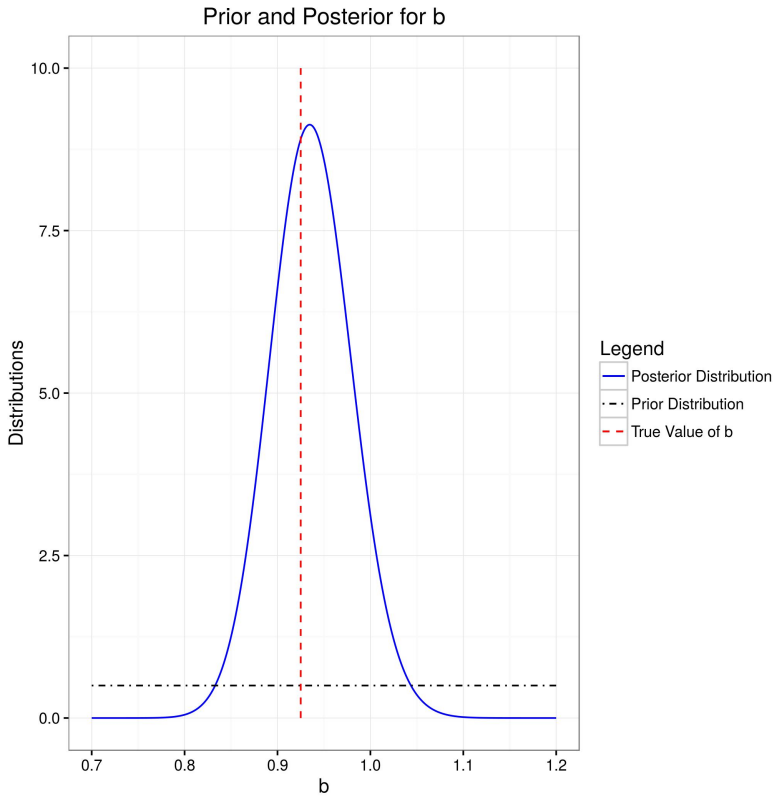

Figure 3.3: Plots for the prior distribution, posterior distribution and true value of the parameter b.

Having a visual representation of the posterior distribution is useful, but, in order to obtain statistics for the possible value of b we need to sample from the posterior. A family of methods that allow to sample from fairly arbitrary distributions are known as Markov Chain Monte Carlo (MCMC). In this work we are going to focus in 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 (3.7) as an example.

Consider the posterior distribution $\mathbb{P}_{post}(b|\mathbf{y})$. The idea is to wander around the support of the distribution in a way that points in the support with high probability are visited more often than those with low probability. For example, 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 $\frac{\mathbb{P}_{post}(q_2|\mathbf{y})}{\mathbb{P}_{post}(q_1|\mathbf{y})}$ otherwise. On the other hand if we are at a point q_1, we choose in what direction to move, randomly, using some probability distribution that is easy to sample from. In this and the next Chapter we choose the uniform distribution to decide in what direction to move. The pseudocode for the MH algorithm as described above is [19]

1. pick a point q_1 in the support of the distribution

   for j=2:N

2. Draw $u \sim U([0, \alpha])$

3. $q_j \leftarrow q_{j-1} + u$

4. Compute $\mathbb{P}_{post}(q_j|D)$

5. $\beta \leftarrow \min(1, \frac{\mathbb{P}_{post}(q_j|D)}{\mathbb{P}_{post}(q_{j-1}|D)})$

6. Draw $w \sim U([0, 1])$

   if $w < \beta$

7. $q_{j-1} = q_j$ (Accept move)

   else

8. $q_{j-1} = q_{j-1}$

   end
   end

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 [3]. 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 $N \to \infty$ the samples come from $\mathbb{P}_{post}(b|\mathbf{y})$ and are independent. 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 [3]. Since we do not have the computational power to let $N \to \infty$ to achieve independence from the samples, what is done in practice is to consider a number N as big as possible given computational and time constraints. Then consider only a fraction of the last samples

obtained. For example we may choose the last $N/2$ samples obtained from the MH scheme and discard the rest of the samples. The iterations where we obtain samples that we discard later is know as the *burning period*.
Using the MH scheme to sample from the posterior distribution $\mathbb{P}_{post}(b|\mathbf{y})$ with $\alpha = 0.23$ and $N = 10000$. We set the burning period to be the set of the first 5000 samples. Below is shown the result obtained from sampling the posterior in equation (3.7).

