The Confluence of Gaussian Process Emulation and Wavelets

Christopher Alexander Pope

Submitted in accordance with the requirements
for the degree of Doctor of Philosophy

The University of Leeds

School of Mathematics

August 2019

Acknowledgments

Firstly, I would like to thank my supervisors, John Paul Gosling and Stuart Barber, for their continuous encouragement, knowledge, and support throughout my studies. I would also like to thank The John E Crowther – Martin Clarke Research Foundation for their financial support. Finally, I would like to thank my family for their emotional support, and for encouraging me to strive to be the best I can be.

Abstract

We discuss two thriving research areas, emulation (in the statistical sense) and wavelet analysis, and explore ways in which the two areas can complement each other to tackle problems that both areas face. The Gaussian process, which is the popular choice in emulation, is used due to its ability to be a surrogate for a function when we are only able to make a limited number of observations from the function. The Gaussian process, however, does not perform well when the underlying function contains a discontinuity. Wavelet analysis, on the other hand, is known for its ability to model and analyse functions that contain discontinuities. Wavelet analysis tends to require a large number of datapoints to be able to model functions accurately, tending to struggle when the amount of data is limited.

As it appears that one area's strength is the other area's weakness, this thesis is aimed at exploring the possible overlaps between the two methods, and the ways in which they could benefit each other. Particular attention in the thesis is paid to the challenges that are faced when the function that we are attempting to model contains discontinuities, or, areas of space in which there is a sharp increase/decrease in the value of our observations. We develop methods to select the location of additional design points after we have observed the function at our original design points with the objective of better defining the location of the discontinuity. We also develop novel methods to model the unknown function that we believe contains discontinuities, and look to accurately find our uncertainty in this function.

Glossary of terms

- Variables/inputs/locations - Parameter values
- Output - Observed value.
- TGP - Treed Gaussian process.
- Standard Gaussian process - a stationary Gaussian process with a euclidean covariance function.
- MAO - Multi-output emulator.
- TAI - Using the time-point of an observation as a parameter value in our Gaussian process.
- IGP - Independent Gaussian process; Building a separate independent (over time) Gaussian process for each time-point.
- GP - Gaussian process.
- running the model - Gaining an observation of a function f with the parameters $\mathbf{x}$.
- SMSE - Standardised mean squared error.
- LHD - Latin hyper-cube design.
- MRA - Multiresolution analysis.
- (I)DWT - (Inverse) Discrete wavelet transformation.
- NDWT - Non-decimated wavelet transformation.
- AAIPE - Averaged absolute individual prediction error.
- Standard model (Chapter 5)- the model introduced by Kim et al. (2005).
- RJMCMC - Reversible-jump Markov chain Monte Carlo.
- MAD - Maximum absolute deviation.

- Smoothness - $f(x)$ will be 'close' to $f(x')$ given that x and x' are 'close'.
- Mathematical smoothness - a function is infinitely differentiable (except for at e.g. jump discontinuities).

Notation glossary

- y - observed value / output
- x - parameter / input / variables
- x^* - unobserved parameter value
- b - length scale
- nu - smoothness term in Matern covariance function
- n - number of observed values
- ψ - the mother wavelet function
- ϕ - the father wavelet function
- $\hat{\psi}$ - the Fourier transform of the mother wavelet function
- d_{jk} - the coefficient of the mother wavelet function at scale j and translation k
- c_{jk} - the coefficient of the father wavelet function at scale j and translation k
- d_{jk} - the coefficient of the mother wavelet function at scale j and translation k, observed with noise
- $\hat{y}$ - estimate of $f(x^*)$
- n_w - moving window size
- WV - windowed variance
- J - coarsest level of the wavelet decomposition
- m - the number of vanishing moments
- $D(y_i)$ - the individual prediction error for y_i
- s - the centres of a Voronoi tessellation
- r - the number of regions for a Voronoi tessellation

- k - the number of centres for a Voronoi tessellation (Chapter 5)
- Σ_p - the mixing parameter for the RJMCMC
- $\mathbf{t}$ - the collection of Voronoi tessellation parameters
- n_p - the number of points to sample
- $g(x)$ - the observed function value without random noise
- W - the forward discrete wavelet transformation in matrix form
- W^T - the inverse discrete wavelet transformation in matrix form
- a_j - a scale parameter at level j of the wavelet decomposition
- ω_j - the mixture probability of the wavelet coefficient at level j
- σ_{MAD} - MAD estimate of σ
- eta^2 - variance term of the wavelet coefficient
- T - The total number of time points
- t - time point

Contents

Chapter 1

Introduction

Emulation (in the statistical sense) and wavelet analysis are two areas of statistics that have had a considerable amount of research undertaken, and continue to be thriving areas of research to this day. In this thesis, we not only look to find new methods that may aid one of these research areas, but we look at ways in which these two respective areas can be combined to solve a problem that these areas face.

Emulation in this thesis refers to the process in which a person's belief and uncertainty about an unknown function $f(\cdot)$ is represented through a probability distribution. This area is particularly important when we have situations in which we do not have access to large amounts of data relating to this unknown function. In these cases, the prior distribution can play a large role in any decisions resulting from the statistical analysis of this function, and, hence, the ability to accurately represent our belief in the unknown function is crucial. Emulation as a research area has been particularly popular in the analysis of computer models. Due to the complex nature of many computer models (both in terms of computational run time and understanding of the underlying process), emulation is often used in the exploration of these models, with the underlying mechanics of the computer model treated as a 'black box'. By black box, we are referring to the fact that we do not fully consider the intricate underlying equations (or set of equations) that make up the computer code, but instead treat it as an unknown function in which we attempt to assess our beliefs about certain features of this function.

A particular branch of research in the emulation methodology, which traditionally utilises the Gaussian process as our prior belief about the unknown function, is the ability of the Gaussian process to accurately represent our beliefs about the function when there is the existence of a discontinuity within the true function. As we will see in various chapters throughout the thesis, the Gaussian process does not perform well when there are discontinuities or sudden increases/decreases in the observations for a small change in parameter value, and, so, the methodology must be adapted to deal with these situations.

Conversely, wavelet methods are a popular tool in the analysis of functions that con-

tain discontinuities. Their use in statistics is often through the use of the discrete version of the wavelet transformation, due to the fact that we are often using a discrete and finite number of data points in our statistical analysis. Within this discrete version, we use a multiscale basis of wave like functions to represent our data vector. This allows us to observe the existence and prevalence of wave-like features within the data for both varying frequencies and the locations in which these frequencies occur. Discontinuities can be represented by a 'large' coefficient in the basis representation, and can be a concise way of displaying a discontinuity.

Discrete wavelet methods tend to require a large number of data points to accurately capture the features that are prevalent in the data. This is another contrast between wavelets and Gaussian process emulation, in that, whilst wavelet methods require many data points, one of the strengths of the Gaussian process is its performance when data are scarce.

The aim of this thesis is to bring together these two branches of research, in which each method's perceived weakness is perceived to be a strength of the other. The thesis shows various ways in which these methods can produce a useful tool in the others field. In Chapters 2 and 3, we give an introduction to the framework of Gaussian process emulation and wavelet methodology respectively that is used throughout the thesis.

In Chapter 4, we begin our exploration by looking at a problem that both literatures have an interest in. That is, at what location to sample the function at to better define a discontinuity if we are able to select the further sampling locations in a one-dimensional parameter problem. We introduce a novel method that aims to sample new points around the location of a discontinuity. The method uses the Gaussian process to identify the existence of a discontinuity by examining the posterior mean of the function. Wavelets are then used to decide on the location of the new sample by analysing the features of the mean of the Gaussian process

In Chapter 5, we then consider the case in which we have a multi-dimensional parameter case; a method is introduced that not only models those situations in which we have a discontinuity in parameter space, but also tackles the problem of selecting new parameter locations to sample at to better define the discontinuity. This novel method partitions the input space using an adapted Voronoi tessellation that allows Voronoi cells to merge and create a larger cell. Each of these partitions, which we call regions, are assumed to have smoothness within their respective regions, and, hence, Gaussian processes are fit to each region. We account for the uncertainty in the number and shape of the regions by assigning a prior distribution to the parameters that define the partition. The sampling method introduced is also novel, in which we use the most probable model in the posterior distribution, and sample new points along the boundaries of the regions of this model.

In Chapter 6, we attempt to tackle the problem in which we have a large number

of data points in a one dimensional parameter problem, and we believe the unknown function to have a discontinuity. The method introduced places emphasis on the ability to accurately represent our uncertainty in the function. This is done using a Gaussian process to model any long term trend that the function possesses, with an additive wavelet function used to attempt to model any discontinuities that are present in the function. We bring together the two functions to create a novel method for not only estimating the underlying function, but also to provide useful uncertainty measures of the estimate.

In Chapter 7, we move on to discuss the problems that we face when our unknown function produces a high dimensional time series-like output. We introduce a method that involves utilising wavelets to reduce the dimensionality of the problem, then using a Gaussian process prior on the reduced dimensional problem. As Wavelets can often represent function and data vectors sparsely, requiring only a few coefficients to capture most of their features, we use this property to reduce the computational heaviness in terms of time that can be associated with the analysis of these high dimensional functions.

Chapter 2

Modelling beliefs about functions using a Gaussian process

2.1 Posterior inference for the Gaussian process model

We may have a situation in which we have a function, $f(\mathbf{x}) \in R^1$, that we are able to evaluate a number of parameters (or inputs in computer models), $\mathbf{x} = (x_1, \ldots, x_n)$, $x_i \in R^1$ and $n \in \mathbb{N}^+$. This gives us a data vector

$$\mathbf{D} = (f(x_1), \ldots, f(x_n)) \,. \tag{2.1}$$

Often, we are faced with functions that are deterministic. By deterministic, we are describing a function in which every time we compute $f(\cdot)$ for an input x, we would get the same output $f(x)$. There are many ways in which we can model this situation, including methods such as linear regression (e.g. Montgomery et al. 2012), smoothing splines (e.g. Wood et al. 2002), wavelets (e.g. Heaton & Silverman 2008), which we will cover in a later chapter, and many other methods. The tool of choice that is used to model a function in this thesis is the Gaussian process (GP). The Gaussian process uses the assumption that the function is 'smooth' or, in other words, that $f(x)$ will be 'close' to $f(x')$ given that x and x' are 'close'. It is important that we differentiate between this definition of smoothness, which will be used throughout the thesis, and the classical definition of smoothness in which a function is infinitely differentiable (except for at e.g. jump discontinuities), which we will call 'mathematical smoothness' throughout the thesis. It is this assumption that is the driving force behind the usefulness of Gaussian process, and why it is often utilised when we do not have much data, or when the data are sparse (Rasmussen & Williams 2006).

For a function $f(\cdot)$, that we believe is distributed as a Gaussian process, we write $f(.) \sim GP(\zeta(\cdot), K(\cdot, \cdot))$. This says that the random function $f(.)$ is distributed as a Gaussian process with mean function $\zeta(\cdot)$ and covariance function $K(\cdot, \cdot)$ as its parame-

ters. The randomness in this case is due to the fact that have not observed the reality and, hence, model our beliefs through randomness. The parameters of the Gaussian process distribution will be discussed in more detail later in this chapter, as they are extremely important for any inference we make about f. We could think about this distribution as a natural extension to the multivariate Gaussian distribution, in which we have an infinite number of variables. Further to this, the Gaussian process has the property that any finite collection of random variables in our Gaussian process has a multivariate Gaussian distribution $\big($that is, $(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n))$ will follow a multivariate Gaussian distribution$\big)$ (Gelman et al. 2013).

Now, for any set of inputs $\mathbf{x}$, we can define our uncertainty around the outputs $\mathbf{D}$ by the multivariate Gaussian distribution, given that we believe f is a Gaussian process.

The multivariate normal distribution is fully defined by its mean and covariance structure (Chatfied & Collins 2013). Firstly, we can define the mean of the function, at any x by

$$\mathbb{E}(f(x)|\boldsymbol{\beta}) = h(x)^T\boldsymbol{\beta},$$

where $h(x)^T$ is the functional form of $f(.)$, with $\dim(\beta) = k$, $k < n$, and defines any prior belief we have about the form of the underlying function $\big($e.g. we may believe it has a quadratic form, as such a natural choice would be $h(x)^T = (1, x, x^2)\big)$, and $\boldsymbol{\beta}$ is a vector of corresponding coefficients.

The covariance between any two values $f(x)$ and $f(x')$ can be defined by

$$\text{Cov}(f(x), f(x')|\sigma^2) = \sigma^2 c(x, x'), \tag{2.2}$$

where $c(x, x')$ is a covariance function. The covariance function must be positive semi-definite function to be a valid covariance function; for many popular choices of covariance function, we also tend to see functions that decreases as $|x - x'|$ increases (that is, the further away two points are, the less correlated we expect them to be). We typically see covariance functions that have the property $c(x, x) = 1$ (Cressie 1993). One classical example of a covariance function is the squared exponential covariance function, which is defined as

$$c(x, x'|b) = \exp\left(-\frac{(x - x')^2}{2b^2}\right). \tag{2.3}$$

There are many decisions that we must make when considering our prior beliefs about the function. Two obvious considerations are the choice of mean function $h^T(\cdot)$ and the choice of covariance function $c(\cdot, \cdot)$. As the mean and covariance depend on $\boldsymbol{\beta}$ and σ^2 respectively, both of which we are uncertain of, we need to define a prior distribution for them. One possible choice of this is a conjugate analysis, in which we use the multivariate normal inverse-gamma distribution $(\boldsymbol{\beta}, \sigma^2 \sim NIG(M, V\sigma^2, r, a))$(O'Hagan &

Forster 2004). In this case, we have

$$\pi(\boldsymbol{\beta}, \sigma^2) \propto \left(\sigma^2\right)^{-\frac{1}{2}(p+r+2)} \exp\left\{-\frac{1}{2\sigma^2}\left[(\boldsymbol{\beta} - M)^T V^{-1} (\boldsymbol{\beta} - M) + a\right]\right\}. \tag{2.4}$$

In this, the hyperparamaters r and a are actually $\frac{r}{2}$ and $\frac{a}{2}$, however, for convenience we write r and a. This distribution should be used if our prior beliefs can actually be represented by the normal inverse-gamma distribution. Prior distributions could be placed on the length scale parameter b, however this tends not to be done in practice due to the posterior distribution being intractable when a non-informative prior is used. Other methods, which are discussed in Section 2.2.3, are used to estimate the value of the parameter.

Now, if we have a set of inputs $(x_1, \ldots, x_n)$ and evaluate our model to produce a set of outputs $(y_1 = f(x_1), \ldots, y_n = f(x_n))$, we can create our likelihood by first defining

$$H^T = \left(h(x_1)^T, \ldots, h(x_n)^T\right), \boldsymbol{y}^T = \left(f(x_1), \ldots, f(x_n)\right),$$

$$\text{and } A = \begin{pmatrix} 1 & c(x_2, x_1) & \ldots & c(x_n, x_1) \\ c(x_1, x_2) & 1 & \ddots & c(x_n, x_2) \\ \vdots & \ddots & \ddots & \vdots \\ c(x_1, x_n) & c(x_2, x_n) & \ldots & 1 \end{pmatrix}.$$

We can then define our likelihood to be

$$\pi\Big(\boldsymbol{y}|\boldsymbol{\beta}, \sigma^2\Big) \propto (\sigma^2)^{-\frac{n}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y} - H\boldsymbol{\beta})^T A^{-1}(\boldsymbol{y} - H\boldsymbol{\beta})\right\}. \tag{2.5}$$

As stated earlier, any finite collection of random variables from a Gaussian process has a multivariate Gaussian distribution. Using this property, we can partition our Gaussian process into known and unknown random variables, updating our beliefs about the Gaussian process using the random variable that we realise by running the model. Consider a random variable $\boldsymbol{P}$, which is multivariate normally distributed — $\boldsymbol{P} \sim N(\boldsymbol{\mu}, \Sigma)$. We can partition the vector $\boldsymbol{P}$ into two separate vectors (Chatfied & Collins 2013), $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ say, such that

$$\boldsymbol{P} = \begin{pmatrix} \boldsymbol{P}_1 \\ \boldsymbol{P}_2 \end{pmatrix}.$$

As such, the mean vector and covariance matrix will also be partitioned into

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix} \text{ and } \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}.$$

From this, we can show (Mardia et al. 1979) that

$$\boldsymbol{P}_1|\boldsymbol{P}_2 \sim N(\boldsymbol{\mu}^*, \Sigma^*), \tag{2.6}$$

where

$$\boldsymbol{\mu}^* = \boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\boldsymbol{P}_2 - \boldsymbol{\mu}_2),$$
$$\Sigma^* = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}.$$

Therefore, using equation (2.6), if we partition our function with $\boldsymbol{P}_1$ the unknown random variables, $f(.)$, and $\boldsymbol{P}_2$ the known random variables, $\boldsymbol{y}$, it follows that

$$f(\mathbf{x}^*)|\boldsymbol{y}, \boldsymbol{\beta}, \sigma^2 \sim N(\mu^*(\mathbf{x}^*), \sigma^2 c^*(.,.)), \tag{2.7}$$

where

$$\begin{aligned}\mu^*(\mathbf{x}^*) &= h(\mathbf{x}^*)^T\beta + t(\mathbf{x}^*)^T A^{-1}(\boldsymbol{y} - H\beta),\\ \sigma^2 c^*(\mathbf{x}, \mathbf{x}') &= \sigma^2 c(\mathbf{x}, \mathbf{x}') - \sigma^2 t(\mathbf{x})^T(\sigma^2 A)^{-1}\sigma^2 t(\mathbf{x}')\\ &= \sigma^2(c(\mathbf{x}, \mathbf{x}') - t(\mathbf{x}')^T(A)^{-1}t(\mathbf{x}')),\end{aligned}$$

and

$$t(x)^T = (c(x, x_1), \ldots, c(x, x_n)).$$

This is also known as kriging in which both β and σ^2 are known.

The second term in μ^* ensures that the mean of our posterior beliefs about the function goes through the points that we observed in our model run. We can see that $\boldsymbol{y} - H\beta$ gives the distance between the 'true values' of the simulator and our estimates, which can be thought of as an error term. When a point x_r has been realised, the vector $t(x_r)^T A^{-1}$ is zero for the output error terms that do not correspond with x_r in the $\boldsymbol{y} - H\beta$ vector, and is 1 for the term that does correspond to the x_r term. This is due to the fact that we are using a deterministic function and so we know the value of the function at the locations that we observe from the function. Hence, the estimate is adjusted so that the mean matches the model output at the input locations.

2.1.1 Posterior beliefs using the NIG

Now, returning to our beliefs about β and σ^2, using Bayes theorem we can update our prior beliefs from equation (2.4) to get our posterior distribution

$$\begin{aligned}\pi(\beta, \sigma^2|\mathbf{y}) \propto (\sigma^2)^{-\frac{1}{2}(p+r+2)}(\sigma^2)^{-\frac{n}{2}} \exp\left\{-\frac{1}{2\sigma^2}\left[(\boldsymbol{\beta} - M)^T V^{-1}(\boldsymbol{\beta} - M) + a\right]\right\}\\ \times \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y} - H\boldsymbol{\beta})^T A^{-1}(\boldsymbol{y} - H\boldsymbol{\beta})\right\}\\ = (\sigma^2)^{-\frac{1}{2}(p+r+n+2)} \exp\left\{-\tfrac{1}{2\sigma^2}\left[(\boldsymbol{\beta}-M)^T V^{-1}(\boldsymbol{\beta}-M)+(\boldsymbol{y}-H\boldsymbol{\beta})^T A^{-1}(\boldsymbol{y}-H\boldsymbol{\beta})+a\right]\right\}\end{aligned}$$

Let $r^* = r+n$. Now we can complete the square inside the exponential to get the posterior in the form

$$\pi(\beta, \sigma^2|\boldsymbol{y}) \propto \left(\sigma^2\right)^{-\frac{1}{2}(p+r^*+2)} \exp\left\{-\frac{1}{2\sigma^2}\left[(\boldsymbol{\beta} - M^*)^T V^{*-1}(\boldsymbol{\beta} - M^*) + a^*\right]\right\},$$

where

$$V^* = (V^{-1} + H^T A^{-1} H)^{-1}, \qquad M^* = V^*(V^{-1}M + H^T A^{-1}\boldsymbol{y}),$$

$$\text{and } a^* = a + \boldsymbol{y}^T A^{-1}\boldsymbol{y} + M^T V^{-1} M + M^{*T} V^{*-1} M^*.$$

As we expected, this is in the same form as the prior distribution.

If we look at the prior distribution in equation (2.4), we can write $\sigma^2 \sim IG(r, a)$ and $\beta|\sigma^2 \sim N(M, \sigma^2 V)$. We know that the inverse gamma has a mean of $\frac{a}{r-2}$. Therefore, we can use this property to find our posterior mean for σ^2, which is

$$\hat{\sigma}^2 = \frac{(a + \boldsymbol{y}^T A^{-1}\boldsymbol{y} + M^T V^{-1} M + M^{*T} V^{*-1} M^*)}{r + n - 2}.$$

We can see that our posterior mean is a combimation of the variance of the data ($\boldsymbol{y}^T A^{-1}\boldsymbol{y}$), and the terms from our prior distribution, which increases the value of σ^2. We can also use equation (2.4) and integrate out σ^2 to get a marginal distribution for β. We do this by noticing that $\sigma^2 \sim IG\left(\frac{p+r+2}{2}, \frac{(\beta-M)^T V^{-1}(\beta-M)+a}{2}\right)$, and β is proportional to a generalised student-t distribution ($\beta \sim t_r(M, aV)$). Therefore, we can use the conjugacy to find the posterior mean

$$\hat{\beta} = M^* = V^*(V^{-1}M + H^T A^{-1}\boldsymbol{y}).$$

2.1.2 Updating with noninformative priors

We may be in a situation where there is little or no prior beliefs about the joint distribution of $\pi(\boldsymbol{\beta}, \sigma^2)$; in such cases, an improper prior may be used $\pi(\boldsymbol{\beta}, \sigma^2) \propto \sigma^{-2}$. We find the posterior distribution by combining this with equation (2.5) and using Bayes theorem to get

$$\pi\Big(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}\Big) \propto (\sigma^2)^{-\frac{(n+2)}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{y} - H\boldsymbol{\beta})^T A^{-1}(\boldsymbol{y} - H\boldsymbol{\beta})\right\}. \tag{2.8}$$

Note that we can write

$$\hat{\beta} = (H^T A^{-1} H)^{-1} H^T A^{-1}\boldsymbol{y},$$
$$\hat{\sigma}^2 = \frac{\boldsymbol{y}^T(A^{-1} - A^{-1}H(H^T A^{-1}H)^{-1}H^T A^{-1})\boldsymbol{y}}{n - q - 2},$$

the generalised least squares estimate of our parameters, to rearrange equation (2.8) and get it into the form

$$\pi\Big(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}\Big) \propto (\sigma^2)^{-\frac{(n+2)}{2}} \exp\left\{-\frac{1}{2\sigma^2}\left[(\beta - \hat{\beta})H^T A^{-1} H(\beta - \hat{\beta}) + (n - q - 2)\hat{\sigma}^2\right]\right\}. \tag{2.9}$$

Now, the objective is to update our beliefs of the function in light of the data that we have have seen. In equation (2.7), we saw our marginal distribution of the function given the data, as well as its two parameters ($f(.)|\boldsymbol{y}, \boldsymbol{\beta}, \sigma^2$), however, we are interested in $f(.)|\boldsymbol{y}$. Therefore, we need to combine equations (2.7) and (2.9), then integrate over β and σ^2 to get the required form.

Firstly, we can integrate over β by noting from equation (2.9) that

$$\beta|\sigma^2, \boldsymbol{y} \sim N(\hat{\beta}, \sigma^2(H^T A^{-1} H)^{-1}).$$

Using Bayes theorem to combine the likelihood and our posterior, then integrating over β gives

$$f(\mathbf{x}^*)|\sigma^2, \boldsymbol{y} \sim N(\mu^{**}(\mathbf{x}^*), \sigma^2 c^{**}(.,.)),$$

where

$$\begin{aligned}\mu^{**}(\mathbf{x}^*) &= h(\mathbf{x}^*)^T\hat{\beta} + t(\mathbf{x}^*)^T A^{-1}(\boldsymbol{y} - H\hat{\beta}),\\ \sigma^2 c^{**}(\mathbf{x}, \mathbf{x}') &= \sigma^2(c(\mathbf{x}, \mathbf{x}') - t(\mathbf{x})^T(A)^{-1}t(\mathbf{x}')) + (h(\mathbf{x})^T - t(\mathbf{x})^T A^{-1}H)(H^T A^{-1}H)^{-1}\\ &\quad \times (h(\mathbf{x}')^T - t(\mathbf{x}')^T A^{-1}H)^T).\end{aligned} \tag{2.10}$$

Another way of proving this is to use the identities $\mathbb{E}(Y) = \mathbb{E}_X(\mathbb{E}_{Y|X}(Y|X))$ and $\text{var}(Y) = \text{var}_X(\mathbb{E}_{Y|X}(Y|X)) + \mathbb{E}_X(\text{var}_{Y|X}(Y|X))$, where in our case $Y|X = f(.)|\beta, \sigma^2, \boldsymbol{y}$ and $X = \beta|\sigma^2, \boldsymbol{y}$.

Now, we need to combine $\sigma^2|\boldsymbol{y}$ and $f(\mathbf{x}^*)|\sigma^2, \boldsymbol{y}$ and integrate over σ^2 to get the desired result. Looking at equation (2.9), it can be seen that

$$\sigma^2|\boldsymbol{y} \sim (n-q-2)\hat{\sigma}^2\chi^{-2}_{n-q},$$

which is a scaled inverse-chi distribution. Therefore we need:

$$\pi(f(\mathbf{x}^*)|\boldsymbol{y}) = \int_0^\infty \pi(f(\mathbf{x}^*)|\boldsymbol{y}, \sigma^2)\pi(\sigma^2|\boldsymbol{y})d\sigma^2.$$

As both $\pi(f(\mathbf{x}^*)|\boldsymbol{y}, \sigma^2)$ and $\pi(\sigma^2|\boldsymbol{y})$ are known distributions, we can write:

$$\begin{aligned}\pi(f(\mathbf{x}^*)|\boldsymbol{y}) \propto \int_0^\infty &\frac{((n-q-2)\hat{\sigma}^2)^{\frac{(n-q)}{2}}}{2^{\frac{n-q}{2}}\Gamma\frac{(n-q)}{2}}(\sigma^2)^{\frac{1}{2}(n-q+3)}c^{**}(\mathbf{x}^*, \mathbf{x}^*)^{-\frac{1}{2}}\\ &\times \exp\left\{\frac{(n-q-2)\hat{\sigma}^2}{2\sigma^2} + \frac{(f(\mathbf{x}^*) - \mu^{**}(\mathbf{x}^*))^2}{2c^{**}(\mathbf{x}^*, \mathbf{x}^*)\sigma^2}\right\}d\sigma^2.\end{aligned}$$

If we say

$$Z^2 = \frac{(f(\mathbf{x}^*) - \mu^{**}(\mathbf{x}^*))^2}{c^{**}(\mathbf{x}^*, \mathbf{x}^*)\hat{\sigma}^2},$$

then

$$\begin{aligned}\pi(f(\mathbf{x}^*)|\boldsymbol{y}) &\propto \int_0^\infty \frac{((n-q-2)\hat{\sigma}^2)^{\frac{(n-q)}{2}}}{2^{\frac{n-q}{2}}\Gamma\frac{(n-q)}{2}}(\sigma^2)^{\frac{1}{2}(n-q+3)}c^{**}(\mathbf{x}^*, \mathbf{x}^*)^{-\frac{1}{2}}\\ &\quad \times \exp\left\{\frac{(n-q-2)\hat{\sigma}^2}{2\sigma^2} + \frac{\hat{\sigma}^2 Z^2}{2\sigma^2}\right\}d\sigma^2\\ &= \int_0^\infty \frac{((n-q-2)\hat{\sigma}^2)^{\frac{(n-q)}{2}}}{2^{\frac{n-q}{2}}\Gamma\frac{(n-q)}{2}}(\sigma^2)^{\frac{1}{2}(n-q+3)}c^{**}(\mathbf{x}^*, \mathbf{x}^*)^{-\frac{1}{2}}\exp\left\{\frac{(n-q-2+Z^2)\hat{\sigma}^2}{2\sigma^2}\right\}d\sigma^2\\ &= \frac{((n-q-2)\hat{\sigma}^2)^{\frac{(n-q)}{2}}}{2^{\frac{n-q}{2}}\Gamma\frac{(n-q)}{2}}c^{**}(\mathbf{x}^*, \mathbf{x}^*)^{-\frac{1}{2}}\left(\frac{((n-q-2+Z^2)\hat{\sigma}^2)^{\frac{(n-q+1)}{2}}}{2^{(n-q+1)/2}\Gamma(\frac{n-q+1}{2})}\right)^{-1}\\ &\quad \times \int_0^\infty (\sigma^2)^{\frac{1}{2}(n-q+3)}\left(\frac{((n-q-2+Z^2)\hat{\sigma}^2)^{\frac{(n-q+1)}{2}}}{2^{(n-q+1)/2}\Gamma(\frac{n-q+1}{2})}\right)\exp\left\{\frac{(n-q-2+Z^2)\hat{\sigma}^2}{2\sigma^2}\right\}d\sigma^2.\end{aligned}$$

In the second line of this equation, we see $\sigma^2|\boldsymbol{y} \sim (n-q-2+Z^2)\hat{\sigma}^2\chi^{-2}_{(n-q+1)}$. Therefore

$$\pi(f(\mathbf{x}^*)|\boldsymbol{y}) \propto \frac{((n-q-2)\hat{\sigma}^2)^{\frac{(n-q)}{2}}}{2^{\frac{n-q}{2}}\Gamma\frac{(n-q)}{2}} c^{**}(\mathbf{x}^*,\mathbf{x}^*)^{-\frac{1}{2}} \left(\frac{2^{(n-q+1)/2}\Gamma(\frac{n-q+1}{2})}{((n-q-2+Z^2)\hat{\sigma}^2)^{\frac{(n-q+1)}{2}}} \right)$$
$$\propto \frac{((n-q-2)\hat{\sigma}^2)^{\frac{(n-q)}{2}}}{c^{**}(\mathbf{x}^*,\mathbf{x}^*)^{\frac{1}{2}}} \left((n-q-2+Z^2)\hat{\sigma}^2\right)^{-\frac{(n-q+1)}{2}},$$

due to proportionality. Now, we can write

$$\begin{aligned}
\pi(f(\mathbf{x}^*)|\boldsymbol{y}) &\propto \frac{((n-q-2)\hat{\sigma}^2)^{\frac{(n-q)}{2}}}{c^{**}(\mathbf{x}^*,\mathbf{x}^*)^{\frac{1}{2}}}((n-q-2+Z^2)\hat{\sigma}^2)^{-\frac{(n-q+1)}{2}} \\
&= \frac{((n-q-2)\hat{\sigma}^2)^{\frac{(n-q)}{2}}}{c^{**}(\mathbf{x}^*,\mathbf{x}^*)^{\frac{1}{2}}((n-q-2)\hat{\sigma}^2)^{\frac{(n-q+1)}{2}}} \left(\left(1+\frac{Z^2}{(n-q-2)}\right)\hat{\sigma}^2 \right)^{-\frac{(n-q+1)}{2}} \\
&\propto \frac{1}{(c^{**}(\mathbf{x}^*,\mathbf{x}^*)\hat{\sigma}^2)^{\frac{1}{2}}} \left(\left(1+\frac{Z^2}{(n-q-2)}\right)\hat{\sigma}^2 \right)^{-\frac{(n-q+1)}{2}} \\
&= \frac{1}{(c^{**}(\mathbf{x}^*,\mathbf{x}^*)\hat{\sigma}^2)^{\frac{1}{2}}} \left(\left(1+\frac{Z^2(n-q)}{(n-q-2)(n-q)}\right)\hat{\sigma}^2 \right)^{-\frac{(n-q+1)}{2}}.
\end{aligned}$$

Defining $K^2 = \frac{Z^2(n-q)}{(n-q-2)}$, we have

$$\pi(f(\mathbf{x}^*)|\boldsymbol{y}) \propto \frac{1}{(c^{**}(\mathbf{x}^*,\mathbf{x}^*)\hat{\sigma}^2)^{\frac{1}{2}}} \left(\left(1+\frac{K^2}{(n-q)}\right)\hat{\sigma}^2 \right)^{-\frac{(n-q+1)}{2}}.$$

This is proportional to the non-standardised univariate t-distribution (e.g. Gelman et al. 2013) in which:

$$\begin{aligned}
\mu^{**}(\mathbf{x}^*) &\text{ is the location parameter,} \\
\frac{(n-q-2)\hat{\sigma}^2 c^{**}(\mathbf{x}^*,\mathbf{x}^*)}{(n-q)} &\text{ is the scale parameter,} \\
(n-q) &\text{ is the degrees of freedom.}
\end{aligned}$$

Hence, we can write

$$\frac{f(\mathbf{x}^*)-\mu^{**}(\mathbf{x}^*)}{\sqrt{\frac{(n-q-2)\hat{\sigma}^2 c^{**}(\mathbf{x}^*,\mathbf{x}^*)}{(n-q)}}}|\boldsymbol{y} \sim t_{n-q}. \tag{2.11}$$

That is, we can see that any point in the function, after it has been standardised, has a student t-distribution with $n-q$ degrees of freedom. We can see from equation (2.11) that we have a quick way to sample a particular point from the posterior distribution of our function. This is because from equation (2.10), we have simple terms that do not involve $f(.)$ and we know random samples can easily be drawn from t_{n-q}.

2.2 Considerations when modelling with a Gaussian process

As we stated earlier, there are multiple considerations we must make regarding our prior beliefs about the Gaussian process, namely the choice of the functional form for the mean function and the choice of covariance function. In the subsequent subsections, we explore the effect that these choices have on our posterior inference. To explore these choices, we shall set up a test function, and examine the posterior distribution that we get when we observe the function at a specific number of points. The test function that will be used is

$$f(x) = \cos(x) + 3\sin(x) + x \qquad x \in (-2, 12). \tag{2.12}$$

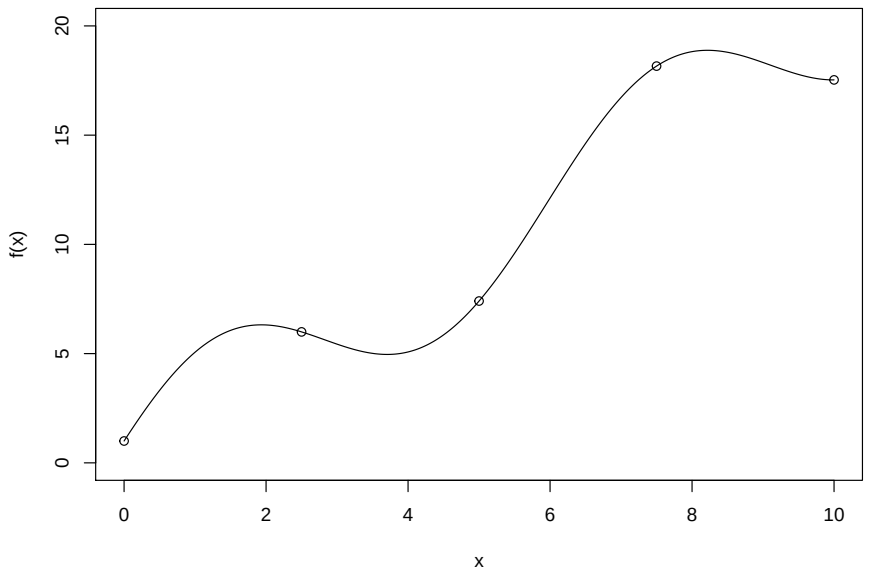


Figure 2.1: The test function from equation (2.12). The true function is in black, and we highlight the observed points using circles.

The test function is set up to be a smooth function so that a Gaussian process is appropriate. We evaluate the test function at the points $\mathbf{x} = (0, 2.5, 5, 7.5, 10)$, which are equally spaced points in the input space. It is easy to calculate that the output of the function at these locations are $D = (1, 5.994, 7.407, 18.161, 17.529)$. A figure showing the test function and the observed points can be seen in Figure 2.1.

2.2.1 The choice of $h(x)$

The first consideration that we look at is the form of our mean function in our prior distribution for the Gaussian process. We are free to choose any function for $h(x)$, and

these should reflect what our beliefs are about the function we are modelling (Oakley & O'Hagan 2002). Popular choices of this function tends to be polynomial regression terms, and so, for this exploration, we will see what affect that different choices of these have on our inference.

We will explore four choices of mean functions

$$\begin{aligned} h(x)^T &= (1), \\ h(x)^T &= (1, x), \\ h(x)^T &= (1, x, x^2), \\ h(x)^T &= (1, x, x^2, x^3). \end{aligned}$$

Using these mean function choices, we can look at what happens to our inference when we make our prior belief more complex. We will use the test function from equation (2.12). It can be seen that the test function does not perfectly match any of our choices of mean function. To test how well the Gaussian process is performing, we will follow the advice given in Bastos & O'Hagan (2009) and use the standardised mean squared error (SMSE) as our choice of diagnostic. The standardised mean squared error has the form

$$SMSE(x_i) = \frac{|y_i - E(\nu(x_i)|\mathbf{y})|}{\sqrt{Var(\nu(x_i)|\mathbf{y})}}. \tag{2.13}$$

The paper suggested that values of greater than two indicate a poor fit for the Gaussian process compared to the true function. For our test set, we will use 1,000 equally spaced points between -2 and 12, meaning that we will be performing both extrapolation and interpolation. We will also make the decision of using a squared exponential covariance function, whilst selecting the parameter b by maximising the marginal likelihood for it. That is, we use the value of b which maximises the equation

$$\log \pi(b; \mathbf{y}, \mathbf{X}) = -\frac{1}{2}\mathbf{y}^T A^{-1}\mathbf{y} - \frac{1}{2}\log(|A|) - \frac{n}{2}\log(2\pi).$$

In Figure 2.2, we can see plots showing the posterior distributions of the function using the different choices of $h(x)$ after we have realised the function at five locations.

It can be seen that for the interpolation, the four choices do not seem to differ a large amount, with the most noticeable difference being the width of the 95% credible intervals. When extrapolating however, we can begin to see a noticeable difference in the inference. Not only do we again see the length of the confidence bands differ to a greater extent, but we also see that the shape of the posterior mean of the function differs greatly. It is easy to see that as we extrapolate further from the observed data points, we tend towards a prediction that matches that of just the prior mean function. For example, in the top graph, which is when we are using just an intercept term, the prediction is tending towards

the overall mean of the observed data points. This is highlighted when we consider the pointwise SMSE of all of the functions, which can be seen in Figure 2.3. Again, we can see that, when interpolating, we are left with values of SMSE that are relatively low, and once we go out of the range of the data and begin to extrapolate, we start to see the SMSE rise to larger values. The challenge of extrapolation is emphasised here, we see that we can have a range of different inferences when the input is one dimensional. Once we are considering high dimensional inference, due to the curse of dimensionality, we know that unless we have a large number of data points, we would be performing extrapolation for many of our unseen locations.

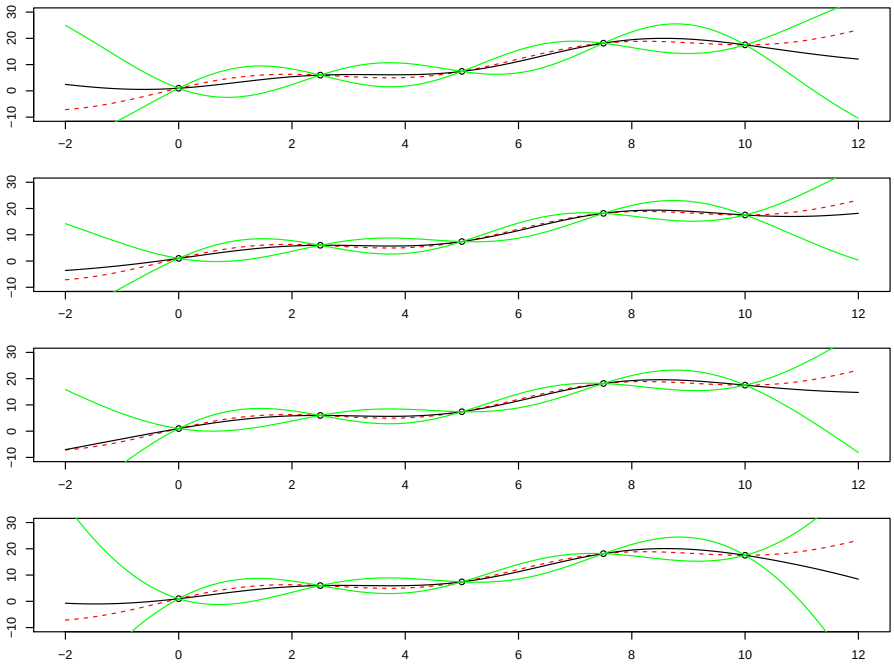


Figure 2.2: The posterior distribution of our test function using different choices of $h(x)$. The red line indicates the true value of the test function, the black is the posterior mean of our Gaussian process, and the green lines are the pointwise 95% credible intervals using the t-distribution. Top: Using just an intercept; Upper middle: Intercept and linear terms; Lower middle: Intercept, linear and quadratic terms; Bottom: Intercept, quadratic and cubic terms.

2.2.2 The choice of covariance function

We also explore the affect that the choice of the covariance function has on our posterior distribution. Now, we saw earlier that the covariance function was an important consideration as it encodes our belief about the smoothness of the underlying function. When data are sparse and we are not able to make many observations of the function, the covariance function and the smoothness that it implies is crucial to the performance of the Gaussian

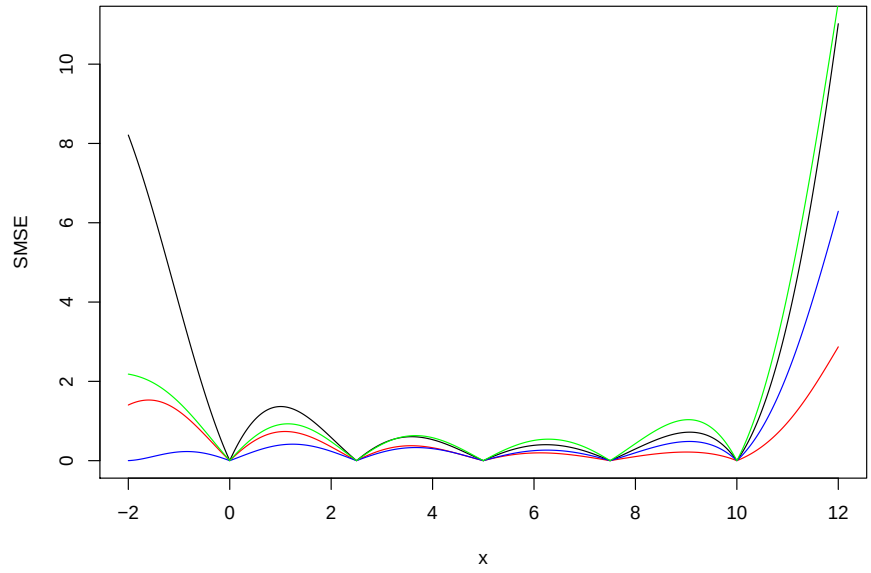


Figure 2.3: The standardised mean squared error of each of the $h(x)$ choices investigated in Figure 2.2. Black: intercept, Red: intercept and linear, blue: intercept, linear and quadratic, green: intercept, linear, quadratic and cubic. It can be seen that both the linear fit (red) and quadratic (blue) perform better than the other choices in terms of SMSE, certainly when considering extrapolating beyond the values of the observations that we have observed. This is to be expected as these choices will be able to model the linear term in the test function more accurately than the other choices.

process. It should be emphasised again that we should use any prior knowledge that we have about the smoothness of the underlying function to help to decide on our covariance function. For this test, we will use the test function in equation (2.12), observing it at the five locations, as stated earlier. We will test three popular choices of covariance functions, which are, the squared exponential covariance function, which we saw in equation (2.3), and we will also look at the Matérn covariance function when we set $\nu = \frac{3}{2}$ and $\nu = \frac{5}{2}$. The Matérn covariance function has the form

$$C_\nu(x, x') = \frac{\sigma^2 2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{|x - x'|}{b} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{|x - x'|}{b} \right),$$

where $K_\nu(\cdot)$ is the modified Bessel function of the second kind (Andrews & Andrews 1992). It should be noted that when $\nu = \frac{1}{2}$, we have the exponential covariance function. We can think of the parameter ν as controlling the smoothness of the covariance function. To be more precise, the sample path of a random process with a Matérn covariance function is $\lceil \nu \rceil - 1$ times differentiable. It is also worth noting that as $\nu \to \infty$, the Matérn covariance function converges to the squared exponential function. The Matérn covari-

ance function tends to be recommended over the squared exponential covariance function if the underlying function is not smooth or contains heterogeneity due to the smoothness (and infinite differentiability of the sample paths produces by use of this covariance function) that the use of a squared exponential function incurs. For each of the covariance functions, similar to Section 2.2.1, we will estimate the parameter b by maximising the marginal likelihood and we will chose a linear function for $h(x)$.

Figures 2.4 and 2.5 show the posterior distribution of the Gaussian process and the pointwise SMSE respectively. There does not appear to be a large difference between the posterior distributions when we consider the means of the Gaussian processes produces. The squared exponential function does indeed appear to have a slightly smoother mean function than the Matérn covariance functions. We can also observe that the lengths of the 95% confidence intervals for the Matérn covariance functions are wider than that of the squared exponential counterpart. It can also be noticed that there is no standout best method when considering the performance of the Gaussian process with respect to the pointwise SMSE. This is due to the fact that our test function is smooth and so all of the covariance functions are suitable choices for this model as the SMSE is below the value of two for all interpolation values. In terms of extrapolation, it can be seen that we get large values of SMSE for values of x which are outside the range that we have already observed, and for extrapolating the use of a linear term or a linear and quadratic term is recommended. We will see later in Chapter 5 the effect that the covariance function has when we are using a Gaussian process on a function that contains discontinuities, contains heterogeneity or is non-smooth.

Whilst we are considering the choice of covariance functions, it is worth noting a possible change that could be made in certain scenarios. Up to now, we have considered the case where the data are deterministic, however, this is not always the case, and we may only be able to sample from the function with error. If that is the case, it is possible to use a Gaussian process that does not interpolate the data exactly and instead account for this error. We do so by adapting equation (2.2), and instead using the equation

$$\text{Cov}(f(x), f(x')|\sigma^2) = \sigma^2 \left(c(x, x') + \delta_{x,x'}\sigma_\epsilon^2\right), \tag{2.14}$$

where σ_ϵ^2 is an error term and $\delta_{x,x'}$ is the Kronecker delta function that is one when $x = x'$ in our set of training data and zero in all other cases. Here we have another form of randomness in our statistical model, which is the noise in the data. To estimate σ_ϵ^2, a prior distribution is assigned to the parameter, with the MLE of its posterior distribution typically used to estimate the value of it. We can show a brief example so that we can visualise the effect that this has on our posterior inference by using the test function. In Figure 2.6, we can see the posterior distributions of the test function when a squared exponential covariance function is used with and without the additive noise term. It can

be seen that the case in which we use the noise term, the posterior mean of the Gaussian process does not interpolate the points that we have observed, and we do not see the length of the credible interval shrink to zero. Instead, we see that the posterior mean function attempts to create a smooth function that does not interpolate the points and the interval widths are inflated around the observed points to emphasise the uncertainty we have around these points.

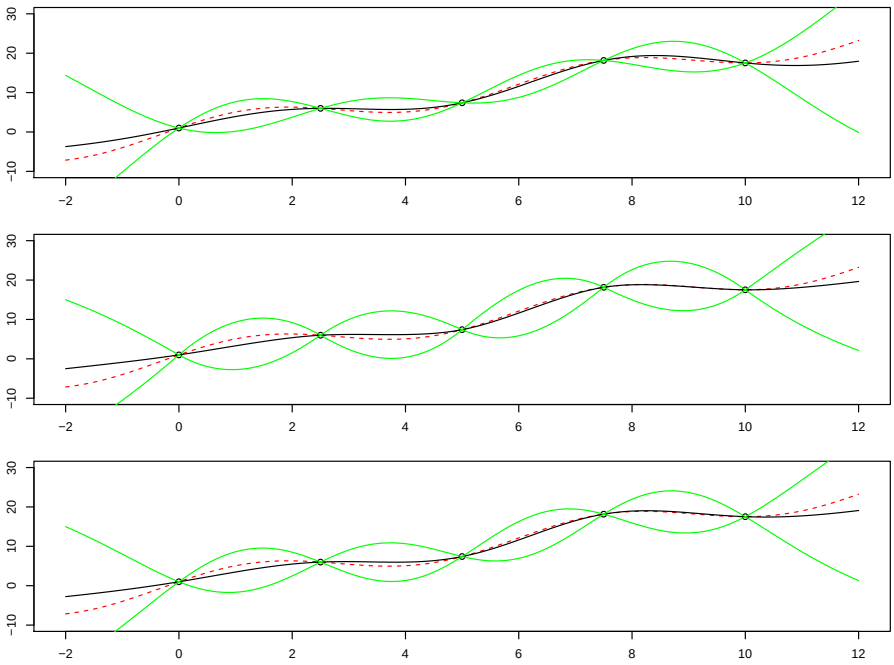


Figure 2.4: The posterior distribution of our test function using different choices of covariance functions. The red line indicates the true value of the test function, the black is the posterior mean of our Gaussian process, and the green lines are a 95% credible interval. Top: Using the squared exponential covariance function; Middle: Using the Matérn covariance function with $\nu = 3/2$; Bottom: Using the Matérn covariance function with $\nu = 5/2$.

2.2.3 The length scale of the covariance function

In Sections 2.2.1 and 2.2.2, we found the length scale parameter b, which determines how close two points need to be to be considered correlated, using the maximum of the marginal likelihood. However, it is important that we emphasise the affect that the parameter b has on our inference, and why it is important that we gain a good estimate for it. Again, any prior information that we have about this parameter should be incorporated in the form of a prior distribution for an analysis. There are an abundance of methods that are available that we can use to estimate the parameter, should we wish the data to drive the estimation of this, such as maximising the marginal likelihood and attempting to

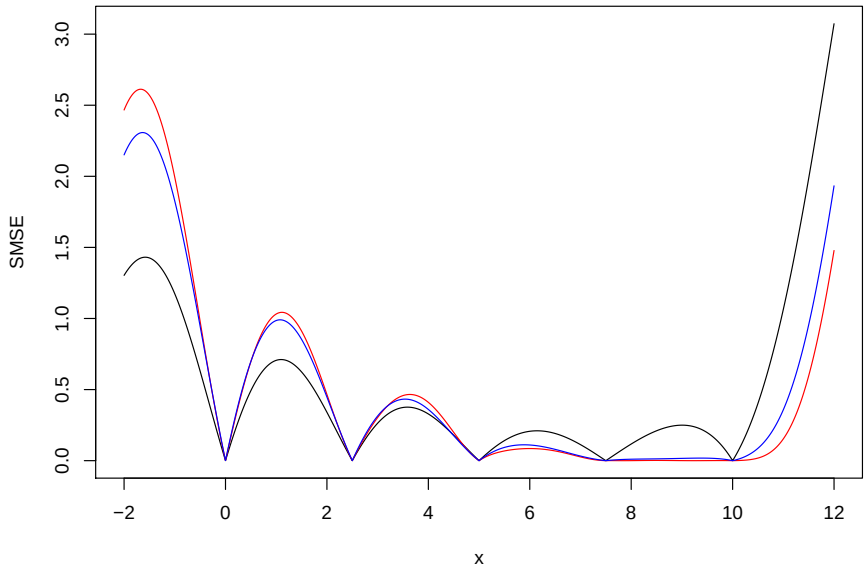


Figure 2.5: The standardised mean squared error of each of the covariance function choices investigated in Figure 2.4. Black: Squared exponential, Red: Matérn 3/2 ; blue: Matérn 5/2. It can be seen that both of the Matérn covariance perform very well compared to the exponential covariance function for the larger values due to the larger variances that they produce, shrinking the SMSE towards zero, whilst their mean value is also close to the true value of the test function.

minimise the leave-one-out cross validation metric (Rasmussen & Williams 2006).