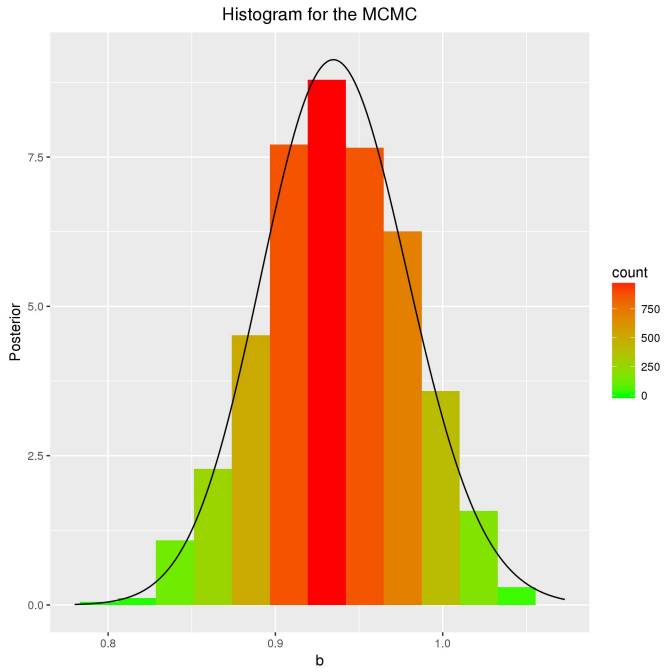

Figure 3.4: Histogram obtained for the posterior distribution (3.2) from 5000 samples from MH algorithm with step size $\alpha = 0.23$. The solid line is the graph for the posterior $\mathbb{P}_{post}(b|D)$.

With the samples obtained we readily obtain useful statistics for b. For example we can estimate the conditional mean by using Monte Carlo integration

$$b_{cm} = \int_{(0,2]} b\mathbb{P}_{post}(b|\mathbf{y})db \approx \frac{1}{5000}\sum_{j=1}^{5000} b_j = 0.9247042, \tag{3.8}$$

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

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

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

With these values we can say that with 95% of probability, the true value of b belongs to the interval

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

Let us digress about the idea behind Monte Carlo integration. Consider the generic problem of evaluating the n-dimensional integral

$$\int_{\mathbb{R}^n} h(x)\rho(x)dx, \tag{3.9}$$

where ρ is the Lebesgue density of some probability measure $\mathbb{P}$. This means that calculating (3.9) 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 $\mathbb{E}[h]$ [4]. 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 $\mathcal{O}(\frac{1}{\sqrt{n}})$. In practice, to estimate the quality of the result from the Monte Carlo integration we use the approximation

$$\mathbb{P}\left(\frac{h_n - \mathbb{E}[h]}{\sqrt{\sigma_n}} \leq x\right) \approx \Phi(x),$$

where

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-\frac{x^2}{2}) dx.$$

To conclude this Chapter we are going to turn our attention into the subjective part of Bayesian statistics. We are going to talk about the role the prior probability has in the inference process.

How important is the prior to make inferences?

It is constructive to explore how relevant is the choice of the prior distribution in making inferences about a parameter of interest. Let us consider the same problem, we want to estimate the value of the parameter b. For demonstration purposes, let us assume two things: the parameter b can be any real number (not just $0 < b \leq 2$ as before) and we assume a prior distribution for b as

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

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

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

As before, assume that the true value of b is 0.925. 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 assign with high probability. Let us evaluate how the posterior distribution for b evolves as we consider more and more experimental data from the measurements of $\tilde{u}$. In Figure 3.5 it is shown 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 frame we proceed adding one more measurement at a time and in the tenth frame we calculate the posterior using the data obtained from all ten points.