For the analysis, we will use the same test function as we have previously been using, a linear function for $h(x)$, and our covariance function of choice will be the squared exponential covariance function. Four values of b will be observed, $b = 0.1, b = 2.5, b = 7.5$, and $b = 30$. These represent choices of prior smoothness that range from not very smooth ($b = 0.1$), to extremely smooth ($b = 30$), and we can begin to see the impact that these choices have on our posterior distribution by looking at Figure 2.7. It can be seen that when the value of b is very small, the posterior mean of the Gaussian process converges to a linear regression line, which is the functional form that was used in our prior. It is also noticeable that the lengths of the 95% credible interval quickly converges to the value of a 95% credible interval that we would see if we performed linear regression. As we increase the value of b, we can see that the posterior mean of the Gaussian process moves away from the linear regression form and interpolates the points more smoothly. The effect of this smoothness is much more noticeable when we consider the extrapolation for the larger values. When b was small, we see the posterior mean converge to a linear regression very quickly, but as b increases, we see that the smooth decrease between the final two points has a longer lasting effect and converges much more slowly to the linear

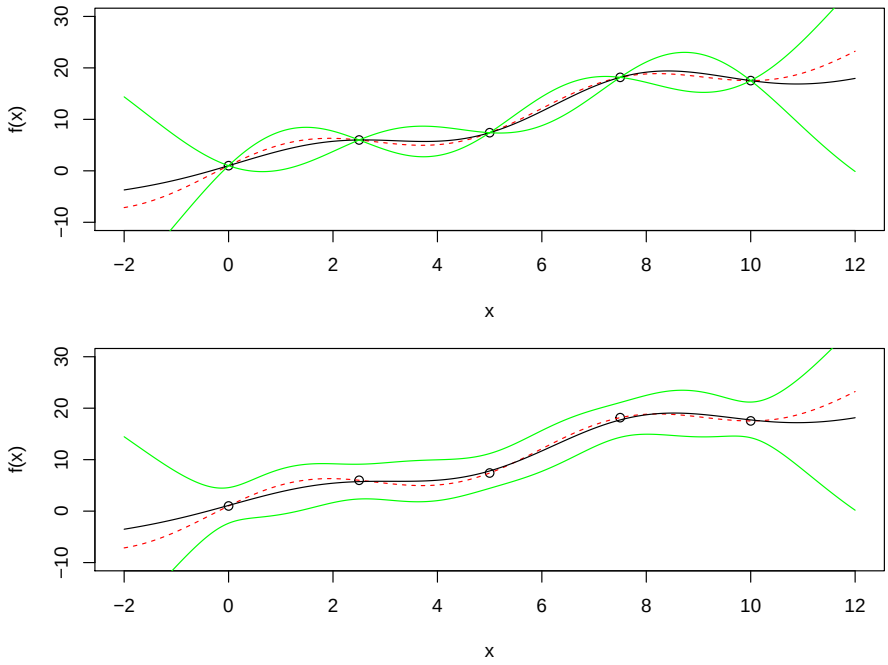


Figure 2.6: The posterior distribuion of the test function without (top) the noise term, and with (bottom) the noise term. We have the true test function in red, the posterior mean of the Gaussian process in black, and the 95% credible interval in green.

regression line.

The differing quality of prediction is evident when we consider the pointwise SMSE in Figure 2.8. We can see that the largest value of b begins to become a poor predictor for values of x larger than 2.5 when considering SMSE as our measure. The narrow credible interval widths contribute to this poor prediction; the posterior distribution when b is this large suggests that we should be extremely confident in our prediction, when in fact it appears that we seem to be overconfident in these areas. The two middle values of b perform better than the more extreme values, certainly when we consider the area of space in which we use interpolation.

2.3 Selection of design points

Now, one consideration that needs to be addressed is the choice of design points, $\mathbf{x} = (x_1, \ldots, x_n)$, that we will use to obtain our sample of the function and, hence, update our beliefs about the function. In some situations, such as when we are performing inference on a computer model (Sacks et al. 1989), we are able to select the locations $\mathbf{x}$ that we observe the function at. In situations such as these, we should explore how important it is that we choose 'good' locations. There is the option of randomly generating the locations $\mathbf{x}$ from a distribution, for example we could use a uniform distribution. However, this is not advisable as there is the possibility that we do not explore the input space adequately.

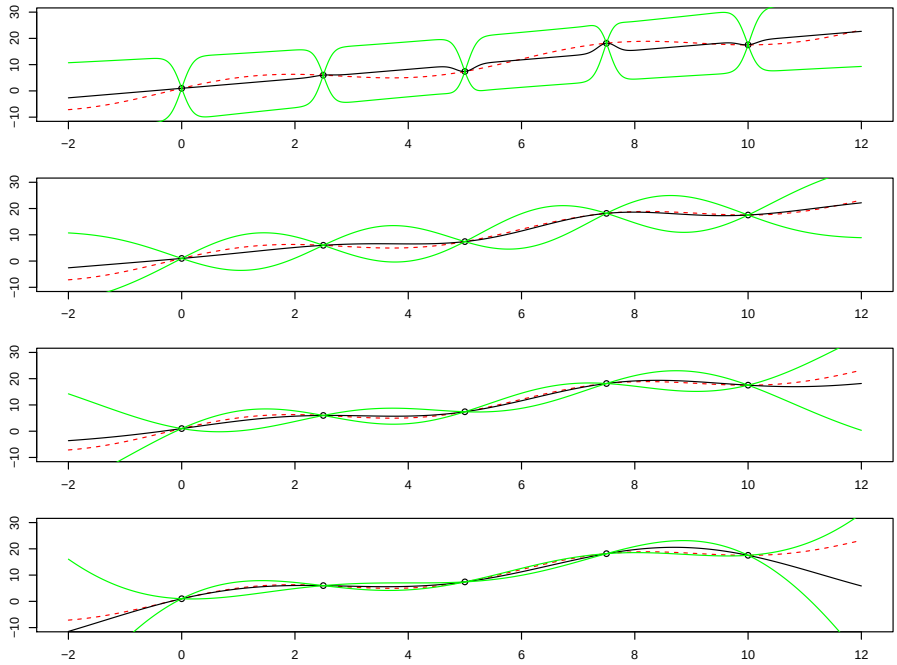


Figure 2.7: The posterior distribution of our test function using different choices of length scale values in the squared exponential covariance function. The red line indicates the true value of the test function, the black is the posterior mean of our Gaussian process, and the green lines are a 95% credible interval. Top: $b = 0.1$; Upper middle: $b = 2.5$; Lower middle: $b = 7.5$; Bottom: $b = 30$.

We may also have the situation in which we are limited with the number of times that we can realise the function f, which could be due to the monetary costs of running the experiment, the ethics, or the cost in terms of time. When this is the case, we need to ensure that we are maximising the amount of information that each point provides us. Of course, we leave the phrase 'amount of information' purposefully ambiguous as there may be different objectives dependent on the aims of the study (e.g. we may want to have the smallest total amount of uncertainty possible in a certain area of parameter space or over the space as a whole).

One of the most popular methods of selecting the initial parameter locations is to use a Latin hypercube design (LHD) (Park 1994). If we wish to sample f at n parameter values $\mathbf{x}$ initially, the LHD selects these parameter values by partitioning the parameter space such that we form a grid using $n - 1$ equally spaced lines parallel the parameter axis in each parameter dimension. n parameter values are then selected so that there is one and only one parameter value $\mathbf{x}$ in each of the n parameter axis partitions. A simple two-dimensional example of a valid LHD can be seen in Figure 2.9. Now, of course, there are many possible valid LHD, and, heuristically, we could think of preferring to pick one that explores as much space as possible. We therefore use the criterion of maximising the minimum distance (maxi-min) to select an optimal hypercube. An example of a hypercube

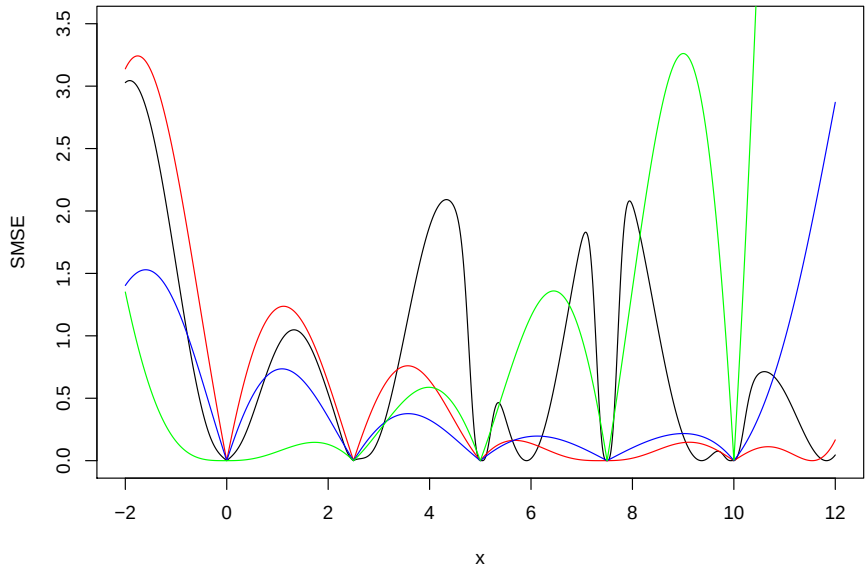


Figure 2.8: The standardised mean squared error of each of the choices of B investigated in Figure 2.7. Black: $B = 0.1$, Red: $B = 2.5$; Blue: $B = 7.5$; Green: $B = 30$. Here we can see that for both $B = 0.1$ and $B = 30$, we have SMSE values larger than two for values of x in which we are interpolating, suggesting that these values would not be good choices for this parameter. It can be seen that values of $B = 2.5$ and $B = 7.5$ both show acceptable SMSE values for this range and hence would be suitable choices for this parameter.

that would optimise this can be seen in Figure 2.9. It is possible to calculate an 'optimal' Latin hypercube analytically, however when the dimensions become large, this becomes computationally heavy. In practice, the quick and typical way of choosing an 'optimal' hypercube is to generate a large number of random LHDs, and select the design from this sample that maximises the maxi-min criterion. It is worth noting that as the dimensions increase, the space also increases with it. This is relevant as, for example, five points in one dimension will not cover as much area as 25 ($5^2 = 25$) points in two dimensions. As such, when dimensions are larger, we will have to use a much larger number of points to achieve the same coverage of the input space at lower dimensions. This is also very relevant when considering the condition number of the matrix A as, when we have a large number of points in low dimensions, we can have numerical errors when trying to invert the matrix. We find that when we use a large value for our length scale parameter, which results in two points becoming more correlated than a small length scale, our condition number becomes large. When the number of dimensions grow and we do not achieve the same coverage with the same number of points, this becomes less of a problem. A general rule of thumb is that we should use at least ten points per dimension to gain a reasonable

estimate for your Gaussian process.

Again, we can explore the effect that the locations of the input have on our inference using our toy example from equation (2.12). We do this by comparing using a uniform distribution to decide where we run the test function to choosing a space filling design equating to an equally spaced points design. We can see from Figure 2.10 that we can tend to see a cluttering of points when we just randomly select where our input locations. As we can see from the plots, we see areas of large uncertainty when the uniform sampling is used compared to when we use the space filling design. If the objective is prediction, then we can see that all of the designs perform reasonably, however, if the objective is to universally reduce uncertainty, then some of the designs are particularly poor, the last two designs in particular are examples of this.

2.4 Conclusions

In this chapter, we provided an introduction to the Gaussian process methodology and explored how this distribution can be used as a prior distribution for an unknown function $f(\cdot)$. We have seen the considerations that need to be made for our choices of hyper-parameters and covariance functions, that all play a significant role in the resulting inference in practical purposes. These results and features will be useful for later chapters in which we look to tackle problems that are faced when applying the Gaussian process to real life scenarios.

We have seen one aspect of our confluence, which was the use of a Gaussian process for emulation, and have observed its ability to provide a powerful prior distribution when data is sparse. The next chapter introduces the wavelet methodology, which is noted for its ability to model discontinuities well — something that the Gaussian process we explored in this chapter does not.

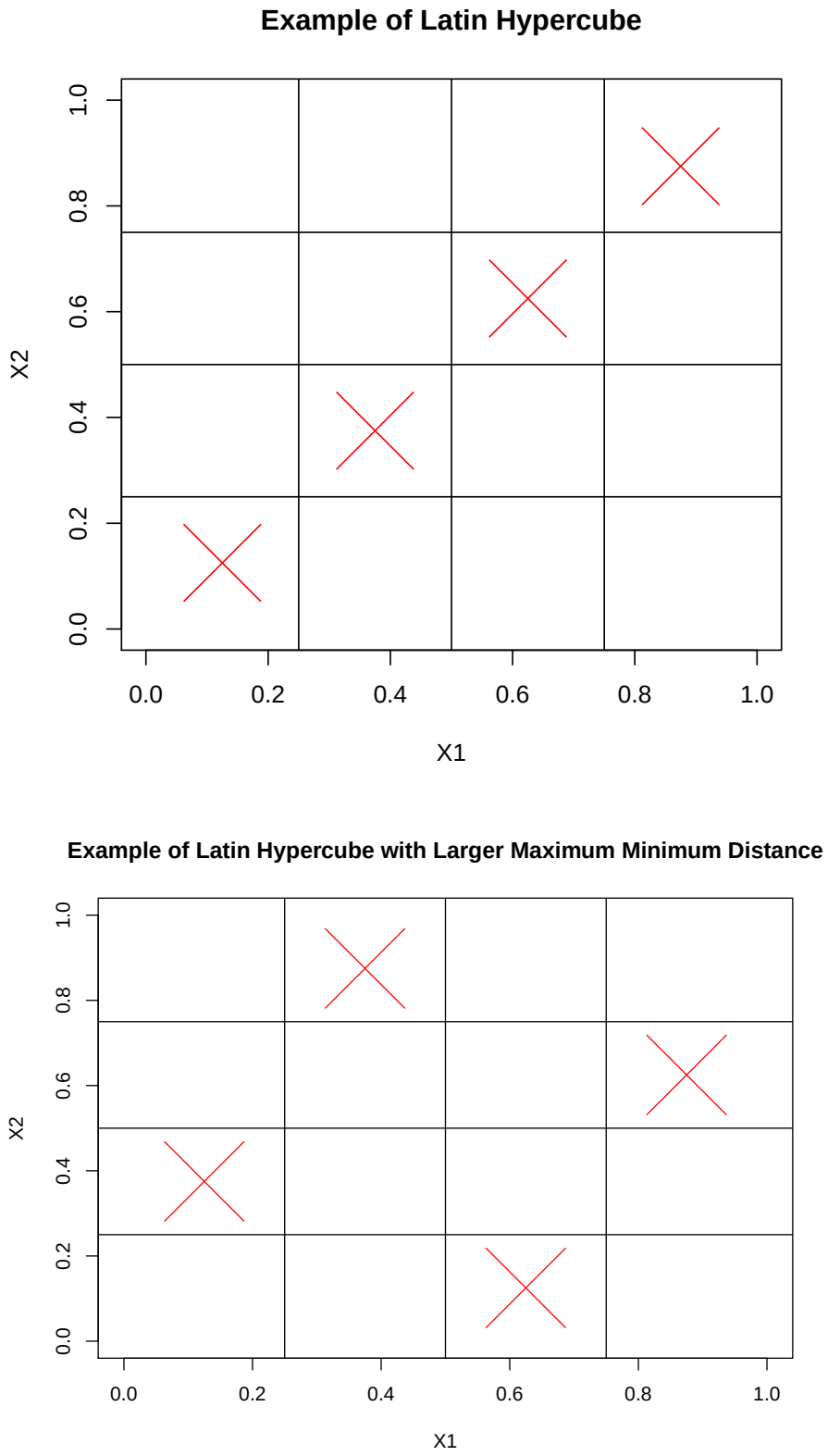


Figure 2.9: Top: A simple example of a valid Latin hypercube in two dimensions. Bottom: An example of an optimal Latin hypercube in terms of a maxi-min criterion

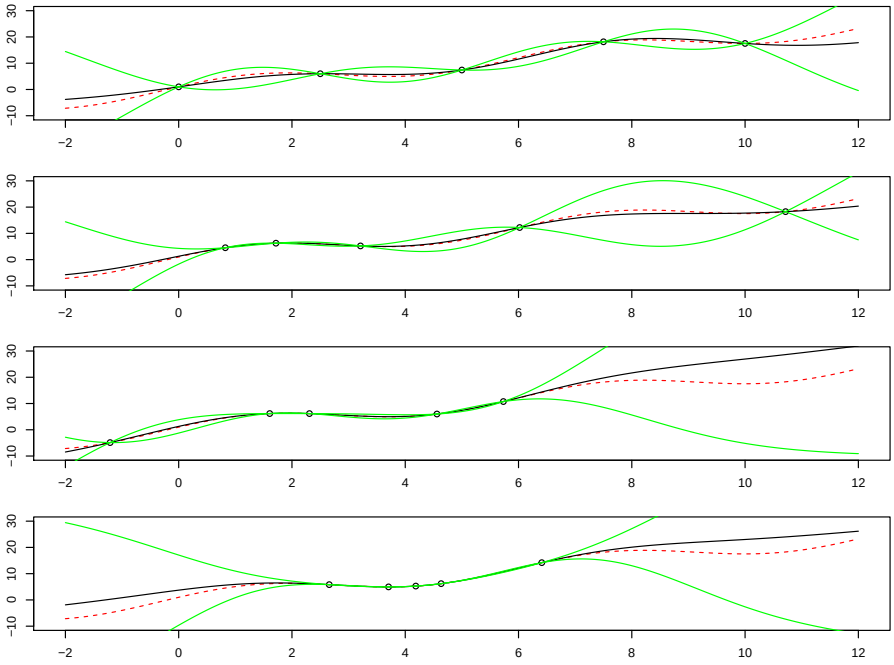


Figure 2.10: We show the test function with a space filling design (Top) and three examples of the test function if we were to choose the locations of the input randomly.

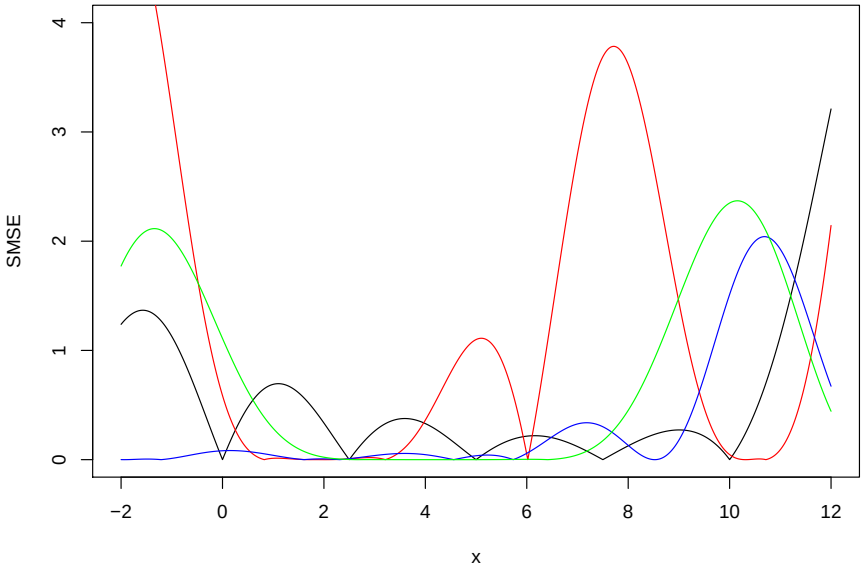


Figure 2.11: The SMSE of the test in which we use different locations for our inputs from Figure 2.10. Black: Top example, Red: upper middle example, Blue: Lower middle example, Green: Bottom example.

Chapter 3

Introduction to wavelets

3.1 Wavelet analysis

Wavelets are a multi-scale method used to represent a function, and in this chapter, we present an introduction to the methodology and their usage in statistics. By multi-scale, we are referring to the ability to represent an object at a set of frequencies or scales. Wavelets are wave like functions that oscillate and are, generally, compactly supported. They can be used to form basis for a multitude of function spaces, and, due to their construction, as we will explore in this chapter, can be informative when considering the activity of the function at a particular frequency. This information is similar to the information that we obtain through using a Fourier analysis (e.g. Stein & Weiss 2016). Using wavelets, however, allows us to also find localisation details for the frequency (e.g. Nason 2010). The reasons as to why we can do this, and in what situations this is advantageous, will be seen in this chapter. Wavelets popular uses include wavelet shrinkage, which will be discussed in more detail in Section 3.8, time series analysis (e.g. Percival & Walden 2006), density estimation (e.g. Walter & Shen 2018), image restoration (e.g. Mallat 1999), and many more.

Firstly, we can look at the wavelet representation of a function. Consider the set of functions in the L-2 space, that is, we consider those $f \in \mathbb{L}^2(\mathbb{R})$. The $\mathbb{L}^2(\mathbb{R})$ is the space of square integrable function on $\mathbb{R}$. These functions are defined as functions that satisfy the relation

$$\int_{-\infty}^{\infty} |f(x)|^2 dx < \infty.$$

Now, a wavelet is a function $\psi \in L^2(\mathbb{R})$ that satisfies an admissibility condition (Misiti et al. 2013) in the frequency domain, which is

$$\int \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty, \tag{3.1}$$

where $\hat{\psi}$ is the Fourier transform of ψ. In most literature on wavelets, ψ is often referred to as the *mother wavelet*. A consequence of equation (3.1) is

$$\int \psi(t)dt = 0,$$

with the function rising and falling below zero with equal density. Wavelets are commonly classified by the number of vanishing moments they possess (Nason 2010). A wavelet ψ is said to have m vanishing moments if it satisfies

$$\int x^j \psi(x)dx = 0 \quad \text{for } j = 0, \ldots, m-1.$$

In Section 3.5, we explore the effect that the number of vanishing moments we use has on our analysis. Heuristically, the larger the number of vanishing moments, the typically smoother the wavelet is in a mathematical sense (in which the function is differentiable a larger number of times at any point).

From the mother wavelet ψ, we are able to create a family of functions, $\psi_{j,k}$ $j > 0, k \in \mathbb{R}$, that are scaled and translated version of ψ. These generally have the form

$$\psi_{j,k}(x) = \frac{1}{\sqrt{j}}\psi\left(\frac{x-k}{j}\right). \tag{3.2}$$

We can see from equation (3.2) that j controls the scale (or frequency) of the function and k controls the translation (or location) of the function $\psi_{j,k}$. We are particularly interested, as generally in statistics we are working with data as opposed to functions, in the dyadic scaling and translation family of any ψ in which we have the relation

$$\psi_{j,k}(x) = 2^{j/2}\psi\left(2^j x - k\right).$$

This wavelet family that we are able to form has the property that they are orthonormal to each other (Nason 2010), that is, we find that

$$< \psi_{j,k}, \psi_{j',k'} >= \int_{\mathbb{R}} \psi_{j,k}(x)\psi_{j',k'}(x)dx = \delta_{j,j'}\delta_{k,k'}, \tag{3.3}$$

where $\langle \cdot, \cdot \rangle$ is the inner product, $\delta_{x,y} = 1$ if $x = y$, and $\delta_{x,y} = 0$ if $x \neq y$.

3.2 Multiresolution analysis

As multiresolution analysis (MRA) provides the foundation in wavelet theory, as we can see in classical papers such as Mallat (1989), we must introduce the idea of it in this chapter. We can think of a subspace (or approximation space) V_j at resolution level j such that $V_j \subset \mathbb{L}^2(\mathbb{R})$. Now, we can also think of a collection of subspaces such that we have the following properties

1. $\ldots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots$,

2. $\bigcup_{j\in\mathbb{Z}} V_j = \mathbb{L}^2(\mathbb{R}); \bigcap_{j\in\mathbb{Z}} V_j = \{0\}$,

3. $f(x) \in V_j$ if and only if $f(2x) \in V_{j+1}$,

4. $f(x-c) \in V_0$ if and only if $f(x) \in V_0 \; \forall \; c \in \mathbb{Z}$.

We can see from the first point that as j becomes greater, the subspace contains a larger number of functions, and so, as $j \to \infty$, the subspace contains the whole $\mathbb{L}^2(\mathbb{R})$ space. Equivalently, as j becomes a larger negative, the number of functions contained in the subspace reduces and as $j \to -\infty$, we are left with the zero space.

We can also see how the subspaces interact with each other — from the third property we can see that if we have the same function $\phi \in L^2(\mathbb{R})$, and the function's frequency is doubled, then the function that varies more quickly lives in the subspace up from the original function. Due to this relationship, we know that, if we find a family of functions $\{\phi_{j,k}(x)\}_k$ that form an orthogonal basis for any V_j, then we have an orthogonal basis for all V_j's. This means that we have a multiresolution analysis of the $\mathbb{L}^2(\mathbb{R})$ space. In wavelet theory, we often refer to $\phi_{j,k}$ as the father wavelet (Jaffard et al. 2001).

Now, we may ask how the decomposition that we saw in equation (3.8) relates to our multiscale analysis. Following on from the multiresolution analysis, we can find an approximation of $f(x) \in \mathbb{L}^2(\mathbb{R})$ at level j by

$$f_j(x) = \sum_{k\in\mathbb{Z}} c_{j,k}\phi_{j,k}(x) = P_j f(x), \tag{3.4}$$

where P_j is the projection operator that spans $\{\phi_{j,k}\}_{k\in\mathbb{Z}}$ onto the approximation space V_j (Daubechies 1988). $c_{j,k}$ are often referred to as the scaling coefficients. The coefficients are found in a similar manner to the wavelet coefficients in equation (3.8), specifically, we use the equation

$$c_{j,k} =< f, \phi >= \int_{\mathbb{R}} f(x)\phi_{j,k}(x)dx. \tag{3.5}$$

If we know that $\phi(x) \in V_1$, and that $\{\phi_{1,k}(x)\}_{k\in\mathbb{Z}}$ forms an orthonormal basis for V_1, we are able to write the relation

$$\phi(x) = \sum_{i\in\mathbb{Z}} h_i\phi_{1,i}(x), \tag{3.6}$$

for some value of h_i. Equation (3.6) is known as the dilation equation, and shows the relation of father wavelet functions at different resolution levels. Daubechies (1992) explains that, as we can see from equation (3.6), a father wavelet can be constructed by a linear combination of a scaled version of itself.

We also consider the details that are lost when moving to a coarser approximation space, that is, what happens to the information we lose going from V_j to V_{j-1}. In terms of spaces, we can consider a detail space W_j such that we have $< V_j, W_j >= 0$ and we have

$$V_{j+1} = V_j \oplus W_j, \tag{3.7}$$

where we define

$$A \oplus B = \{f + g : f \in A, g \in B\}.$$

assuming that $A \perp B$.

We can see that equation (3.7) has the interpretation that the next finer level approximation space V_{j+1} is made up of the previous approximation space V_j and the lost details W_j. We can hence, similar to the V_j space, define a set of functions $\psi(x)$ such that $\{\psi(x-k)\}_{k\in\mathbb{Z}}$ is an orthonormal basis for W_0, and that these functions are orthogonal to everything in V_0. Again, similar to the MRA structure that was seen for the V_j space, we have an MRA structure for the Ws in that the spaces are scaled versions of each other, so we have

$$f(x) \in W_j \text{ if and only if } f(2x) \in W_{j+1}.$$

We again see that $\{\psi(2^j x - k)\}_{k\in\mathbb{Z}}$ forms an orthonormal basis for W_j.

Using the relation similar to equation (3.3), in which we have

$$< \phi_{j,k}, \phi_{j',k'} >= \int_{\mathbb{R}} \phi_{j,k}(x)\phi_{j',k'}(x)dx = \delta_{j,j'}\delta_{k,k'},$$

it is known that (e.g. Nunes et al. 2006) that these families of functions can be used to form an orthonormal basis for the $\mathbb{L}^2$ space. This means that for any function $f \in \mathbb{L}^2(\mathbb{R})$, a decomposition can be done using the equation

$$f(x) = \sum_{k=-\infty}^{\infty} c_{J,k}\phi_{J,k}(x) \sum_{j=-\infty}^{J}\sum_{k=-\infty}^{\infty} d_{j,k}\psi_{j,k}(x), \tag{3.8}$$

where

$$\begin{aligned} d_{j,k} &=< f, \psi_{j,k} >= \int_{-\infty}^{\infty} f(x)\psi_{j,k}(x)dx, \\ c_{j,k} &=< f, \phi_{j,k} >= \int_{-\infty}^{\infty} f(x)\phi_{j,k}(x)dx, \\ J &\in \mathbb{N}. \end{aligned}$$

For any function $f \in \mathbb{L}^2(\mathbb{R})$, we are able to define the continuous wavelet transformation (Ogden 2012) for a given wavelet ψ by

$$F(j,k) = \int_{-\infty}^{\infty} f(x)\psi_{j,k}(x)dx,$$

for $j, k \in \mathbb{R}, j \neq 0$ and where $\psi \in \mathbb{L}^2(\mathbb{R})$ satisfies equation (3.1).

Daubechies (1992) showed in her paper an explicit method, using the dilation equation, for finding functions ϕ and ψ that we can use for our MRA wavelet analysis. A classical example of functions that we can use for MRA wavelet analysis can be seen in the next section.

3.3 The Haar wavelet

The oldest (and most simple) wavelet discovered was the Haar wavelet in Haar (1910), although the term wavelet was not to be used until many years later. As we saw in the Section 3.2, we need a function ϕ that can be used as a basis for our approximation space V_j for the $\mathbb{L}^2(\mathbb{R})$ space. One such simple function that was used is the Haar scaling function, which is defined by

$$\phi(x) = \begin{cases} 1 & x \in [0,1), \\ 0 & \text{else.} \end{cases} \tag{3.9}$$

The Haar scaling function is also known as a step function. It is easy to see that $\phi(x-k)_{k\in\mathbb{Z}}$ would correspond to the function being shifted by a value of k, and that there would be no overlap between these functions when we shift them. To be more precise, we can see how the set $\{\phi(x-k)\}_{k\in\mathbb{Z}}$ can form an orthonormal set when we consider

$$< \phi(x), \phi(x-k) >= 0 \; \forall \; k \in \mathbb{N}.$$

We look in more detail at the Haar wavelet to allow us to visualise what a muliresolution analysis entails. If we consider equation (3.4), we can see that we have an equation for the approximation of a function f at level j. Using the definition of the Haar scaling function in equation (3.9), we can clearly see that if we use the Haar wavelet, then our approximation to the function at level j would be step functions whose heights are determined by the average of the function in the interval $[k, k+1)$. We can also start to visualise how the approximation spaces interlink with each other, if we wanted to look at a finer level approximation we can do so with ease. We could think of the Haar scaling function that we saw in equation (3.9) as our basis at level $j = 0$, that is, $\phi(x) = \phi_{0,0}(x)$ is our scaling function for $j = 0$ and $k = 0$. Now, to move to the next approximation level, we saw that if $f(x) \in V_j$, then $f(2x) \in V_{j+1}$, and so we could define

$$\phi_{j,k}(x) = \begin{cases} 1 & x \in \left[\frac{k}{2^j}, \frac{k+1}{2^j}\right), \\ 0 & \text{else.} \end{cases} \tag{3.10}$$

We are then able to see from equation (3.10) that we have

$$\phi_{0,0}(x) = \phi_{1,0}(x) + \phi_{1,1}(x),$$

or, similarly, we have

$$\phi_{0,0}(x) = \phi_{0,0}(2x) + \phi_{0,0}(2x-1).$$

Hence, we see that for our finer scale approximation, we use twice as many step functions, which have half the width of the step functions of the next coarser level. Now, we

have seen how the Haar wavelet can give us an approximation at any multiresolution level j, however we are also interested in finding the function that describes the difference between approximation levels. If we have two functions that describe our approximation to the function $f(x)$ at level $j+1$, that is we have $a\phi(2x) + b\phi(2x-1)$, then our approximation at level j is naturally $\frac{a+b}{2}\phi(x)$. We are interested in a basis for the approximation space W_j, we consider the information that is lost by using a coarser approximation. To see this, we consider

$$\begin{aligned} a\phi(2x) + b\phi(2x-1) - \frac{a+b}{2}\phi(x) &= a\phi(2x) + b\phi(2x-1) - \frac{a+b}{2}[\phi(2x) + \phi(2x-1)] \\ &= \frac{a-b}{2}\phi(2x) - \frac{a-b}{2}\phi(2x-1) \\ &= \frac{a-b}{2}\psi(x), \end{aligned}$$

where

$$\psi(x) = \begin{cases} 1 & x \in \left[0, \frac{1}{2}\right), \\ -1 & x \in \left[\frac{1}{2}, 1\right), \\ 0 & \text{else.} \end{cases} \tag{3.11}$$

Equation 3.11 shows that the 'lost detail' (that is, the accuracy that is lost from using a coarser approximation) between levels can be represented by a step function. The interpretation of the coefficients $d_{j,k}$ that are associated with the wavelet function $\psi_{j,k}(x)$ then becomes very natural:

We are attempting to model the function f using a basis of $\psi_{j,k}$, which is compactly supported and can have varying levels of mathematical smoothness (based on the number of vanishing moments). Hence, if the function f is very mathematically smooth locally, and we are attempting to represent this using a ψ_{jk} that has a low number of vanishing moments and is not mathematically smooth, then the basis will not provide a good approximation to this function locally and we would expect the values of $d_{j,k}$ to be large to account for this. Conversely, if our basis $\psi_{j,k}$ is a good approximation for the function f, we would expect the value of $d_{j,k}$ to be small due to the goodness of fit. We can see that if we have a function f that does not contain discontinuities and is relatively mathematically smooth, then we would expect the values of $d_{j,k}$ to be small and around zero due to the ability of ψ_{jk} to model the function f. This leads to the idea of a sparse representation of the function, with the main features of the function being encapsulated by a small number of large coefficients. (Donoho & Johnstone 1995).

3.4 The discrete wavelet transformation

As alluded to in Chapter 1, in statistics we typically work with a finite and discrete number of datapoints, as opposed to a function. As such, the discrete version of the wavelet

transformation, the discrete wavelet transformation (DWT), will be used throughout the thesis. One key element of the DWT is the ability to express a wavelet function in terms of the wavelet function at a finer resolution level, as this helps to build the full possible decomposition. Using the dilation equation from equation (3.6), and the notation which we saw in equation (3.10), we are able to write the relation

$$\phi_{j,k} = \sum_i h_i \phi_{j+1,2k+i}(x).$$

If we substitute this relation into equation (3.6), we find that

$$\begin{aligned} c_{j,k} &= \int_{\mathbb{R}} f(x) \sum_i h_i \phi_{j+1,2k+i}(x) dx \\ &= \sum_i h_i \int_{\mathbb{R}} f(x) \phi_{j+1,2k+i}(x) dx \\ &= \sum_i h_i c_{j+1,i+2k}. \end{aligned} \tag{3.12}$$

Similarly, using the results from Daubechies (1992), we find that

$$d_{j,k} = \sum_i g_i c_{j+1,i+2k}, \tag{3.13}$$

for some scalar value $g_i \in \mathbb{R}^1$.

Hence, using equations (3.12) and (3.13), we can find the value of any coarser scale mother or father wavelet coefficient using the finer scales father wavelet coefficients. This result is important as, by finding the values of g_i and h_i, we are able to use the data that we have observed to find all of the coefficients in an efficient manner due to the linearity of the calculations.

For the full DWT, the pyramid algorithm is typically used (Nason 2010); we assume that we can make N observations of the function $f(x)$ at locations $\{x_0, \ldots, x_{N-1}\}$, in which the observations are equally spaced across $\mathcal{X}$, the space of x, and that N is dyadic. That is, N has the form $N = 2^J,\ j \in \mathbb{Z}^+$. We observe the function at the locations to obtain our observations

$$\mathbf{f}^T = (f(x_0), \ldots, f(x_{N-1})).$$

We then define our N datapoints to be the finest resolution level, J, father coefficients in our wavelet decomposition, such that we have

$$c_{J,k} = f(x_k).$$

This follows as the finest resolution level that we can observe from the data is the data itself. By defining the finest resolution level coefficients, using equations (3.12) and (3.13), we are able to find all of the subsequent coarser scale mother and father wavelet coefficients. The output of the DWT of the data $\mathbf{f}^T$ is the data vector which is comprised

of

$$\mathbf{z}^T = \left(c_{0,0}, d_{0,0}, d_{1,0}, d_{1,1}, d_{2,0}, \ldots, d_{J-1,\frac{N}{2}-1}\right).$$

As the DWT is orthogonal, and, due to the way in which the wavelets are constructed,as we can see in equations (3.12) and (3.13), linear, it is possible to represent the transformation as an orthogonal matrix, W. That is, we can use an orthogonal matrix W such that

$$W^T\mathbf{f} = \mathbf{z}.$$

Now, as the transformation and the matrix is orthogonal, we know that

$$WW^T = I_N.$$

This property means that we are also able to invert the discrete wavelet transformation using a matrix — W^T. The use of this matrix to convert $\mathbf{z}$ into data $\mathbf{y}$ is commonly known as the inverse discrete wavelet transformation (IDWT).

3.5 The choice of vanishing moments

As mentioned earlier, an important consideration when considering our wavelet decomposition is selecting which wavelet function to use. We saw in Section 3.1 that wavelets can be categorised by the number of vanishing moments that they possess. To see the effect that this choice has on our decomposition, we can look at a brief example. We use the HeaviSine test function from Donoho & Johnstone (1994), which is

$$f(x) = 4\sin(4\pi x) - \text{sign}(x - 0.3) - \text{sign}(0.72 - x). \tag{3.14}$$

We can visualise what wavelets with different numbers of vanishing moments look like. We consider three examples; the first we will look at is the Haar wavelet from Section 3.3, the second we will look at is a wavelet with three vanishing moments, and, lastly, we will look at a wavelet with nine vanishing moments. It is worth noting that the Haar wavelet has one vanishing moment. We can see the scaling functions of those examples in Figure 3.1, and the wavelet functions in Figure 3.2. We can see indeed see the intuition that the wavelets with more vanishing moments are mathematically smoother than those with less vanishing moments.

For our test, we observe equation (3.14) at $n = 1024$ equally spaced locations between zero and one. We then observed the different projections (equation (3.4)) of the wavelet decomposition to see the effect that the choice of wavelet had on this. We looked at the projections at $j = 1$, $j = 3$ and $j = 7$ for the three different choices of wavelets, which can be seen in Figures 3.3, 3.4 and 3.5. We can again see from these examples that the wavelets with more vanishing moments produce a smoother resulting projection.

When deciding on the number of vanishing moments that we are going to use, we must carefully consider the mathematical smoothness of the underlying function. Figures 3.3, 3.4 and 3.5 highlight that functions which are not very smooth, for example a combination of step functions, would be best modelled by wavelets with low vanishing moments, such as the Haar wavelet. For example, let us consider the sharp increase that we see in the test function at 0.72. We can see that this increase is not smooth and is almost like a step function at this point. The fact that this is not smooth at this point leads to a rather poor estimate at this point using wavelets with more vanishing moments (Figures 3.4 and 3.5) compared to the Haar function (Figure 3.3). Vice versa, we can see that the smooth parts of the test function (which is made up of the mathematically smooth sin function) is modelled much more accurately using smoother wavelet functions.

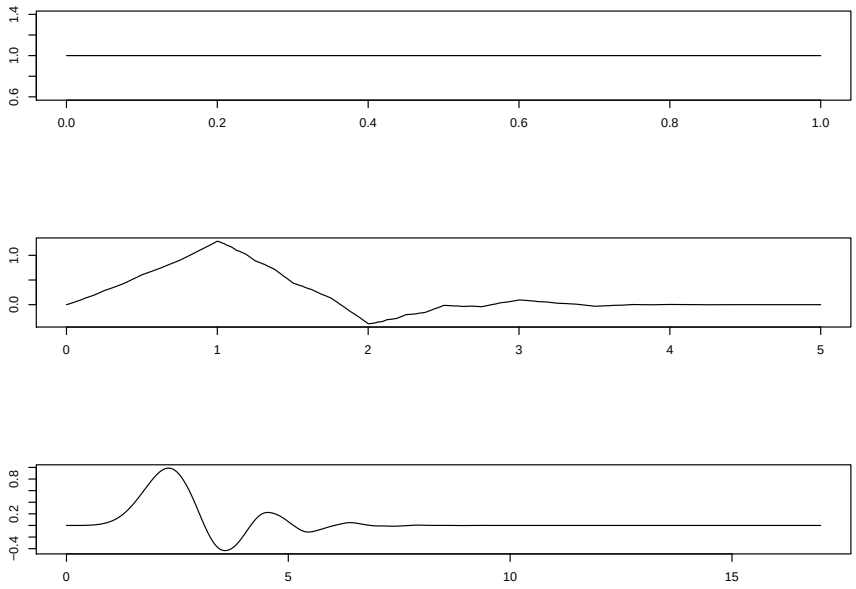


Figure 3.1: The scaling functions associated with the Haar wavelet (top), a wavelet with three vanishing moments (middle) and nine vanishing moments (bottom).

3.6 Boundary conditions

We considered the number of vanishing moments we use when choosing our wavelet decomposition. Not only do we have to think about our belief in the smoothness of the underlying function, but we also have to consider the boundary conditions. Boundary conditions are the assumptions that may need to be made that may effect those coefficients that are computed near the extreme values of our support. In the Haar wavelet, we do not need to take this into consideration. We have dyadic, equally spaced data typically, and so the Haar wavelet, which uses a power of two data points for each wavelet coefficient,

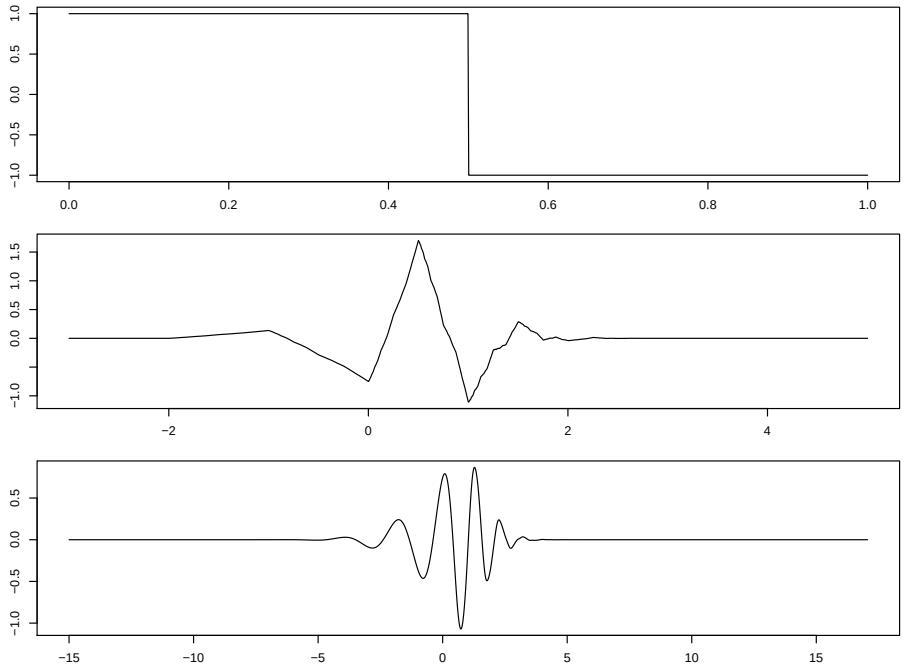


Figure 3.2: The wavelet functions associated with the Haar wavelet (top), a wavelet with three vanishing moments (middle) and nine vanishing moments (bottom).

aligns perfectly with the number of data points. For the finest level, the number of coefficients we have is half the number of data points. However, the number of data points needed to calculate a coefficient at the finest level are double the number of vanishing moments, and increases exponentially as we calculate coarser level coefficients (Nason 2010). This means that, unlike with the Haar wavelet, we are left with the situation in which we do not have enough data points to find all of our coefficients using the same strategy as the Haar, specifically, those coefficients near the boundary of the function will not have the required data to compute these. We therefore need to make another decision regarding what to do with those coefficients that do not have enough data points.

The main method that is used to deal with this problem is to use surrogate datapoints to estimate these coefficients, typically by making some assumptions. One of these assumptions that is commonly used is to assume periodicity (Ogden 2012), that is, we make the assumption that the underlying function is periodic. By making this assumption, we have a rather natural set of numbers that we can use for our surrogate data points – we use the data points that are on the opposite boundary. For example, if we had $n = 8$ data points, $\mathbf{x} = (x_1, \ldots, x_8)$, and we are using a wavelet with two vanishing moments. Our first coefficient at the finest level would be calculated using x_1, x_2, x_3 and x_4, our second using x_3, x_4, x_5 and x_6, our third using x_5, x_6, x_7, x_8, an,d finally, our fourth using x_7, x_8, x_1 and x_2.

Another popular method is again pretending that we have the data points to calculate the coefficients, but rather than assuming the underlying function is periodic, we make the

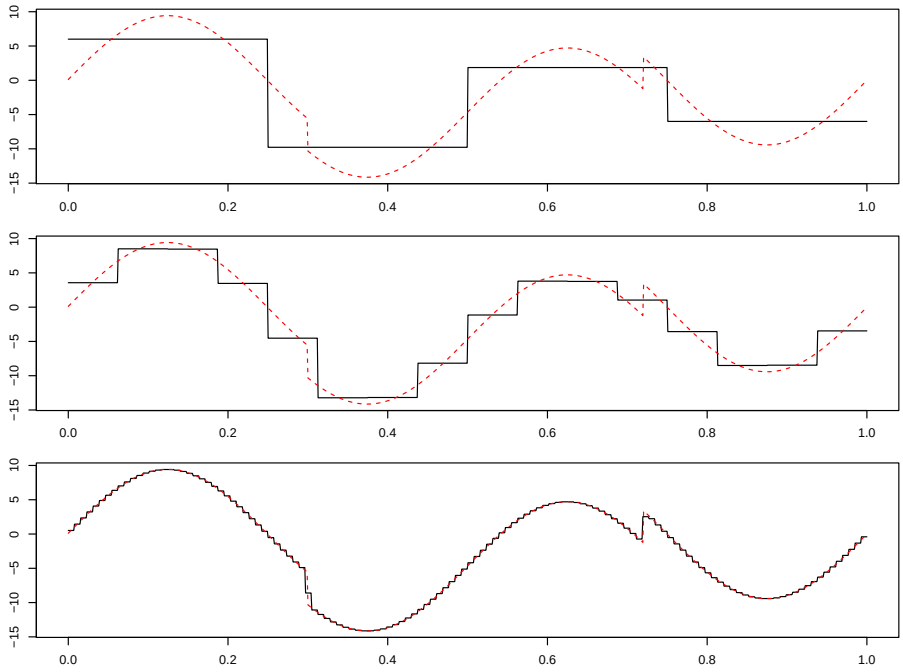


Figure 3.3: The true test function (red) and the different approximations at different levels using a Haar wavelet. Top: $j = 1$, Middle: $j = 3$, Bottom: $j = 7$.

assumption that it is symmetric (Ogden 2012). This assumption differs from the periodic assumption; in the periodic case, we presumed that the underlying function repeats itself, whereas for the symmetric case, we believe the data points are going to be just like those that we have just seen. For example, if we again have $n = 8$ data points, $\mathbf{x} = (x_1, \ldots, x_8)$, and we are using a wavelet with two vanishing moments; our first coefficient at the finest level would be calculated using x_1, x_2, x_3 and x_4, our second using x_3, x_4, x_5 and x_6, our third using x_5, x_6, x_7, x_8, and finally our fourth using x_7, x_8, x_8 and x_7.

These two popular methods can result in different decompositions, and so our beliefs about the function should be carefully considered when selecting which boundary conditions to use for our wavelet decomposition. These are of course not the only two methods, although they are the most popular methods that are used to handle the boundary conditions for this kind of solution (Nason & Silverman 1994). Of course, there are other types of solutions that we can use to get around this problem. Rather than adapting the data so that they fit the wavelets that we are using, there is the idea that we can instead adapt the wavelet so that it remains in the data domain. One popular method was discovered by Cohen et al. (1993), called 'wavelets on the interval', in which we modify wavelets using operations such as reflection, so that the wavelet functions use only the data that we have to calculate the coefficients.

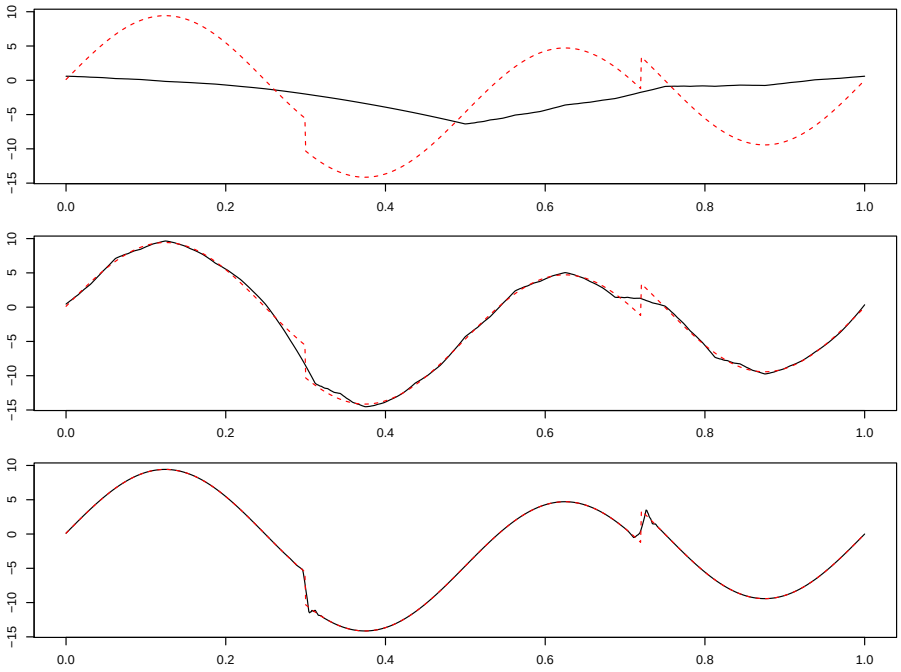


Figure 3.4: The true test function (red) and the different approximations at different levels using a Daubechies wavelet with three vanishing moments. Top: $j = 1$, Middle: $j = 3$, Bottom: $j = 7$.

3.7 The non-decimated wavelet transformation

As we saw in the earlier section, for the Haar wavelet, we find the first coefficient at the finest level by looking at the difference between the first two data points, the second coefficient at the finest level by looking at the difference between the third and fourth data points, and so forth. Now, although we know that this set-up provides us with an orthogonal transformation, we can think about possible adaptations to the method. One thing that we can consider is the information that we are missing by considering the difference between the second and third data points, the fourth and fifth data points, and so forth. If we were to instead use a transformation that takes into consideration this information as opposed to the original, we would still get an orthogonal transformation in which we can consider frequencies at different scales and locations. However, we have come full circle as we again are not considering the information that we would have from the original transformation. As such, it should therefore be noted that the standard wavelet transformation is not invariant to shift changes (Nason 2010).

Instead, we would prefer to use a method that is invariant and considers all possible information. One idea for doing this is using a non-decimated wavelet transformation. Rather than using just one of the possible transformations with regards to the origin of the wavelet transformation, we use all possible shift transformations. That is, the non-decimated wavelet transformation (NDWT) uses the information from each shift for the

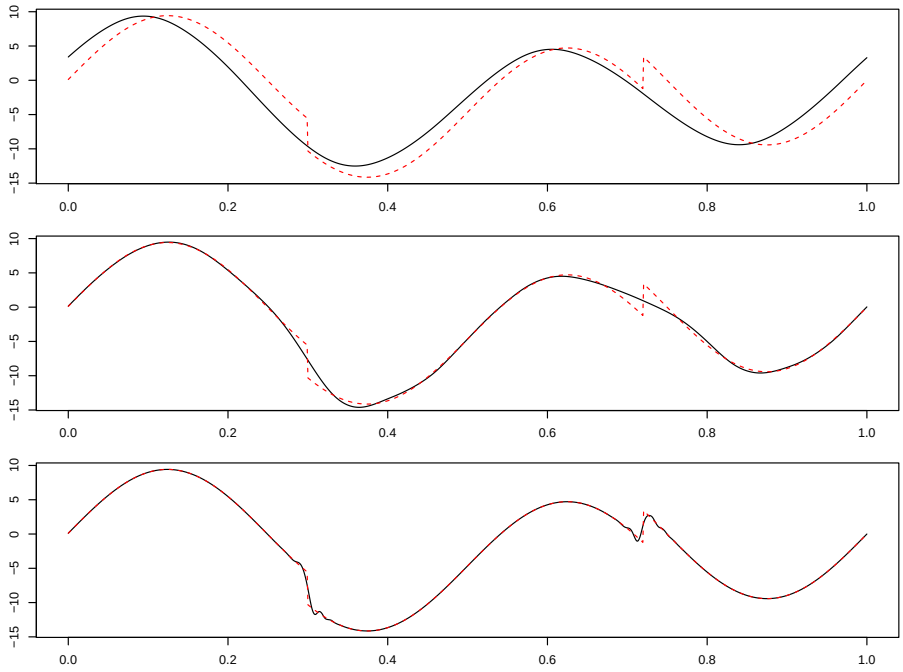


Figure 3.5: The true test function (red) and the different approximations at different levels using a Daubechies wavelet with nine vanishing moments. Top: $j = 1$, Middle: $j = 3$, Bottom: $j = 7$.

finest level, and then continues to do so at each subsequent coarser level. As stated earlier, the standard wavelet decomposition has $n/2$ coefficients for the finest level, and each wavelet level has half as many coefficients as the previous finest level. By using the non-decimated wavelet decomposition and, hence, taking into consideration all possible shifts, we have n coefficients for each level.