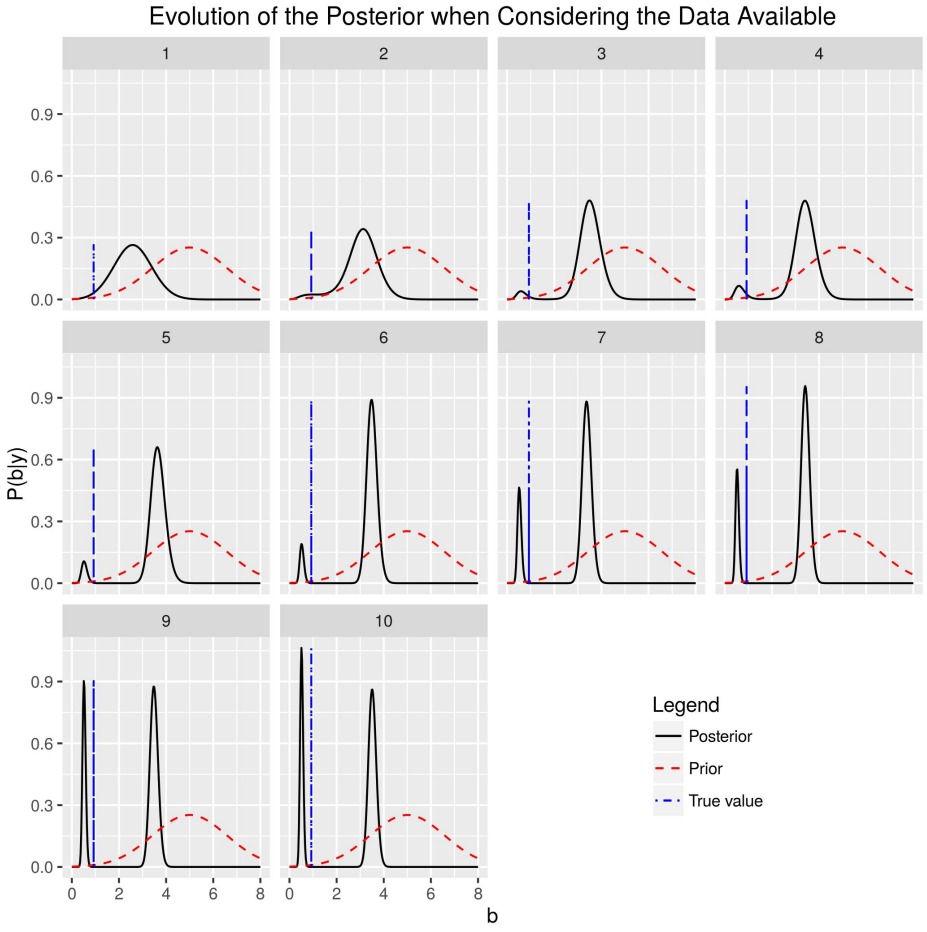

Figure 3.5: Evolution of the posterior distribution when more experimental data is taken into account

The sequence of plots in Figure 3.5 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 originate by the prior at the point $b = 4$. The reason for this final posterior distribution is that the prior gives a large probability to values around $b = 4$, whereas give a close to zero probability for values around $b = 0.925$. When the experimental data is used, the likelihood distribution points to values close to $b = 0.925$. The more data we use the stronger the weight that the likelihood has compared to the prior distribution. However since the prior distribution gives negligible probability to the true value of b, when all data are used there is an equilibrium between the likelihood and prior that is expressed as a bimodal distribution. This result is a cautionary tale. If we know how to choose the prior distribution in a way that is meaningful to the problem, reliable inference could be done with small amount of data. On the contrary if the prior distribution is not realistic, inference may not be reliable and a big amount of data is needed to correct for the bias included in the prior distribution. We can summarize this with the following analogy: if you really believe in Santa you will need significant evidence that he does not exist to stop believing in his existence.

Bibliography

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

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

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

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

[5] Delphine Dupuy, Céline Helbert, Jessica Franco, et al. Dicedesign and diceeval: Two r packages for design and analysis of computer experiments. *Journal of Statistical Software*, 65(11):1–38, 2015.

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

[7] 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.

[8] 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.

[9] 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.

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

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

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

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

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

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

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

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

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

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

[20] 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.