Using a similar scheme to the discrete decimated wavelet transformation, which is $\mathcal{O}(n)$, it can be seen that the NDWT is $\mathcal{O}(n \log_2 n)$ (Nason 2010). This makes the NDWT still a reasonably quick operation and so it can still be used when we are faced with large amounts of data. The disadvantage of using a NDWT is the fact that we no longer have an orthogonal transformation. Therefore, we are left with a decision to make when selecting what type of wavelet transformation we are going to use. The primary way of selecting this should be the purpose that we want to use it for, for example, if our objective is to use the wavelet as a form of data compression, then the normal decimated wavelet transformation would be advisable. If we would like to use the wavelet decomposition for uses in which we would like to utilise all of the information available, then the NDWT should be used. For example, it may be advantageous to use this decomposition when analysing time series data, as is done in locally stationary wavelet processes (Nason et al. 2000), or when considering wavelet shrinkage, which will be discussed in Section 3.8, due to the shift invariability in the DWT which may result in features not accurately represented.

3.8 Wavelet shrinkage

3.8.1 Classical wavelet shrinkage

Donoho (1993) introduced a wavelet-based method to estimate a function when we are able to make observations from the function with error. They began by imagining that we observe a function $g(x)$ with additive noise ϵ, such that we have the model

$$y(x) = g(x) + \epsilon(x).$$

As we can see, the randomness in this case is through the additive noise that we observe the data with. We are able to use the linearity of the wavelet transformation, using the matrix representation, to also write this model as

$$d^* = d + e,$$

where $d^* = Wy$, $d = Wg$, and $e = W\epsilon$ and W is the matrix representation of the wavelet transformation. We are able to use a matrix representation for the forward wavelet transformation as the calculation of the coefficients simply involves multiplication and addition of the data points.

At first glance, it appears that we have just given ourselves an equivalent problem and we have just moved into the frequency domain. We are, however, able to utilise the power of the wavelet transformation to provide ourselves a method to estimate g.

As we have stated earlier in Section 3.3, the wavelet transformation gives a sparse representation for most functions, including those that contain jump discontinuities and varying oscillations. We expect the resulting vector $\mathbf{d}$ to be a sparse one. We also know that if the noise term in the time domain ϵ is white noise, then the term e is also white noise due to the orthogonality of the wavelet transformation. To show this, we can look at the resulting distribution of the noisy coefficients d^*. Firstly, we start with assuming that we have i.i.d white noise, so we have

$$\mathbf{y} = \mathbf{g} + \epsilon \qquad \epsilon \sim N(\mathbf{0}, \sigma_\epsilon^2 I_n),$$

where I_n is the $n \times n$ identity matrix. We can see that $\mathbf{y}$ is normally distributed due to linearity. We perform the forward wavelet transformation to give us

$$\begin{aligned} W\mathbf{y} &= W(\mathbf{g} + \epsilon) \\ \mathbf{d}^* &= W\mathbf{g} + W\epsilon. \end{aligned}$$

As W is a linear transformation, we can also see that $\mathbf{d}^*$ is normally distributed. We can

now look at the expectation and variance of this distribution.

$$\begin{aligned}\mathrm{E}(\mathbf{d}^*) &= \mathrm{E}(W\mathbf{g} + W\epsilon) \\ &= W\mathrm{E}(\mathbf{g}) + W\mathrm{E}(\epsilon) \\ &= W\mathbf{g}.\end{aligned}$$

$$\begin{aligned}\mathrm{var}(\mathbf{d}^*) &= \mathrm{var}(W\mathbf{g} + W\epsilon) \\ &= W\mathrm{var}(\mathbf{g})W^T + W\mathrm{var}(\epsilon)W^T \\ &= 0 + W\sigma_\epsilon^2 W^T \\ &= \sigma_\epsilon^2 WW^T \\ &= \sigma_\epsilon^2.\end{aligned}$$

Hence, we can see that we still have the same error structure that we saw in the time domain, which we will see is advantageous when it comes to trying to recover the underlying function.

Many methods have been proposed for wavelet smoothing, and all of these involve utilising the sparseness of the wavelet transformation. The general idea is that the signal from g will result in large observed wavelet coefficients, and that coefficients that are close to zero will be attributed to just the noise that we have observed. As such, many authors have attempted to create threshold schemes in which they remove the small coefficients and retain the large ones so that we can form an estimate for $\mathbf{d}$. The original (and simplest) threshold schemes are the 'soft' and 'hard' threshold methods introduced by Donoho & Johnstone (1994). The soft and hard threshold functions are respectively defined as

$$f_s(\mathbf{d}^*, \lambda) = \mathrm{sgn}(\mathbf{d}^*)(|\mathbf{d}^*| - \lambda)\mathbb{1}_{[x<\lambda]}(\mathbf{d}^*), \tag{3.15}$$

$$f_h(\mathbf{d}^*, \lambda) = \mathbb{1}_{[x<\lambda]}(\mathbf{d}^*), \tag{3.16}$$

where $\mathbb{1}_{[A]}(\cdot)$ is the indicator function, outputting values of 1 when the input is in the space A and 0 otherwise, and λ is a threshold value. We can see that we have two different ways of looking at the shrinkage problem in these two methods. Hard thresholding, as we can see in equation (3.16), sets all values less than our threshold towards zero, whilst soft threshold, in equation (3.15), does the same whilst also shrinking the values that are above this threshold to zero. Hence, these methods were described using the phrases 'keep or kill' and 'shrink or kill' respectively. Intuitively, we can think that the soft threshold methods should result in a smoother estimate of the function g than the hard threshold.

Of course, there are many considerations when performing these methods. The most obvious of these considerations is the choice of the threshold value; in an ideal situation, we would like to select a value for λ such that it is larger than all values of the coefficients

attributed to noise, and smaller than all values of the wavelet coefficients of the underlying function g. This ideal situation however is extremely unlikely, we may, for example, have a situation in which the signal from the function and the noise of the samples are of a similar magnitude, making it very difficult to differentiate between the two. Secondly, we must also make a choice about the wavelet basis that we are using for our decomposition as this can often have a huge effect on our estimate. The choice is important not only when considering the number of vanishing moments that the wavelet has, and hence how smooth we believe the function is, but also we must consider the boundary conditions that are associated with the wavelet we choose.

One threshold method that was developed in Donoho & Johnstone (1994) is the universal threshold, and is defined by

$$\lambda^u = \sigma_\epsilon \sqrt{2 \log n},$$

where n is the number of observations. In applications, we rarely know the value of σ_ϵ and so it must be estimated by $\hat{\sigma}_\epsilon$. In their paper, they show that a bound can be put on the risk such that the performance of the shrinkage can be within a factor of $2 \log n$ of the ideal shrinkage. The two threshold methods and the universal threshold can be seen in the following example. We use the Blocks test function from Donoho & Johnstone (1994), in which we make $n = 1024$ equispaced observations from the function with a signal to noise ratio (SNR) of 7. We can see the results of this shrinkage in Figures 3.6 and 3.7. Firstly, we see the test function in Figure 3.6 and the noisy realisations of this. In the figure we can see the estimate of the function after we use hard and soft thresholding, in which we should note that the soft thresholding scheme produces a smoother estimate compared to the hard threshold scheme. We can also see the affect that the threshold scheme has had on the wavelet coefficients themselves in Figure 3.7. The top picture of Figure 3.7 shows the wavelet coefficients at the different resolution levels for the data we observe from the Bocks test function. In the middle and bottom pictures, we can see the wavelet coefficients after hard and soft thresholding respectively. More noticeably in the finer resolution levels, we can see that a very large proportion of the coefficients have been set to zero by the thresholding scheme. The difference between the two threshold methods is also easier to notice at the high resolution levels, with the large coefficients that are retained in both thresholds appearing to be of a smaller magnitude in soft thresholding compared to the hard variation.

3.8.2 Bayesian shrinkage

Another popular wavelet shrinkage method is the use of Bayesian methods for our shrinkage (Chipman et al. 1997). The idea is to use the sparsity that we expect from the decomposition and incorporate it into our prior distribution for the wavelet coefficients. Once

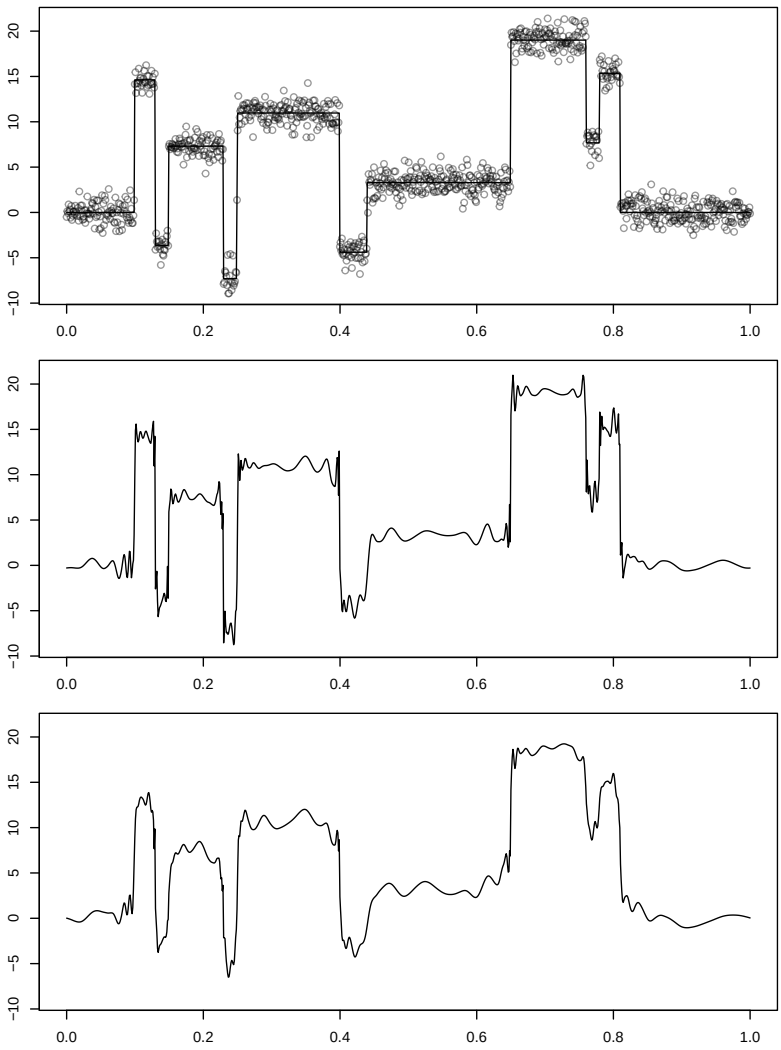


Figure 3.6: The blocks test function with SNR of 7 (top), the hard threshold (middle), and soft threshold (bottom) using the universal threshold.

we have built our prior distributions for the wavelet coefficients, we are able to use the observations from the function to find our posterior distribution for the wavelet coefficients. By performing an inverse wavelet transformation on the posterior distributions of the wavelet coefficients, we are, in theory, able to build a posterior distribution for the underlying function. This is extremely difficult to do analytically however (Barber et al. 2002), and most authors rely on sampling from the distribution, or using a summary statistic such as the posterior mean or median of the coefficients. Due to the inability to be able to calculate the posterior distribution of the function analytically in most cases, it is often difficult to report the uncertainty that we have in our estimate of the function f when using wavelets. Due to this difficulty, methods such as the Gaussian process (which was discussed in Chapter 2) are preferred when reporting uncertainty is a key aspect of our analysis.

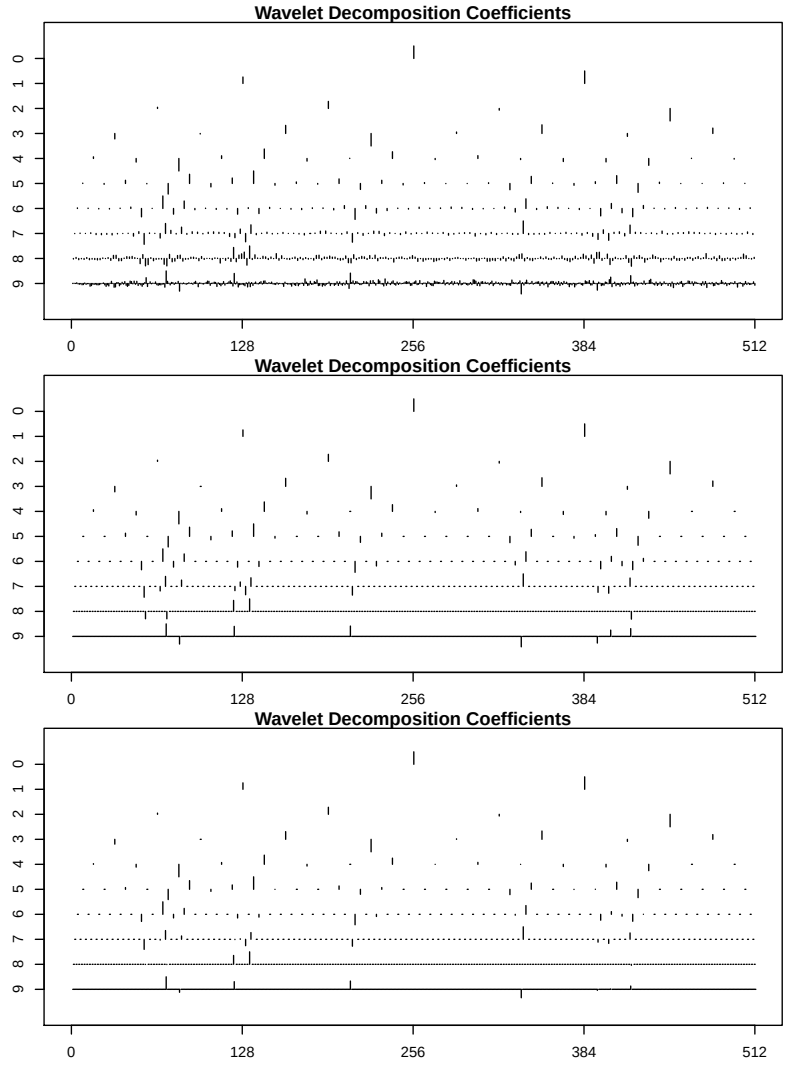


Figure 3.7: The wavelet decomposition of the nosy data (top), and the decomposition after the universal threshold is used with hard threshold (middle), and soft threshold (bottom).

The Gaussian mixture model

Early work by Vidakovic (1998) attempted to perform a fully Bayesian analysis using the prior knowledge that we expected the coefficients to be sparse. His first suggestion was to use a mixture prior for the prior distribution of these coefficients, in which we had one component of the mixture prior to model the large coefficients which we expect from the function g, and the other component modelling the small noise term. In his paper, two Gaussian distributions were used, such that the prior distribution for any wavelet coefficient d_{jk} was of the form

$$d_{j,k} = \gamma_{jk} N(0, \tau_j^2 c_j^2) + (1 - \gamma_{j,k}) N(0, \tau_j^2),$$

where $\gamma_{j,k}$ is a Bernoulli random variable, $\gamma_{j,k} \sim Ber(w_j)$, w_j is a hyperparameter that encodes our belief that the probability that the coefficient is non-zero, and $c_j^2 > 1$. Hence, we can see that we need to select the hyperparameters w_j, c_j^2, and τ_j^2. τ_j^2 is typically set to be small as this helps to represent our belief in those coefficients that we believe to

be zero. It was suggested in Chipman et al. (1997) that if we have a coefficient that is between $(-3\tau_j^2, 3\tau_j^2)$, it should effectively be zero.

This method was particularly desirable as the posterior distribution for the wavelet coefficient was easy to calculate and so inference was also very simple. We find that if we observe some noisy data and perform a wavelet decomposition on it, then we have noisy coefficients, which we can call $d^*_{j,k}$. A natural likelihood for the observed wavelet coefficients will be

$$d^*_{j,k}|d_{j,k} \sim N(d_{j,k}, \sigma^2).$$

Then, we are able to perform our posterior inference, firstly by looking at the marginal posterior distribution of the weights $\gamma_{j,k}$,

$$\begin{aligned}\text{Pr}(\gamma_{j,k} = 1|d^*_{j,k}) &= \frac{\pi(d^*_{j,k}|\gamma_{j,k} = 1)\text{Pr}(\gamma_{j,k} = 1)}{\sum_{i=0}^{1} \pi(d^*_{j,k}|\gamma_{j,k} = i)\text{Pr}(\gamma_{j,k} = i)} \\ &= \frac{\eta_{jk}}{\eta_{jk} + 1},\end{aligned}$$

where $\eta_{jk} = \frac{w_j \pi(d^*_{j,k}|\gamma_{j,k}=1)}{(1-w_j)\pi(d^*_{j,k}|\gamma_{j,k}=0)}$. We can hence also calculate that

$$\text{Pr}(\gamma_{j,k} = 0|d^*_{j,k}) = \frac{1}{\eta_{jk} + 1}$$

.

We are also able to calculate the other conditional posterior distribution of interest, which is the distribution of the underlying function after we have observed data, $d_{j,k}|d^*_{j,k}$. We can find this by first considering $d_{j,k}|d^*_{j,k}, \gamma_{j,k} = 1$ and $d_{j,k}|d^*_{j,k}, \gamma_{j,k} = 0$. We find that

$$\begin{aligned}d_{j,k}|d^*_{j,k}, \gamma_{j,k} = 1 &\sim N\left(\frac{d^*_{j,k}(c_j\tau_j)^2}{\sigma^2 + (c_j\tau_j)^2}, \frac{\sigma^2(c_j\tau_j)^2}{\sigma^2 + (c_j\tau_j)^2}\right), \\ d_{j,k}|d^*_{j,k}, \gamma_{j,k} = 0 &\sim N\left(\frac{d^*_{j,k}\tau_j^2}{\sigma^2 + \tau_j^2}, \frac{\sigma^2\tau_j^2}{\sigma^2 + \tau_j^2}\right).\end{aligned}$$

These are easy to calculate due to the fact that we are using a Gaussian prior and a Gaussian likelihood, which gives us a Gaussian posterior distribution due to conjugacy.

The two parts can be put together and we can therefore find our posterior distribution for the underlying function coefficients

$$\pi(d_{j,k}|d^*_{j,k}) = \text{Pr}(\gamma_{j,k} = 1|d^*_{j,k})\pi(d_{j,k}|d^*_{j,k}, \gamma_{j,k} = 1) + \text{Pr}(\gamma_{j,k} = 0|d^*_{j,k})\pi(d_{j,k}|d^*_{j,k}, \gamma_{j,k} = 0).$$

3.8.3 The point mass and symmetric distribution prior

The idea of the previous subsection is that we have a mixture prior, one Gaussian mixture component representing the large coefficients that encode the information of the underlying function, and another Gaussian mixture component, with very small variance,

representing those coefficients that we expect to be around zero. Johnstone & Silverman (2005) suggested replacing the second of these Gaussian components with a point mass placed at zero. If we believe that the wavelet decomposition is truly sparse, then we should believe that these coefficients are exactly zero rather than following a very narrow Gaussian distribution. They also put forward the suggestion of changing the first mixture component, which is a Gaussian distribution, to a distribution that is also symmetric, but with tails that decay to zero more slowly. They believed that giving the symmetric mixture component a larger density in the tails allows the coefficient to remain large in the posterior distribution. The Gaussian distribution, they discussed, had a tendency to shrink the large coefficients towards zero.

3.9 Conclusions

We have seen the theoretical framework of the wavelet methodology in this chapter. Its most popular use in statistics, through the discrete version of the wavelet transformation, the DWT, has been introduced. We have also highlighted the wavelets use in problems such as wavelet shrinkage, giving an introduction into this area. These two utilities, as well as the wavelets ability to represent a vectors main features through a small number of coefficients, will be used and expanded on in subsequent chapters.

In the next chapter, we introduce the first crossover between the wavelet methodology and Gaussian process emulation in the thesis. We begin with the situation in which we have a one-dimensional parameter (or input) for our unknown scalar function f, in which the function is believed to contain a discontinuity. If we are able to sample this function at further parameter values of our choice, one objective that may be of interest is to choose these points to be close to the location of the discontinuity, to help better define this challenging feature. A method is introduced that looks to exploit the characteristics of both the Gaussian process and wavelets to select these locations using the available information that we already posses of function.

Chapter 4

The one-dimensional wavelet sampler to find discontinuities

4.1 Introduction

We are sometimes faced with one-dimensional modeling problems, in which we are asked to predict a variable based on the value of a parameter. In other words, we are attempting to model the output of a function using the value of its input. In some real life applications, the underlying true function that we are attempting to model contains discontinuities (e.g. Alley et al. 2003). That is, the function that we are modelling has a sudden increase or decrease in the value of the output for certain parts of parameter space. To model the function to a sufficient standard, we therefore need to be able to model this challenging area of parameter space accurately.

Instrumental to the ability to model such a function accurately is the placement of the design points that we use to build our model (Chen et al. 2006). If we have a lack of design points around the location of the discontinuity, we may not have the required information about the features and placement of the discontinuity. Therefore, if the design of the initial experiment suggests that a discontinuity may be present, the design points may be inadequate to model the discontinuity. In this case, a method that suggests candidate parameter values that we should test to supplement our understanding of the discontinuity would be helpful.

The aim of this chapter is to develop a method, in situations where the function of interest contains a discontinuity, to select new input locations to sample. The method looks to select these new locations to be close to the area of the discontinuity so that we can model this area of space more accurately. In this chapter, we begin with a discussion of the problem and introduce methodology in Section 4.2, we state our algorithm for the method in Section 4.3, explore changes and considerations of the parameters of the algorithm in Section 4.4, and finally, we show the utility of the method on simulated

examples in Section 4.5.

4.2 The wavelet sampler

As we saw in Chapter 2, the Gaussian process can be used as a prior distribution for an unknown function. Using a Gaussian process to model a smooth function can provide us with a powerful tool for objectives such as prediction when the assumptions of the Gaussian process are valid. If we use the Gaussian process when the assumptions for the method are not valid however, the Gaussian process does not fit well. In one dimension, if the function we are attempting to model is non-smooth and contains discontinuities, the assumptions are invalid. If we observe the true function on both side of the discontinuity's location, the Gaussian process performs poorly due to the sudden change in output at this location, compared to the smoothness that occurs in the rest of the function.

When modelling this situation using a Gaussian process, the mean of the Gaussian process reacts to the non-smooth part of the function. It does this by oscillating around the location of the discontinuity. To keep the function's smoothness whilst also accounting for the sudden increase/decrease in the observed points, the function needs to go to extreme values to retain the smoothness and hence can provide poor prediction power around this location; this is known as Gibbs phenomenon. As stated in Section 4.1, one objective for our sampling may be to better define the location of this discontinuity, and so we can use this lack of fit to aid us in finding the location of the discontinuity and hence sample around this location. As we have seen in Chapter 3, wavelets can be a useful tool when measuring the oscillations of a function. Further to this, the wavelet methodology has an advantage over alternatives, such as spectral analysis, for our purpose in that we are also able to observe the location of the oscillation, as well as its size and frequency.

We can use an example scenario to visualise the methodology and the aspects of the method. Firstly, we set up a smooth test function which contains a discontinuity

$$f(x) = x^2 - 1.3x + \frac{1}{2}\mathbb{1}(x > 0.6). \tag{4.1}$$

The test function is then observed at seven different random training locations x between zero and one to build our dataset. The i.i.d. uniform distribution was used to decide on these locations. It can be seen that we have a very smooth function, one that is quadratic except for a jump discontinuity at $x = 0.6$. Using a Gaussian process on this function without the discontinuity present should provide a good model and accurate predictions. If we observe Figure 4.1 however, we can see that the discontinuity makes the Gaussian process inappropriate.

One argument is the question of why we are using a Gaussian process first to find the location of the discontinuity. Of course, as we saw in Chapter 3, if we are able to use a

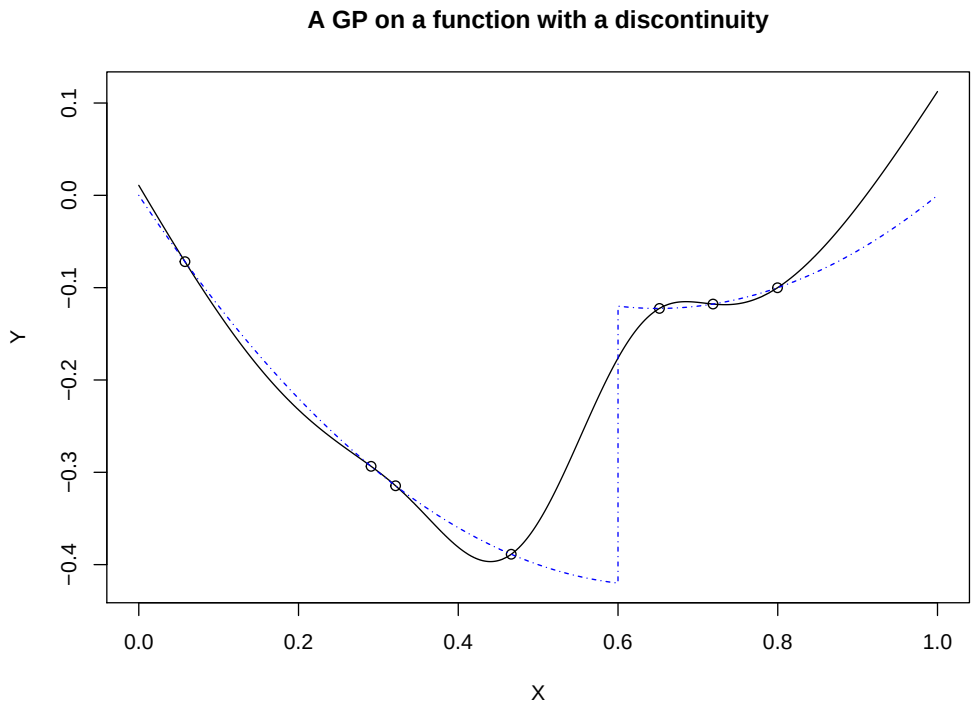


Figure 4.1: The example that we will use for the chapter. We can see the test function in equation (4.1), which is shown in blue. We observed the test function at seven random locations between zero and one which are shown by circles. A Gaussian process was fitted to the data and the mean of the posterior distribution in the space of interest is displayed as a black solid line.

wavelet decomposition on the dataset then we should do so and utilise that information accordingly. In many situations however, we do not have access to a dataset that is both equidistant and also dyadic. These are two very strong assumptions that we require, and we are often instead faced with situations in which the number of points that we are able to sample are both sparse and whose locations are chosen by other designs. Wavelets tend to perform poorly when the number of data points are very small.

4.2.1 Sampling an equally spaced dyadic dataset

If we are, in fact, faced with the situation in which we have data such that a wavelet decomposition is immediately applicable, then we should use wavelets as a direct tool. When we have the existence of a discontinuity, as we have already discussed, we will see a large sudden change in the output. In theory, if the data is of a setup that allows us to perform a DWT, the wavelet methodology should be able to pick up this feature at the finest scale levels (in which j is large), and situations like this will produce large (in the absolute sense) coefficients compared to the rest of the level (Nason 2010). To be more precise, if J is our finest resolution level of the DWT, if there is the existence of a

discontinuity, we will have a single or set of indices G such that

$$|d_{J,G}| > |d_{J,k}| \quad \forall k \in \{1, \ldots, K\}/G,$$

where K is the total number of coefficients in resolution level J.

Now, we know that we are looking for coefficients that are 'large' at the finest scale levels, indicating the presence of a discontinuity. We then need to decide on how to use this information to select candidate points for us to sample at, to better define the location of the discontinuity. If we are able to sample only one point, and the data allows us to perform a DWT, then the obvious choice of location is that which lines up with the largest absolute coefficient. At the finest resolution level, a wavelet coefficient is calculated using $2m$ data points, where m is the number of vanishing moments that the wavelet possesses. Hence, for those data points that correspond to the largest coefficient, a suitable location to sample a new data point is in the middle of these. That is, we sample between the mth and $m + 1$th data point of those used for calculating the largest absolute fine scale coefficient. Another sampling choice that could be used for this example is to sample uniformly over the support of the wavelet belonging to the coefficient of interest.

4.2.2 Sampling when we do not have an equally spaced dyadic dataset

In many real life cases, we are not able to observe the function at many input locations, such as in our example, seen in Figure 4.1, in which we have only observed the function at seven input values. In this situation, not only can we not use a wavelet decomposition directly, but we have also realised the function at a limited number of locations. As the locations are not equally spaced, nor are they dyadic in number, to perform a wavelet decomposition, we must perform some kind of pre-processing procedure to manipulate the data into the required form. The method that we choose for this pre-processing is the use of a Gaussian process. As we saw in Chapter 2, given data, we are able to find the posterior distribution of any unseen parameter values using the Gaussian process (Rasmussen 1996). We can hence use this property to predict values at locations such that we have a new dyadic and equispaced dataset.

For our data, we define the n observed variable values by

$$\mathbf{x} = \{x_1, \ldots, x_n\},$$

and their respective scalar response values as

$$\mathbf{y} = \{y_1, \ldots, y_n\}.$$

If we want to run a wavelet analysis that has a maximum resolution level of J, then we can define the number of points that we will use in our wavelet analysis as $n^* = 2^J$. We

can then define the parameter values as $\mathbf{x}^* = \{x_1^*, \ldots, x_{n^*}^*\}$, where the values of x^* are equally spaced across the parameter space, and we also define

$$\hat{y}(x^*) = E(y|x^*, \mathbf{x}, \mathbf{y}),$$

which is a function that takes a location x^*, and returns the posterior mean of the Gaussian process at that location. The predicted output values are then defined as

$$\mathbf{y}^* = \{\hat{y}(x_1^*), \ldots, \hat{y}(x_n^*)\}.$$

The pre-processing stage of our method is advantageous as it allows us to use the data that we have seen, and adapt it such that we can create a dataset that is in a form such that we can utilise wavelet analysis. Further to this, as the Gaussian process can give us a posterior distribution for any parameter value, we are able to select the finest resolution level that we will use, and, hence, make the wavelet decomposition as coarse or as fine as we require. There is, however, a consideration to this pre-processing that we have alluded to previously that must be addressed. By utilising the Gaussian process, we are using a smooth function to find those estimated points y^*, and hence these estimated points that we are going to use have a smooth structure imposed onto them. This imposed structure needs to have serious consideration as, in the situations that we have discussed and are interested in, the underlying true structure has some non-smooth feature that we are smoothing over.

As mentioned previously, the mean of the Gaussian process oscillates around the location of a discontinuity to deal with the sudden change in output. We therefore need to find the location in which the oscillation occurs so that we have an idea of where the discontinuity is in the parameter space, and can suggest candidate points to sample based on this information. This is obviously a different feature that we are looking for when we contrast it with the wavelet method that we discussed without pre-processing. Now, let us consider the simulated example from Figure 4.1 and observe the information that the wavelet decomposition provides us. For this, we have used a Haar wavelet with $J = 14$ to give us a very fine resolution level for our exploration, whilst still remaining computationally quick; this can be seen in Figure 4.2. A wavelet is used to find Gibbs phenomenon due to its ability to model smoother functions sparsely, and, hence, emphasise significant changes in the output that we would expect when we have this phenomenon. This ability to model smoother functions makes it advantageous over $f''(x)$ directly, with the wavelet scheme emphasising any jump discontinuities or sudden change in the function f through its coefficients more than $f''(x)$.

We can see in Figure 4.2 that the finest resolution level displays the oscillating feature that we have discussed. We see that the discontinuity is characterised by large wavelet coefficients that shrink towards zero, followed by a small number of negative coefficients,

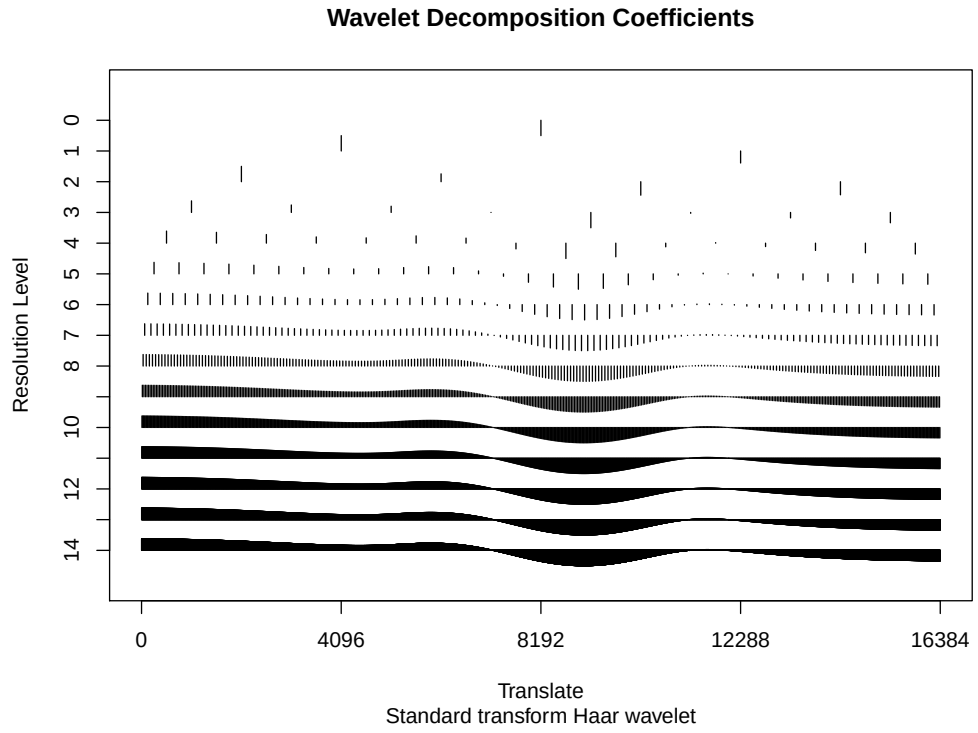


Figure 4.2: The wavelet decomposition of the Gaussian process from the seven realisations in equation (4.1) using $m = 1$.

and then increases back towards the large coefficients that we saw at the start of the oscillation. Our best guess of where the discontinuity occurs is therefore in the small number of coefficients that we observe between the large coefficients of the oscillation. Hence, a metric must be used to identify where these coefficients are and provide us with candidate points for our discontinuity sampler. As we have noted that there is also oscillation in the wavelet coefficients, we can use this to identify the region. One possible solution that is suggested is using a moving windowed variance. It is easy to see that if we were to measure the variance of something that is oscillating to extreme values, we would have a large variance. The moving windowed variance finds the local variance of a subset of the data points, which is useful for our objective.

For clarity, we must define the moving windowed variance. Heuristically, in one dimension, a windowed variance looks at a consecutive subset of the data, and measures the variance of this subset. The windowed variance uses a set number of points, and by removing the first data point of the subset and adding the neighbour of the last data point which is not in the subset, we are able to find a range of localised variances for the data set.

We can formalise this idea mathematically; firstly, we define n_w to be the size of window, such that $n_w \leq n^*$. The size of the window determines how localised our point measure will be, with a larger window using a larger subset of points, and a smaller

window using a smaller subset of points. We define the ith windowed variance for the set of wavelet coefficients $\mathbf{d}_J$ as

$$WV_i = \text{var}(d_{J,i}, \ldots, d_{J,i+n_w-1}) \qquad i = 1, \ldots, n^* - n_w + 1.$$

The full collection of these variances are called the moving windowed variance. Using the finest resolution level coefficients from Figure 4.2, we can visualise the moving windowed variance values from our example in Figure 4.3. Noticeably from this figure, we can see the two large peaks in variance, corresponding to the oscillation seen in the finest resolution level of the wavelet decomposition that occurs before and after the discontinuity. The wavelet variance has maxima at either side of the discontinuity due to the oscillation that is present, and, hence, our best guess is that the discontinuity has occurred between these peaks.

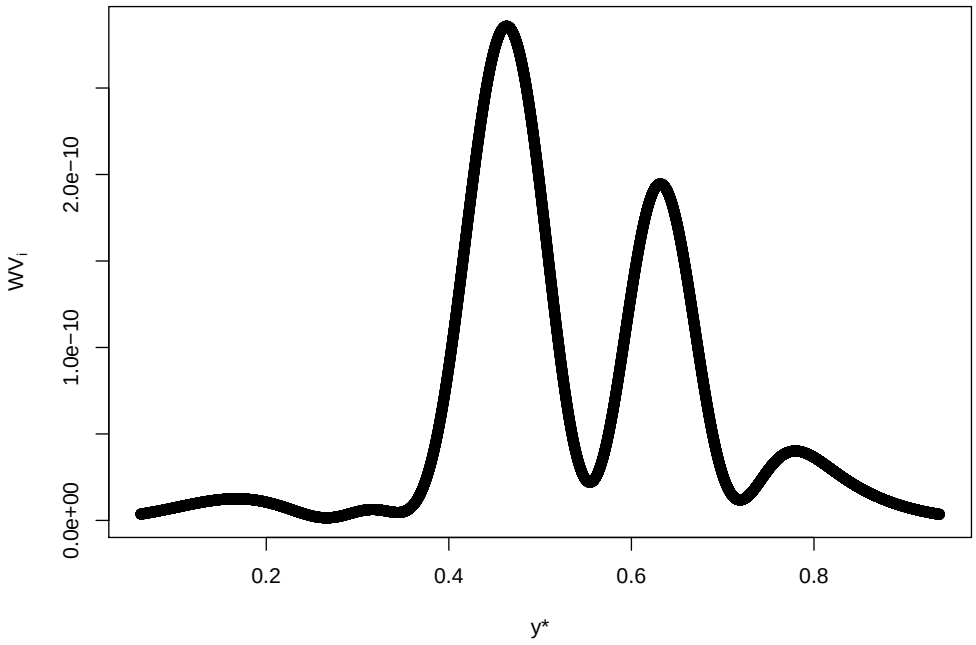


Figure 4.3: The windowed variance values for the finest resolution level of the wavelet decomposition in Figure 4.2. The coefficients of a wavelet decomposition with one vanishing moment at level $J = 14$ were used with a moving windowed variance size of $n_w = 2048$. On the x-axis, we have used the central value of y^* that was used for the WV calculation.

The feature that has been discussed can be used to propose the locations of a new sample. To propose a new sample that is near what we believe to be the location of the discontinuity, we find the input location that corresponds to the minima that we observe between the two large maxima. We can see from Figure 4.3 that there are multiple local

maximums in the windowed variance, and so if we are only able to only sample the function further at a single point, that points should correspond to the local minima that occurs between the two largest maxima. This point is chosen as we expect the largest variance to occur around the location of the discontinuity due to Gibbs phenomenon, and, hence, the minima that occurs between these maximas would be a sensible estimate for the exact location of the discontinuity. We can formulate the method in the form of an algorithm, which can be seen in Algorithm 1. Performing the algorithm on this example, with one vanishing moment (which is equivalent to the Haar wavelet) for the wavelet, $J = 14$, and $n_w = 2048$, gets the new sample location at $x^* = 0.553$, which can be seen in Figure 4.4. As this point is closer to the location of the discontinuity from below than any of the existing design points, we can see that the sampler has selected a suitable location. If the algorithm is ran a second time, with the additional design point, we find that the algorithm gives us a new sample location at $x^* = 0.599$, which is again closer to the location of the discontinuity than any of the existing design points from below, and can be seen in Figure 4.5. A third sample gives us a new sample location at $x = 0.624$, which is closer to the closer to the discontinuity than any existing design points from above, and can be seen in Figure 4.6.

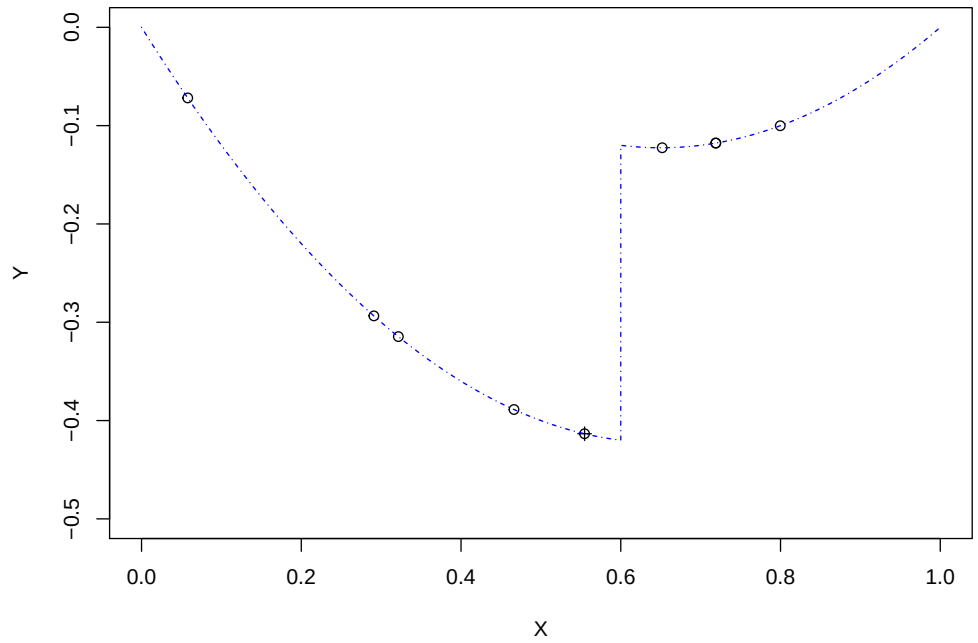


Figure 4.4: The first sample location that is produced from our wavelet sampler method. The existing design points are denoted by circles and the new location is shown by a circle with a cross.

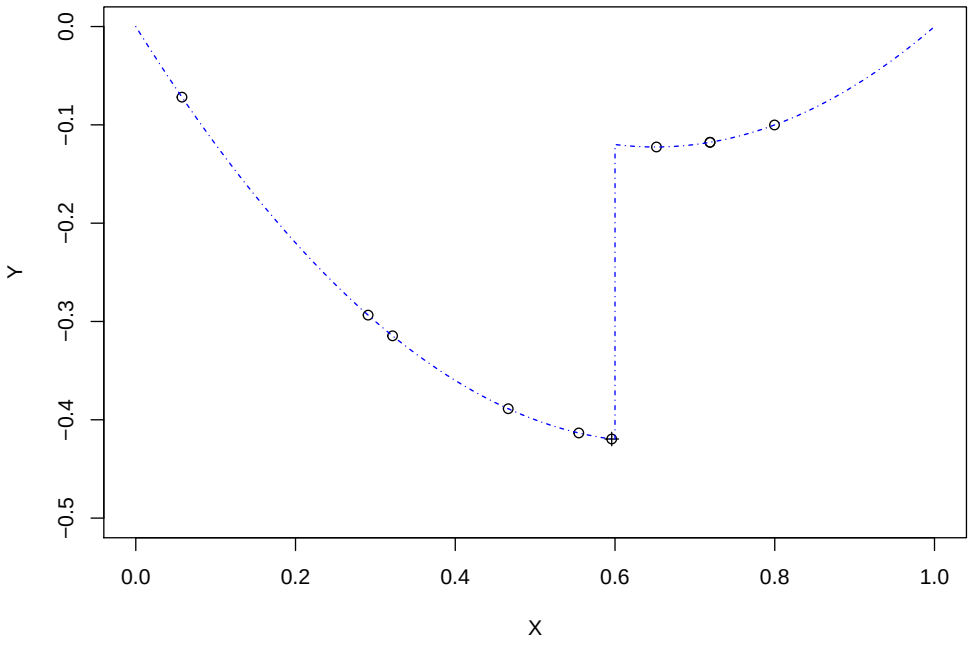


Figure 4.5: The second sample location that is produced from our wavelet sampler method. The existing design points are denoted by circles and the new location is shown by a circle with a cross.

4.3 The algorithm

Algorithm 1 The one-dimensional wavelet sampler

Require: $J > 0$ - Finest resolution level of wavelet decomposition;

Require: $n_w < 2^J$ - the size of the moving window variance;

Require: $m \in \{1, 2, 3\}$ - the number of vanishing moments;

If we do not have design points $\mathbf{x}$, select them using a design of your choice and evaluate $\mathbf{y}$;

Fit a Gaussian process using non-informative priors and a Gaussian covariance function to the underlying process and find its posterior distribution;

Sample mean of Gaussian process at locations x^*, such that x^* is equally spaced, and of length $n^* = 2^J$;

Perform wavelet decomposition on y^*, using a wavelet with m vanishing moments;

Find the moving windowed variance for the finest level coefficients $\mathbf{d}_J$;

Find the n_s largest local minimas for the windowed moving variance;

Find the data points that are used to calculate the n_s largest local minima

Our sample, x^*_{new}, are those locations that lie in the centre of those data points.

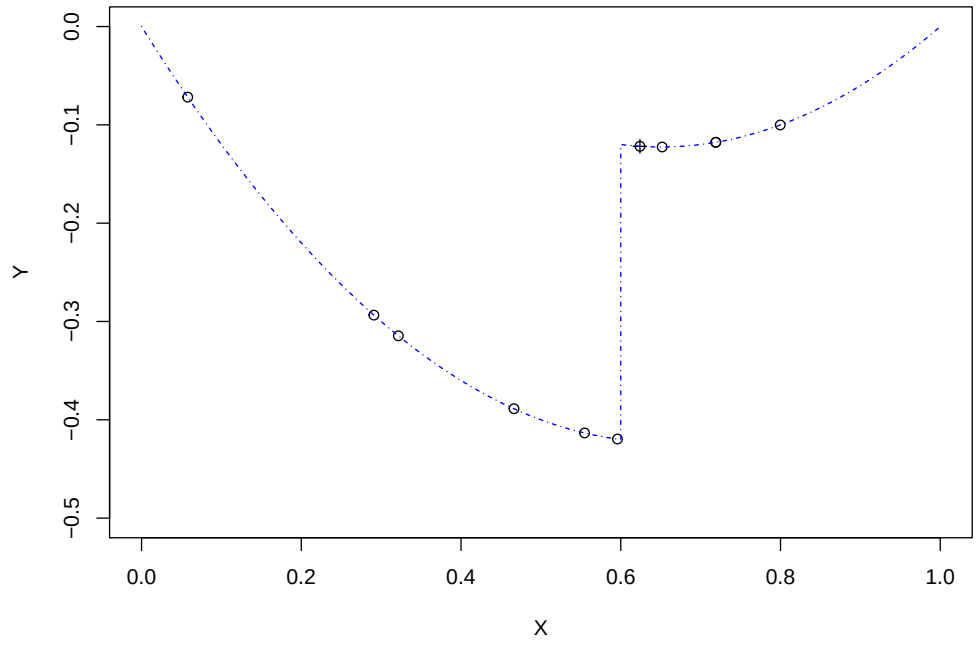


Figure 4.6: The third sample location that is produced from our wavelet sampler method. The existing design points are denoted by circles and the new location is shown by a circle with a cross.

4.4 Changing the parameters of the sampler

In this section, we discuss the changes that can be made to the parameters of Algorithm 1, and explore the effect that this has on our sampler.

4.4.1 The window size

Firstly, we can compare the effect that changing the size of the window, n_w, has on our sampling method. We start by considering the effect of using a larger window; one obvious difference is the amount of computation time that is required in our sampling method. To calculate the moving windowed variance, we must find $(n - n_w + 1)$ local variances. Hence, as the window size n_w becomes larger, we are having to calculate a smaller number of variances.

Changing the size of the window will also have further effects than just the computation time required; it will also change the location that we will choose for our point(s) to be sampled. Consider again the windowed variance, as we alluded to earlier, it is a metric that can be used to describe the variance of a subset of points around a location. The window size therefore describes the number of points that are used in each subset, and, hence, the locality of the variance we are measuring. We can see this point emphasised visually

when we consider the windowed variance for our example, we can see a large window (Figure 4.7), and a small window (Figure 4.8). In Figure 4.7, when we use a large windowed variance, we observe that we have smoothed over the features of the oscillation that we are looking for to perform our sampler. This suggests that we must take great care when selecting the size of the window. Selecting a large window size, although saving upon computation time, will result in a poor sample location being selected due to the lack of locality in the variances. We can see however in Figure 4.8, that selecting a small value for the size of the window, on first sight, appears to be more effective in identifying the oscillatory features that we are searching for than a window that is too large.

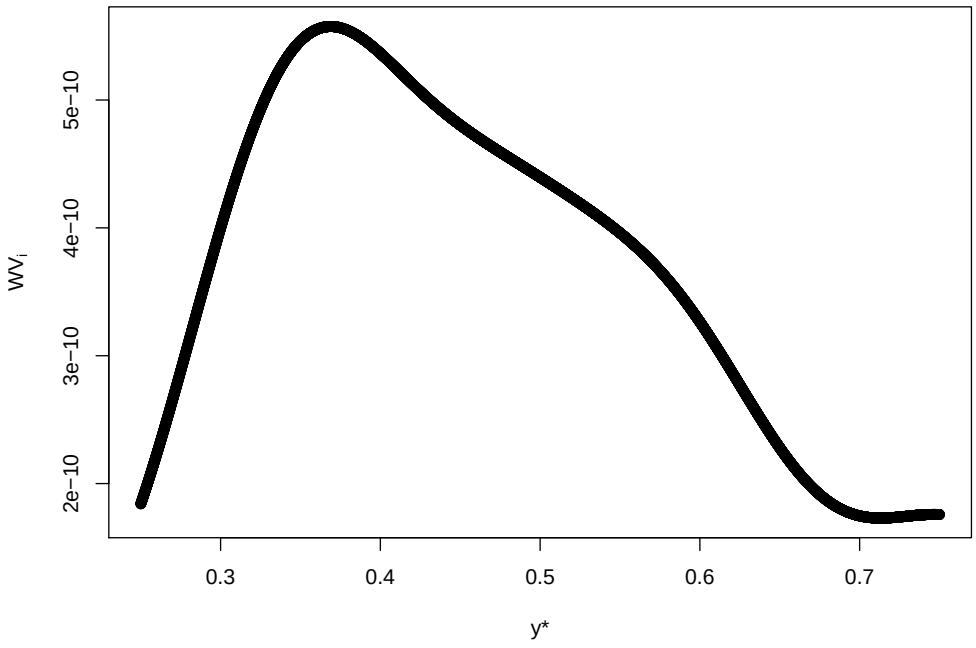


Figure 4.7: The windowed variance values for the finest resolution level of the wavelet decomposition in Figure 4.2. Here, we see a large window size of $n_w = 8192$. On the x-axis, we have used the central value of y^* that was used for the WV calculation.

We can observe the point, x^*, that we would sample for both window sizes using Algorithm 1 to highlight the problems of using a window that is too large or too small. When we use a large window, $n_w = 8192$, we sample at the point $x^* = 0.713$, which we can see in Figure 4.9. When we use a small variance window, $n_w = 64$, we sample at the point $x^* = 0.172$, which we can see in Figure 4.10. We can see that the algorithm has failed to perform correctly based on our criterion when both a window too large and a window too small is used. When the window is too large, the windowed variance has smoothed over the features of the oscillation, and the large local minima that we are searching for, and hence will not sample in the correct region. We can also see that when

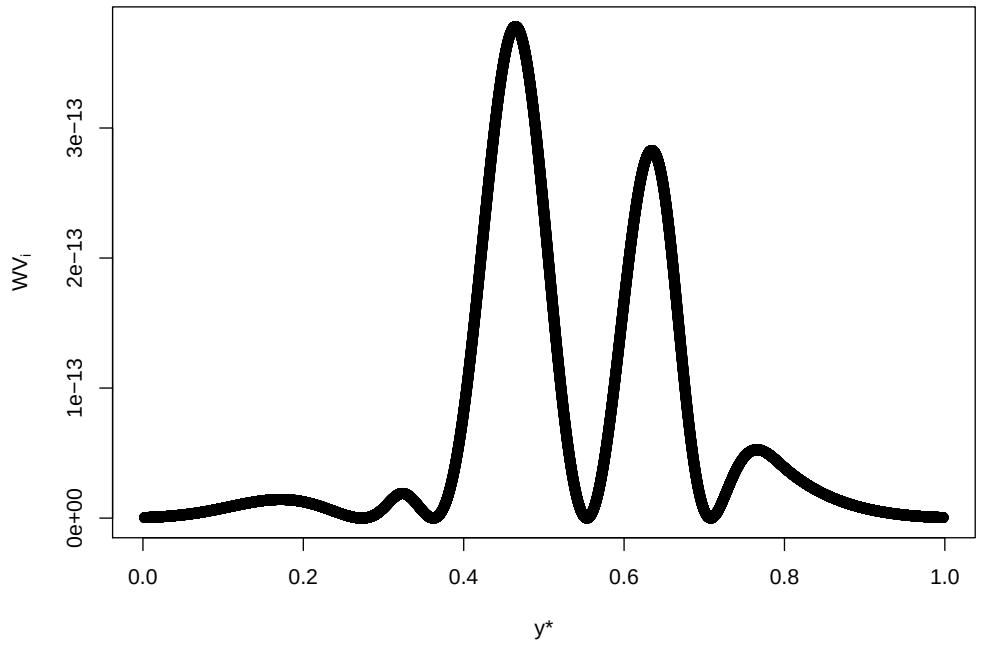


Figure 4.8: The windowed variance values for the finest resolution level of the wavelet decomposition in Figure 4.2. Here, we see a small window size of $n_w = 64$. On the x-axis, we have used the central value of y^* that was used for the WV calculation.

the window is too small, the two local maximas that occur due to the oscillation become disjoint, meaning that the local minima that occurs between the two is no longer large. To select the size of the moving window, n_w, in practice, n_w should be based on the resolution level that we have used in such a way that it is based on a ratio. In our examples, we found that the best results were found when the window size was $\frac{1}{4}$ the number of coefficients that are used, and so this could be used as a strategy to determine n_w.

4.4.2 The resolution level

Another parameter that can be changed is J, the finest resolution level coefficients of our wavelet analysis that we will use to identify the oscillation. From the properties of the Gaussian process, we know that we are able to select the value of J as we see fit. We can firstly observe our method when we reduce the value to $J = 10$. As the value of J has been reduced, we must also reduce the value of n_w, as we require $n_w \leq 2^J$. We reduce the size of the window such that the ratio between window size and J is equivalent to the original example, so that we now have $n_w = 128$. In Figure 4.11, we can see the windowed variance looks similar to Figure 4.3. We see the two maxima and the large local minima between the maxima that we are trying to identify for the sampler; when we perform our algorithm, we are told to sample at the point $x^* = 0.569$. This shows that the

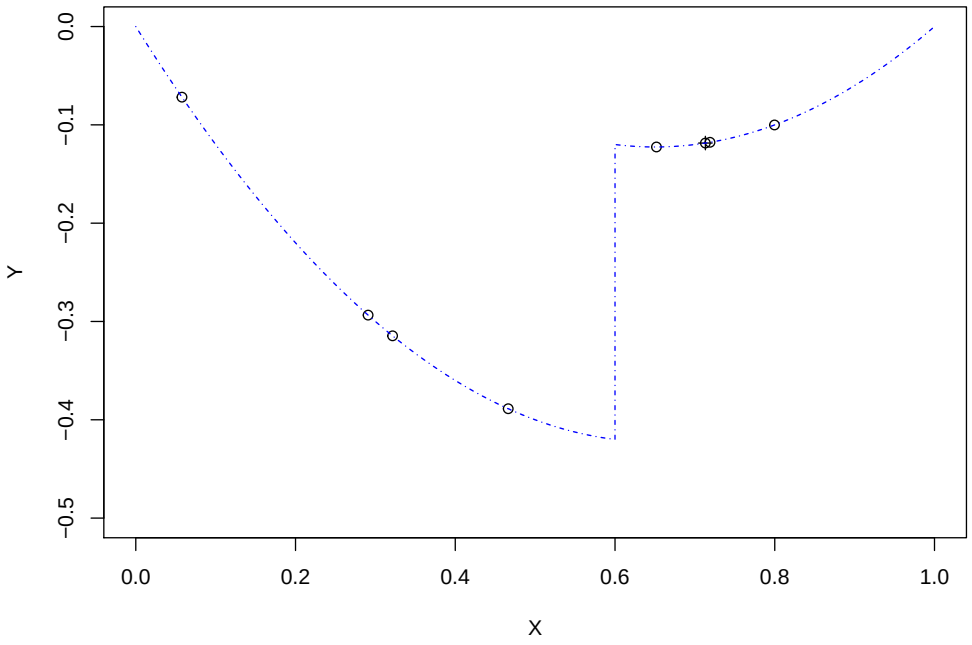


Figure 4.9: The original data points used for our example (circles), and the new data point (crossed circle) selected by our method when the window is large ($n_w = 8192$).

sampler can still perform well when reducing J, with the new point that we have chosen to sample at lying between the two design points that neighbour the discontinuity location. We are even able to reduce the value of J further whilst maintaining the characteristics of the Gaussian process and gain a good sample, with the windowed variance when $J = 5$ and $n_w = 4$ shown in Figure 4.13.

4.4.3 The number of vanishing moments

Finally, we can also take into consideration the number of vanishing moments that the wavelet has for our method. For all of the previous examples, we have used a wavelet with one vanishing moment – which is the Haar wavelet. We could instead use wavelets with more vanishing moments. One thing that we must account for when using wavelets with more than one vanishing moments is the boundary problem. If, as in our example, we have a function that is not periodic or symmetric, our decomposition will have problems with those coefficients next to the boundary.

We can see a visual example of this in Figure 4.14, in which we have performed a wavelet decomposition on our example. It can be seen that the first wavelet coefficient is affected by this problem, with an extremely large value present (relative to the rest of the level's coefficients) using $m = 2$ vanishing moments . We could, which is proposed as a solution to the boundary condition in Nason (2010), perform some pre-processing to

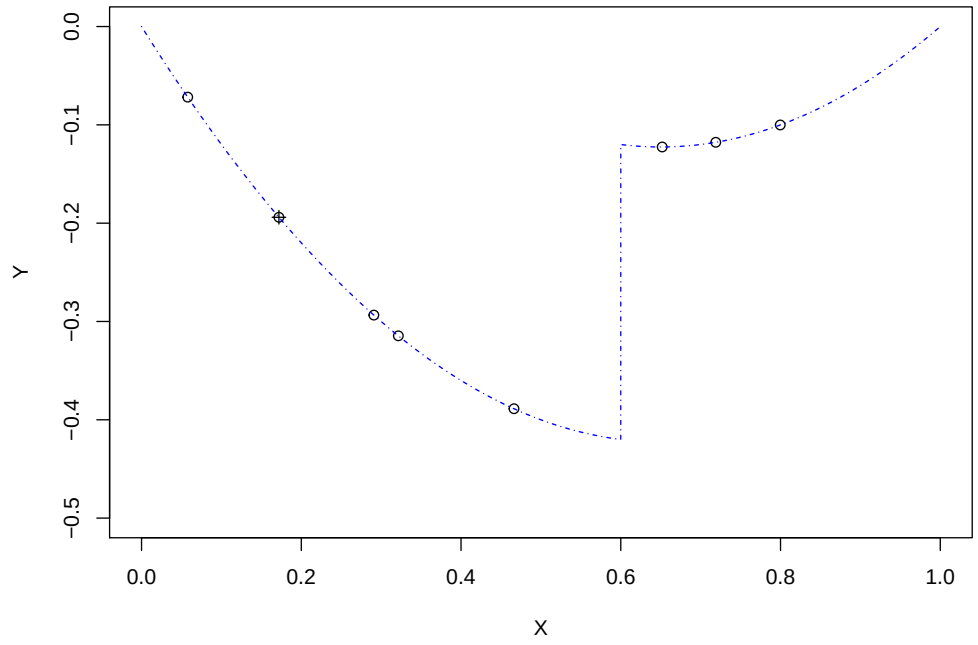


Figure 4.10: The original data points used for our example (circles), and the new data point (crossed circle) selected by our method when the window is small ($n_w = 64$).

the original data points before we perform the wavelet decomposition. Alternatively, we could, assume that the discontinuity does not occur at the very start or end of the input space, and hence ignore those coefficients that are affected by the boundary problem. The latter assumption is used in this example as we want to impose as little structure as possible on the data points, and ensure that the smoothness of the Gaussian process is retained when we perform a wavelet decomposition.

The same setup is used as the original example, in which we use $J = 14$ and $n_w = 2048$. We can observe the windowed variance when we use a wavelet with two vanishing moments in Figure 4.15. It can be seen in the moving windowed variance that we have one large global maximum, and two smaller local maxima next to it. These peaks correspond to the oscillation that we can see in the Gaussian process. The location where the first large local minima occurs corresponds to the region between the two design points that neighbour the discontinuity, $x = 0.466$ and $x = 0.652$, but is closer to the point 0.466, whilst the second large local minima corresponds to the same space but is closer to 0.652. As the second local minima between the three maximas is the largest, this location is where the algorithm decides to sample at, and gives us the value $x^* = 0.631$. Values of $m > 3$ were tested for Algorithm 1, but appeared to lose the smoothness that we see in the finest level resolution coefficients at $m \leq 3$. We therefore do not recommend that values of $m > 3$ are used in the algorithm.

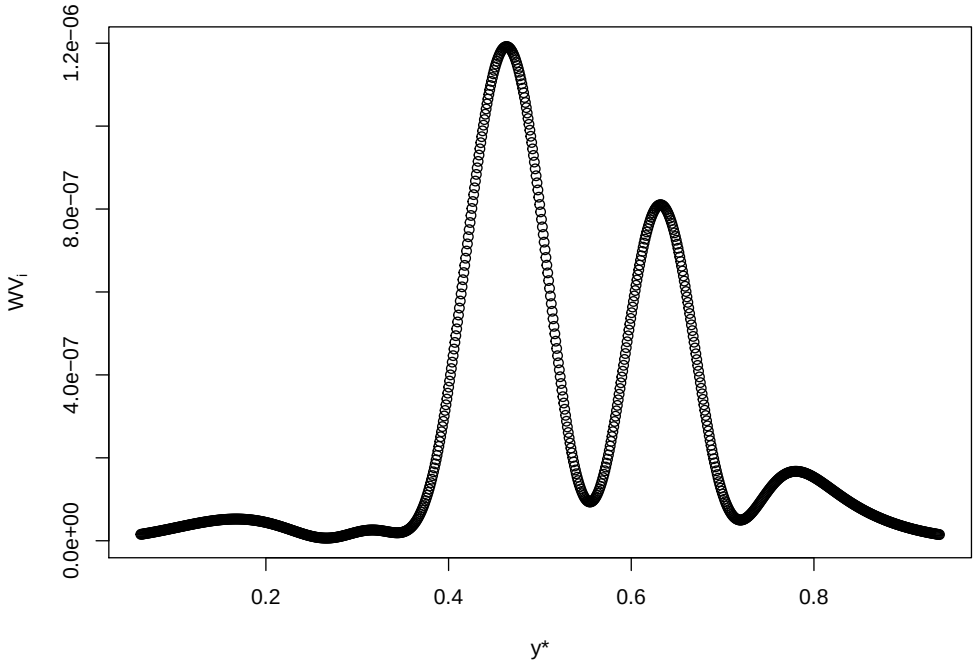


Figure 4.11: The Windowed variance of the wavelet coefficients when $J = 10$. On the x-axis, we have used the central value of y^* that was used for the WV calculation.

4.5 Simulated examples

In this section, we test the method that has been introduced in this chapter on a variety of test functions and situations. The test function used is of the form

$$f(x) = \begin{cases} \beta_0 + \beta_1 x + \beta_2 x^2 & 0 \leq x < c, \\ \beta_0 + \beta_1 x + \beta_2 x^2 + \alpha & c \leq x \leq 1. \end{cases} \tag{4.2}$$

Here we will have additional randomness through the parameters of f. To simulate different scenarios, all of parameters, as well as the number of datapoints n_x and their placement, will be drawn randomly from a distribution.

$$\begin{aligned} n_x &\sim \text{DU}[4, 14], \\ x_i &\sim U(0, 1) \qquad i \in \{1, \ldots, n_x\}, \\ \beta_0 &\sim U(-5, 5), \\ \beta_1 &\sim U(-10, 10), \\ \beta_2 &\sim U(-15, 15), \\ \alpha &\sim U(0.5, 5), \\ c &\sim U(0, 1), \end{aligned}$$

This will be a challenging test function as we will have a lot of potential problematic situations for all types of sampling. As was discussed in Chapter 2, a general rule of

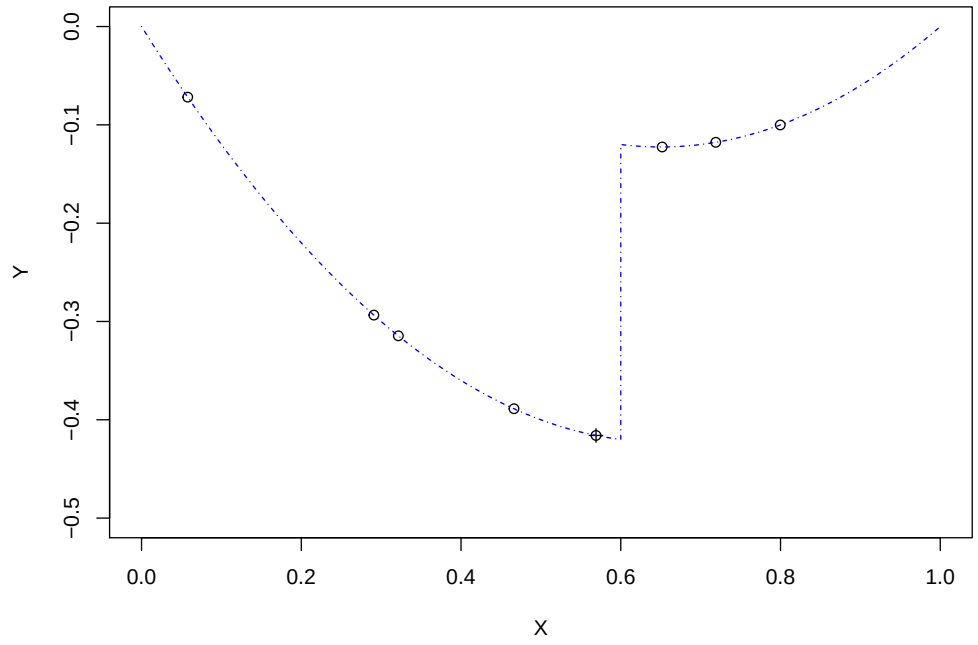


Figure 4.12: The original data points used for our example (circles), and the new data point (crossed circle) selected by our method when we have $n_w = 128$ and $J = 10$

thumb is that we should observe around ten samples in each dimension to ensure that we have a reasonable coverage. We can see from the distributions of the parameters that we will use for the example that we will often not reach this number of observations and, hence, may not have a good coverage of the space. We may also find situations in which the magnitude of α is small compared to the total activity in the function. This may mean that, unless the original design points have been well selected, with design points close to both sides of the discontinuity, the discontinuity can be easily missed. The location of the design points may also prove problematic, as we are using a uniform distribution to select the locations of the design points — we may not sample near the location of the discontinuity and hence be unaware of its existence.

For this test, we will use a wavelet with three vanishing moments for our method. We will assess the success of our sampling using a few aspects; Firstly, we can observe the consecutive number of samples that we make in which we are closer to the location of the discontinuity than any previously observed design points that immediately neighbour the discontinuity. That is, we began with n_x design points that were inputted into our test function from equation (4.2), and we will choose a new design point based on a sampling method. If that point is closer to the discontinuity than any previously existing design point, using the euclidean distance of the sampled point to c from before or after c, then we will add the new design point and continue to sample until this criteria is not satisfied.

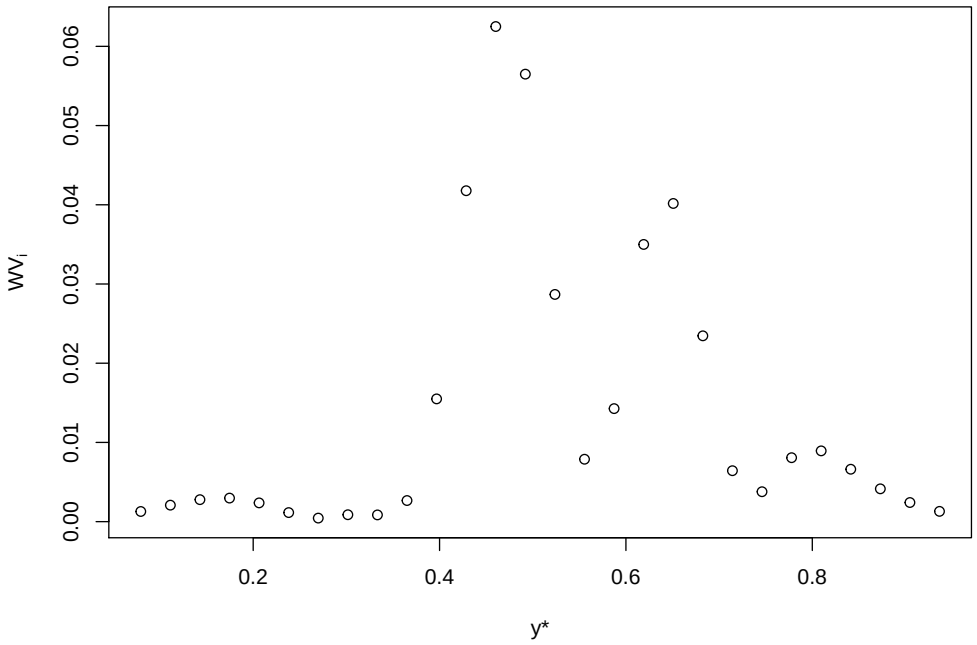


Figure 4.13: The windowed variance of the wavelet coefficients when $J = 5$ and $n_w = 4$. On the x-axis, we have used the central value of y^* that was used for the WV calculation.

Sampling method	0	1	2	3	4	5	> 5
Our method	0.512	0.072	0.120	0.144	0.121	0.015	0.008
Largest variance	0.760	0.136	0.064	0.032	0.000	0.000	0.000

Table 4.1: The proportion of times that the sampler was consecutively closer to the discontinuity than any existing design points. It can be seen that our method results in more consecutive samples that are closer to the location of c than the largest variance method for a larger proportion of our simulations.

Using this method, we can not only analyse the number of efficient design points that we have sampled, but we can also look at the location of these samples to see how efficient they have been. We can do this by looking at the distance of the closest design point to the discontinuity, and also looking at the distance between the two design points immediately neighbouring the discontinuity. We will also be comparing our method to another popular sampling method in the Gaussian process literature, which is choosing the new design point to be the location that has the largest posterior variance in the Gaussian process (Shewry & Wynn 1987).

The table shows that if we have a very limited number of points that we are able to sample, then we are more likely to select a point closer to the location of the discontinuity using our method as opposed to the largest variance method. In Table 4.1, we can see

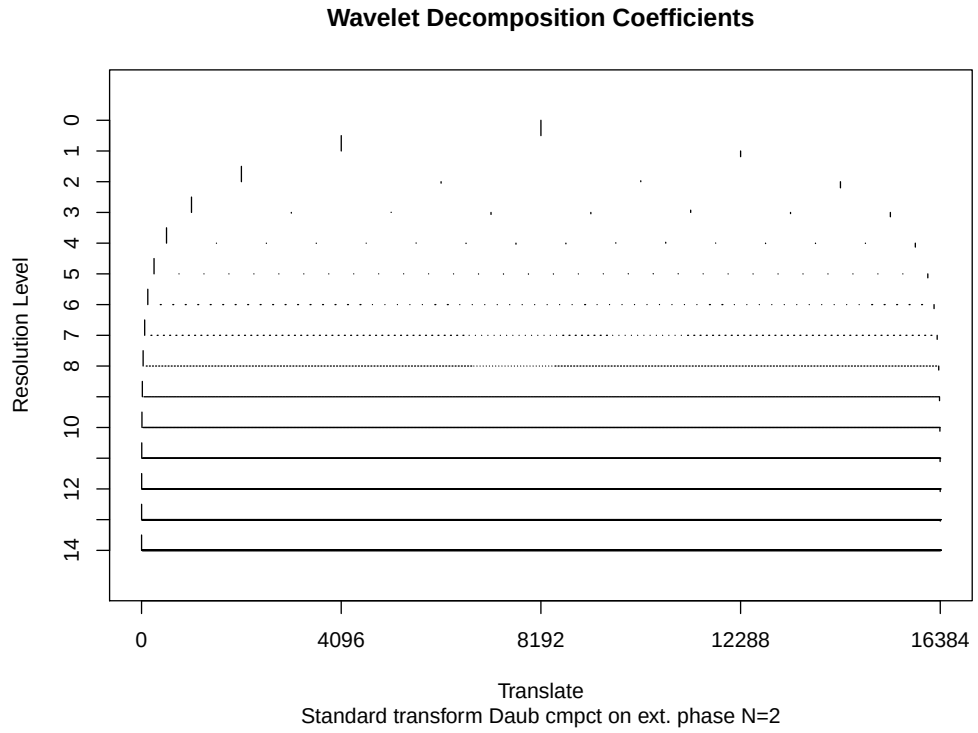


Figure 4.14: The wavelet decomposition of our Gaussian process when we use a wavelet with two vanishing moments.

that a reasonably large proportion of both sampling methods started their sampling with a location that is not closer to the discontinuity than any of the existing design points. This may seem like a surprising aspect initially, but further investigation uncovered that the difficult situation that the test functions can produce is responsible for this large proportion. Figure 4.16 shows three examples of test functions and samples in which we did not sample correctly in the first iteration for our sampling method. We can see that in these situations, there is often large areas of the input space that has not been sampled, in which the discontinuity lies, with only one (or zero) points on one side of the discontinuity. This large space with no data, as can be seen in the middle and bottom test functions in Figure 4.16, has the consequence that the Gaussian process is able to easily model the data without any kind of oscillatory features that we would expect when there is the existence of a discontinuity. The top test function in this figure is problematic due to the unfortunate way we have sampled, with the point to the right of the discontinuity having a similar output value to those to the left of the discontinuity. The bottom test function in Figure 4.16 would also prove problematic due to the small size of the jump discontinuity compared to the overall trend of the function. These results highlight the need to have a good initial set of design points as we could easily miss key features of the function.

We could also, rather than generalising the results for different sample sizes, condition on the number of samples that we have. We do so by looking at two cases, one in which

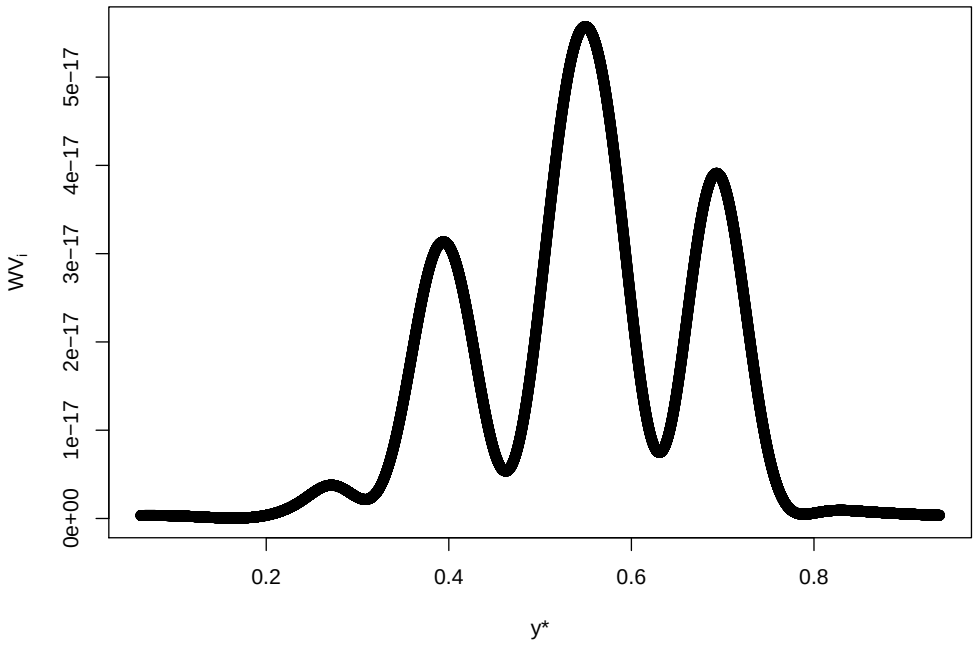


Figure 4.15: The Windowed variance of the wavelet coefficients in Figure 4.14, with $n_w = 2048$. On the x-axis, we have used the central value of y^* that was used for the WV calculation.

we have a small number of observations of the test function, with $n_x = 5$, and one in which we have a large number of observations of the test function, with $n_x = 13$. The results can be seen in Tables 4.2 and 4.3 respectively.

In Table 4.2, we can see that we have a reasonably large proportion of cases in which our initial sample is not closer to the discontinuity than the existing design points. As discussed earlier this is due to the number of cases in which we do not have enough information about the discontinuity from the initial design points. When this happens the method does not perform as well when we have a good number of design points. We can see that the largest uncertainty method actually ends up a smaller proportion of cases in which the first sample is not closer than the existing design points than our method.

Sampling method	0	1	2	3	4	5	> 5
Our method	0.646	0.223	0.012	0.051	0.051	0.012	0.000
Largest variance	0.608	0.304	0.063	0.025	0.000	0.000	0.000

Table 4.2: The proportion of times that the sampler was consecutively closer to the discontinuity than any existing design points for $n_x = 5$. It can be seen that our method results in more consecutive samples that are closer to the location of c than the largest variance method for a larger proportion of our simulations.

Sampling method	0	1	2	3	4	5	> 5
Our method	0.372	0.069	0.083	0.393	0.048	0.028	0.007
Largest variance	0.924	0.076	0.000	0.000	0.000	0.000	0.000

Table 4.3: The proportion of times that the sampler was consecutively closer to the discontinuity than any existing design points for $n_x = 13$. It can be seen that our method results in more consecutive samples that are closer to the location of c than the largest variance method for a larger proportion of our simulations.

Both of these values are reasonably close however, and we can see that if our first sample is indeed closer to the discontinuity than the existing design points, then our method is more likely to have more successive samples closer to the discontinuity than the largest variance method.

In Table 4.3, we are able to see the efficiency of our method over the largest uncertainty sampler. Typically we have a large amount of information from our initial design when the number of design points is large. We can see that our method is able to utilise this information to select sample locations that are closer to the discontinuity than existing points on most occasions. This is in direct contrast to the largest uncertainty sampler which does not sample close to the location of the discontinuity a large proportion of the time. This is often due to the initial design points lying reasonably close to the location of the discontinuity, with other parts of parameter space more sparsely sampled. As the uncertainty sampler is trying to sample the parts of space in which there are few data points, it often does not sample a point in a 'good' location.

In Figure 4.17, we can see that there has indeed been a general reduction in the distance to the location of the discontinuity after the sampling method for both our method and the largest posterior variance sampling. This is to be expected when we consider the information from Table 4.1. We can see, however, that our sampling method tends to lead to a greater reduction in the distance to the location of the discontinuity when compared to the largest uncertainty method. This point is especially prevalent when we consider Figure 4.18. We can see in this figure that the total distance to the location of the discontinuity, when we consider the two design points neighbouring the discontinuity, for our method has greatly reduced this distance compared to the initial design points. We can also notice that there is a noticeable improvement in sampling over the largest variance sampling method, indicating that our method is more efficient at sampling close to the discontinuity.

Further to this analysis, we could also simply sample three new design point locations using the method detailed in this chapter, and the largest uncertainty method. We then compare how close the new set of design points are to the location of the discontinuity.

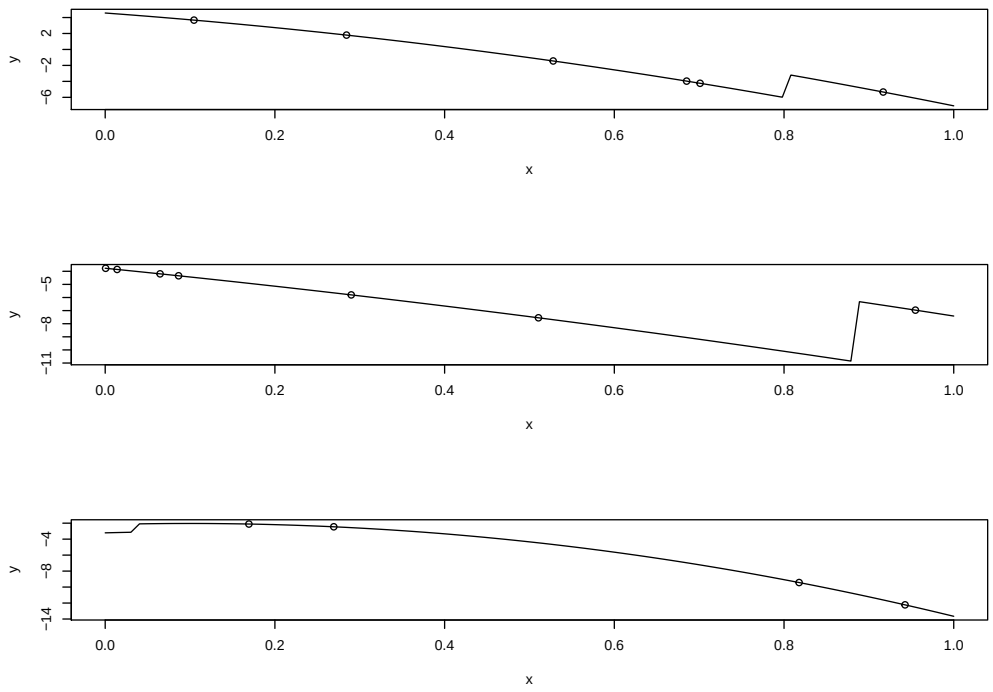


Figure 4.16: Three examples of test functions that were difficult to correctly sample initially. The solid black line is the true test function, the circles are the design points and respective outputs which we have observed the test function at.

We again looked at the two criterions that were observed in Figures 4.17 and 4.18 as our measure for how well the sampling method has performed. The results of this can be seen in Figures 4.16 and 4.20. It can be noticed in both figures that the sampler indeed clearly reduced the distance between the two points neighbouring the discontinuity and the discontinuity. Further to this, it can also be noticed that the method introduced in this chapter appears to sample closer to the location of the discontinuity than the largest uncertainty method. On average, after the three points sampled, the reduction in distance between the two points on either side of the discontinuity was 0.114 for our method, and 0.077 for the largest uncertainty sampler. For our sampling method, after three new design points were sampled, the average total absolute distance of the two neighbouring points to the discontinuity to the location of the discontinuity was 0.101. For the largest uncertainty method, this average distance was 0.138.

4.6 Conclusions

We have introduced a method to select locations of further design points using the information that we possess from the existing observations. The oscillatory effect that occurs in the posterior mean of a smooth Gaussian process when it is used as a prior distribution

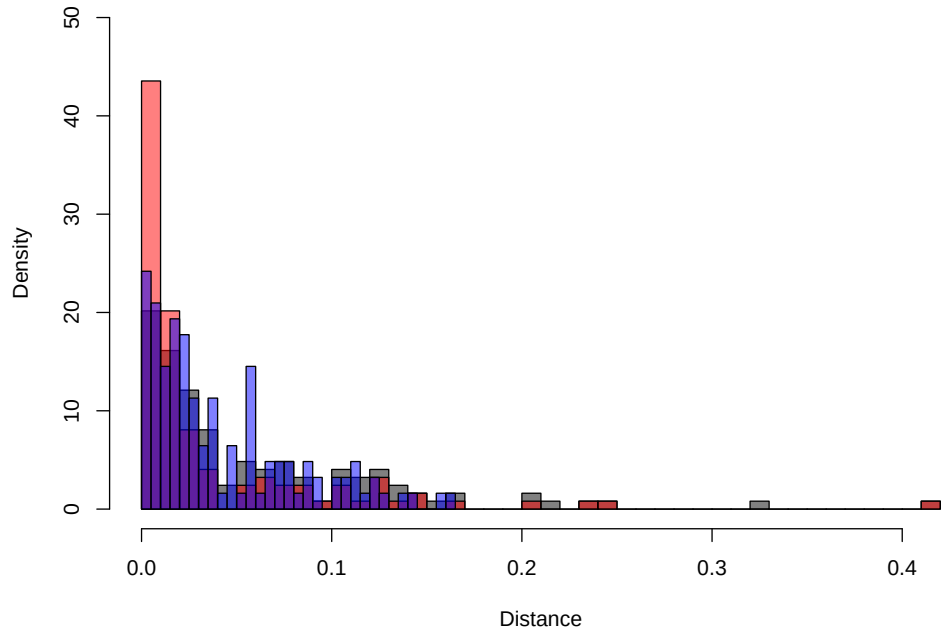


Figure 4.17: A histogram showing the distance of the closest design point to the location of the discontinuity: initially (grey), after we have sampled using our method (red), and using the alternative sampling of the largest posterior variance location (blue).

for a function that contains a discontinuity is utilised. The discrete wavelet decomposition is used to analyse the mean of the Gaussian process and identify the oscillatory effect, with that information subsequently used to decide the locations of our new additional design points. These locations are chosen as they are our best guess at the location of the discontinuity.

In the chapter, the effect that changing parameters of the method had on our sampling location was explored. We also explored the detrimental effect that a poor choice of parameter had and the reasons as to why this could cause us to select these poor locations. We have seen that for this one-dimensional setting, the sampling method appears to perform well. As was seen in Section 4.5, the ability of the sampling method is extremely hindered if we do not have an adequate set of original design points. As expected, and as would be the case for most methods, if we do not sample near the discontinuity in our original design points, or even have no samples in the region altogether, the method will fail to find the location of the discontinuity. Hence, the sampler could be seen to be advantageous over the largest uncertainty method when we have an adequate number of design points. If we suspect that this is not the case, or we have very large regions of the input space which has not been sampled, then the largest uncertainty sampler could be a good method to use to gain this adequate sample.

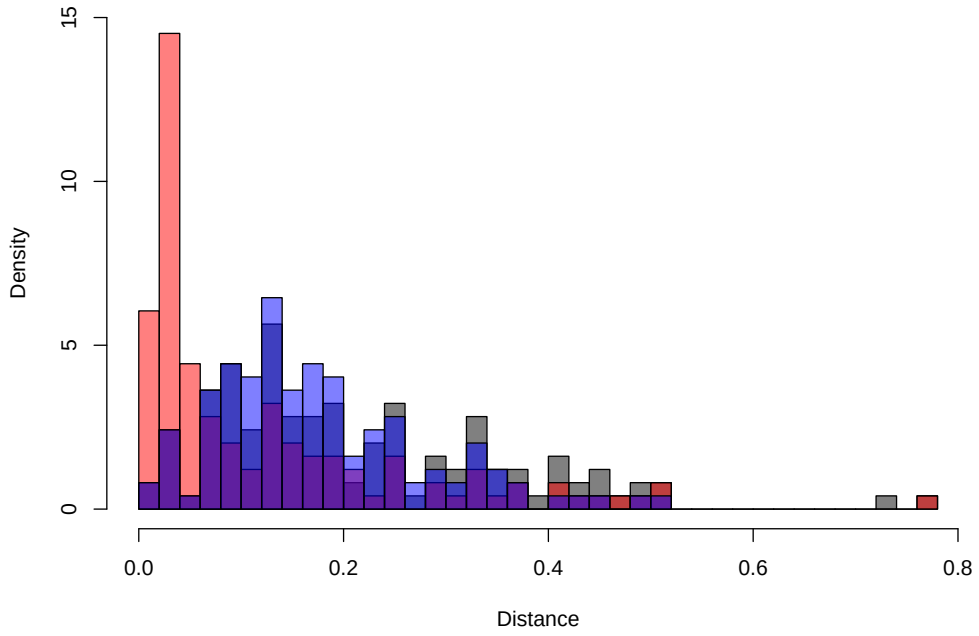


Figure 4.18: A histogram showing the summed absolute distance of the closest design points either side of the discontinuity to the location of the discontinuity: initially (grey), after we have sampled using our method (red), and using the alternative sampling of the largest posterior variance location (blue).

One obvious exploration for the future is how to extend the method into a multi-dimensional parameter problem. Using wavelets for data that is multidimensional is still possible; Nason (2010) describes the use of wavelet methods in two and three dimensions. The pitfall of this method's extension, however, is due to the need for a neighbourhood structure when attempting to identify the oscillating features of the Gaussian process. An alternative to wavelets, which was introduced in Sweldens (1998), is the method known as lifting. Sweldens showed that there were crossovers between wavelets and lifting, with a Haar wavelet decomposition shown to be a special case of lifting in one dimension. Although lifting is more useful than wavelets in theory to solve this problem, in that it can scale dimensions well, the neighbourhood structure again appears to be a problem when attempting to generalise the method from this chapter.

In the following chapter, the challenge of choosing the location of additional design points for a multidimensional parameter function which contains a discontinuity is explored. Not only is a novel method to select these locations introduced, but a full modelling scheme is created that allows us to model these types of functions more accurately than popular alternative methods.

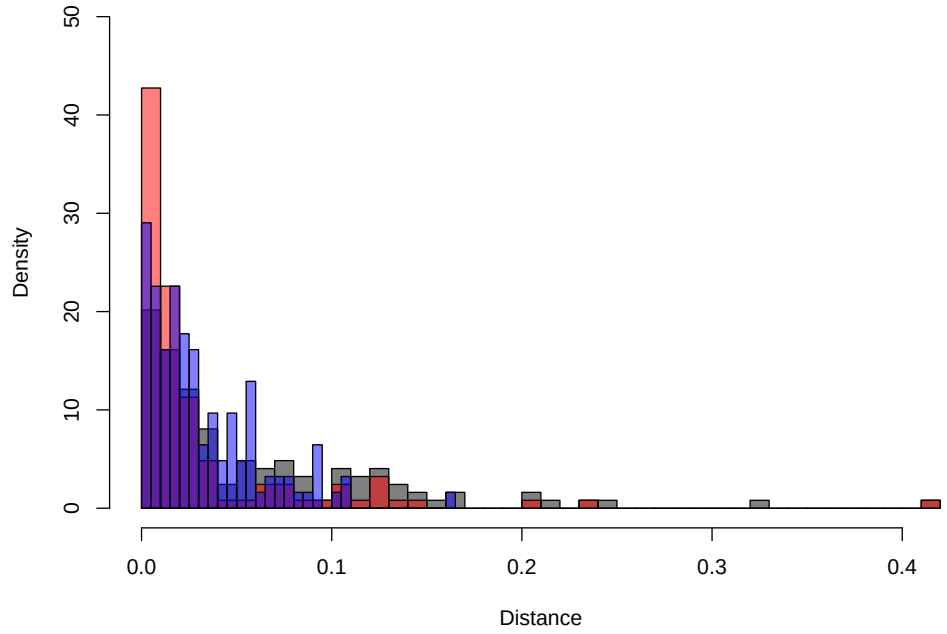


Figure 4.19: A histogram showing the distance of the closest design point to the location of the discontinuity initially (grey), after we have sampled three new design points using our method (red), and using the alternative sampling of the largest posterior variance location (blue).

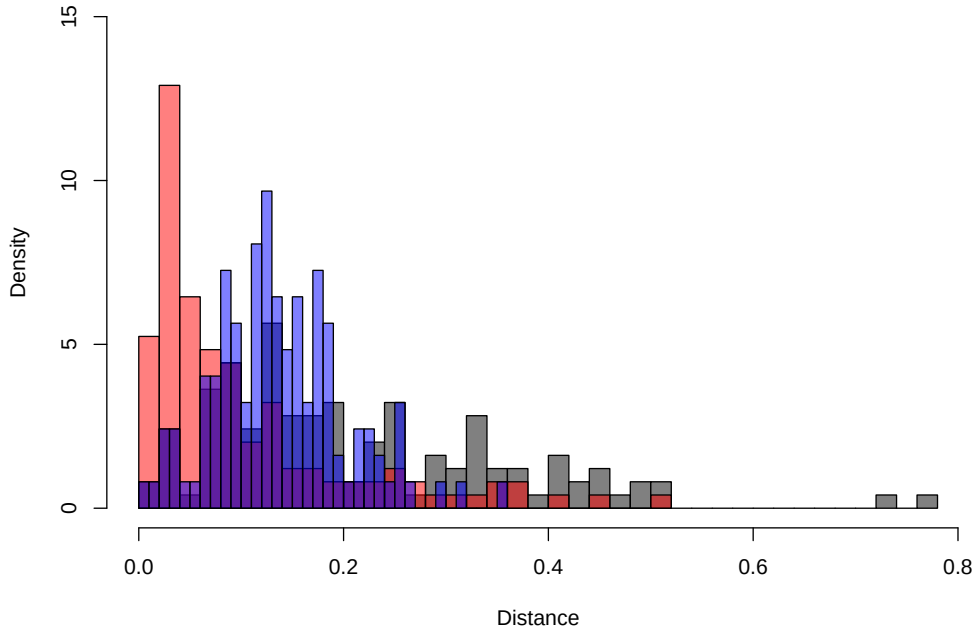


Figure 4.20: A histogram showing the summed distance of the closest design points either side of the discontinuity to the location of the discontinuity initially (grey), after we have sampled three new design points using our method (red), and using the alternative sampling of the largest posterior variance location (blue).

Chapter 5

Joint centre Voronoi tessellation Gaussian processes

5.1 Introduction

Often, when attempting to model a spatial process, smoothness is assumed. By smoothness, we are referring to the assumption that minor perturbations in the input locations lead to only minor changes in the output observations. Examples such as climate models are well known for small changes in the input resulting in large changes in the output for certain areas of input space (Alley et al. 2003). Naively modelling these processes using methods that rely on the assumption of smoothness can lead to poor results when the models are used for analysis such as prediction (Paciorek & Schervish 2006). Simple examples will be shown in Section 5.2 to explore and emphasise the problems that discontinuities pose. In this chapter we have two distinct aims: Firstly, to estimate a function f given a set of observation locations, where $f(\cdot)$ is potentially piecewise discontinuous, described in Section 5.4. Secondly, to select the locations to observe new design points with the objective of improving estimation of the discontinuity boundary, which is explored in Section 5.5.

Non-stationary methods have been used to build approximations for the underlying functions of processes that contain discontinuities and heterogeneity, such as mixtures of thin plate splines (Wood et al. 2002), which build a function $f(\cdot)$ that minimises the integral of the square of the second derivative; local linear regression (Cleveland et al. 1992), in which linear regression is used for prediction using a subset of the total number of model runs, with the subset chosen based on their distance to the prediction point; and wavelet-based imputation (Heaton & Silverman 2008), in which we estimate unseen locations using a Bayesian wavelet method that incorporates those locations into the wavelet decomposition and estimates those coefficients that are effected by the unobserved points via a Gibbs sampler. By making so few assumptions about the data, methods such as

these have the drawback that a large number of observations are often needed to build an accurate model of the underlying process. The method developed in this chapter is applicable to any situation in which we suspect that a process displays heterogeneity or that the function contains discontinuities.

One well-established method for spatial modelling is Gaussian process regression or kriging (Cressie 1993), which was already discussed in Chapter 2. By using a Gaussian process to model the underlying function (or random field), we are making an assumption of smoothness in the underlying function over the entire input space. As mentioned previously, this assumption is rarely justified. To deal with this, adaptations to the stationary Gaussian process methodology must be made to accommodate non-stationarity. Two of the main methods that have been focused on in the literature are changes to the covariance function, such as spatial deformations (Schmidt & O'Hagan 2003), which involves transforming the input space such that the output is more smooth over the newly transformed input space, with this functional space being used for our analysis instead of the original space, or convolution based methods (Risser & Calder 2015*a*), and the use of piecewise Gaussian processes, such as treed Gaussian process (TGP) (Gramacy & Lee 2008) or Voronoi-tessellation Gaussian process (Kim et al. 2005). Our method focuses on the latter of the two categories, and readers interested in adaptations to the covariance function are directed to Risser (2016) for a review. In this chapter, we introduce the relevant prerequisite for the method, a way in which the input space can be partitioned, and heuristic reasoning behind it in Section 5.3. The method is introduced and developed in Section 5.4; as the method is Bayesian we discuss how we will sample from the posterior distribution using a reversible-jump MCMC in Section 5.4.2. We show a useful supplementary sampling method for our model for situations in which we want to better define the boundaries of regions in Section 5.5. The method is test on a two-dimensional simulated examples in Section 5.6. The method is then tested on two real datasets in Section 5.7.

5.2 The use of a Gaussian process for a non-smooth function

We began our illustration of the use of a Gaussian process on a non-smooth with the test function

$$f(x) = 0.75\sin(2\pi x) + 0.75\cos(3\pi x) + x + \mathbb{1}_{x>0.65}(c) \quad x \in [0,1], \tag{5.1}$$

where

$$\mathbb{1}_A(c) = \begin{cases} c & \text{if } A \text{ is true,} \\ 0 & \text{else,} \end{cases}$$

and is referred to as an indicator function. If $c = 0$, the function would be smooth and using a Gaussian process prior seems logical, with the posterior distribution shown in Figure 5.1. It is easy to see that by increasing the constant c, the size of the jump discontinuity also increases, and we explore whether the size of this constant affects the goodness of fit for our (potentially inappropriate) Gaussian process.

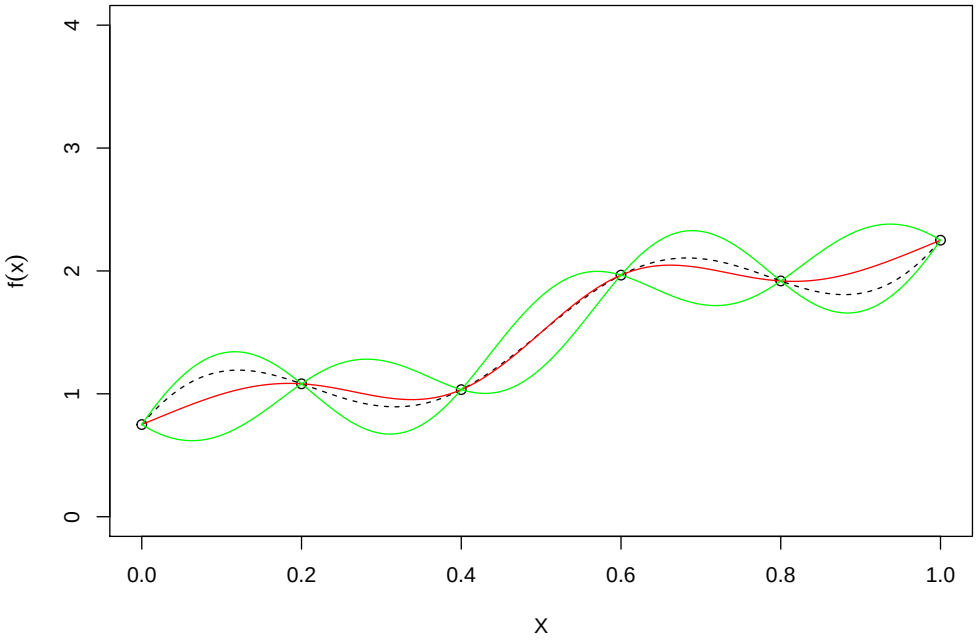


Figure 5.1: The posterior predictive distribution of 1,000 unobserved x locations between zero and one for the function in equation (5.1) with $c = 0$, observed at six locations. The true function is represented as a broken black line, the posterior mean function is the solid red line, and the central 95 % credible interval can be seen in green.

With regards to the Gaussian process, we used a squared exponential for the covariance function, seen in equation (2.3), the non-informative prior was used for (β, σ^2), σ^2_ϵ was set to zero (see equation (2.14)), and the length scale parameter b was chosen using maximum likelihood. To begin with, we start by observing equation (5.1) at six locations, which are $x = (0, 0.2, 0.4, 0.6, 0.8, 1)$, and we look at the effect on the posterior distribution that the value c has. The resulting posterior predictive distribution can be seen in Figure 5.2 for three different values of c in our test function: $c = 0.5, c = 1$, and $c = 1.75$. We show in this figure the true value of the function, the mean of the posterior distribution and the 95% credible interval of the function.

To analyse the fit of the examples seen in Figure 5.2, we look at the individual prediction errors, as suggested in Bastos & O'Hagan (2009). The individual prediction error

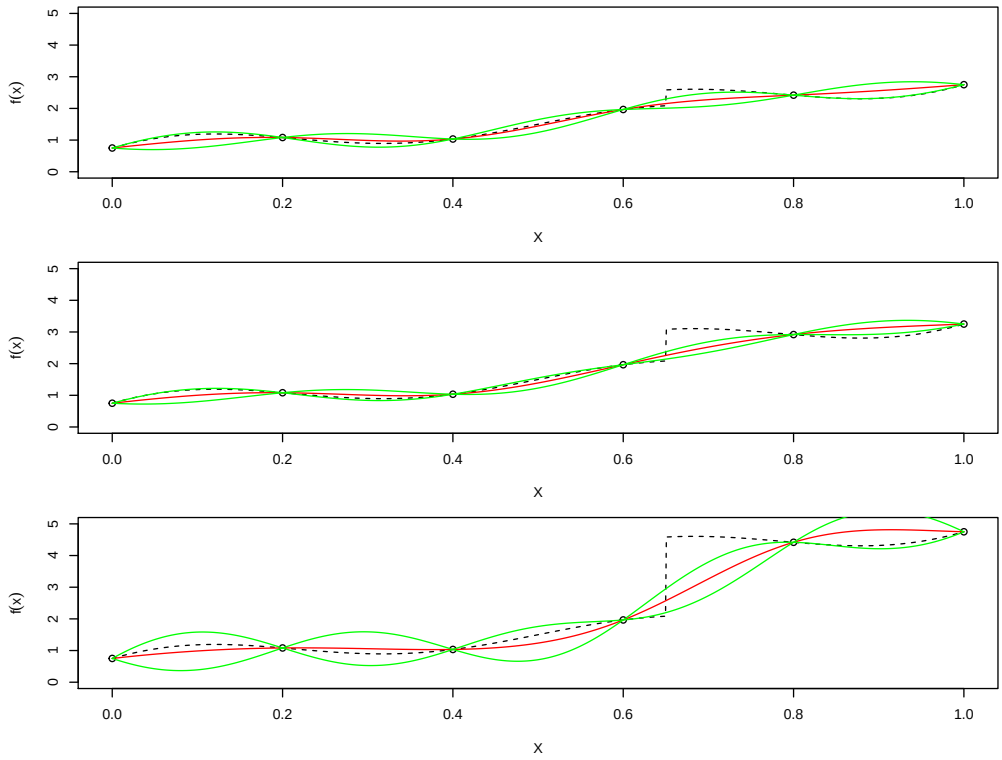


Figure 5.2: The posterior predictive distributions of the function from equation 5.1 with differing values of c. Top: $c = 0.5$, middle $c = 1$, bottom: $c = 1.75$. The true function is represented as a broken black line, the posterior mean function is the solid red line, and the central 95 % credible interval can be seen in green.

has the form

$$D_i(y^*) = \frac{y^* - E(f(x_i)|\mathbf{y})}{\sqrt{V(f(x_i)|\mathbf{y})}}, \tag{5.2}$$

where $E(f(x_i)|\mathbf{y})$ is the point-wise posterior expectation of the function f at x_i, y^* is the true output value at x_i, and $V(f(x_i)|\mathbf{y})$ is point-wise posterior variance of the function f at x_i. This measure is used as opposed to simply the mean squared error as it also takes into account the uncertainty in the posterior distribution, standardising the error values using this. The metric not only informs us of the prediction power of the Gaussian process, but it also indicates whether the uncertainty has been correctly encapsulated. We can see the numerator measures the difference between the true data-point outputted by the function and the posterior expectation of the function using the data $\mathbf{X}$ and $\mathbf{y}$. This difference is then standardised by the point-wise posterior variance of the Gaussian process. Hence, if we have a large posterior variance, there is a large amount of uncertainty in our prediction and will reduce the value of the difference to reflect the models ability to reflect this uncertainty.

We could compare the fits of the Gaussian processes to each other by looking at the average of the absolute values of these prediction errors at 1,000 equispaced points between 0 and 1. We can also see the individual prediction errors in the corresponding values of c

c	AAIPE
0	0.67
0.5	1.49
1	2.65
1.75	2.52

Table 5.1: The averaged absolute individual prediction error to two decimal places for the varying values of c in our test function from equation (5.1).

used for Figure 5.2 in Figure 5.3. We can see the averaged absolute individual prediction errors (AAIPE) over $x \in [0, 1]$ to two decimal places in Table 5.1, in which

$$\text{AAIPE} = \frac{1}{n_{D_i}} \sum_i |D_i(y^*)|,$$

where n_{D_i} is the number of locations we are testing the function at (i.e $n_{D_i} = 1000$ in our comparison). In this table, we can see that, as the value of the jump discontinuity increases, the averaged absolute individual prediction errors generally increases, showing that the Gaussian process is generally a poor fit for functions that contain jump discontinuities. As a rule of thumb, Bastos & O'Hagan (2009) stated that an absolute value greater than two indicates a poor fit from the Gaussian process for the function. We can see that as the average value of the absolute individual error is greater than two for $c \geq 1$, the Gaussian process is a poor fit for these non-smooth test functions.

Another way that we can explore the effect that a discontinuity has on our posterior inference is by changing how close the points that we observe the function are to the discontinuity. To do this, we change the locations of the points x_4 and x_5, which are the two closest points to the discontinuity (which is at $x = 0.65$). We observe three different situations, in which we have reduced the distance of both points to the discontinuity compared to the previous situation. In the first situation we have $(x_4, x_5) = (0.625, 0.725)$, those points are then moved such that $(x_4, x_5) = (0.633, 0.7)$, and finally we move the points to $(x_4, x_5) = (0.64, 0.68)$. We again optimise the parameter b using maximum likelihood. The posterior predictive distributions of these situations can be seen in Figure 5.4.

One immediate noticeable difference between the posterior distributions is the size of the 95% credible intervals. We can see that the size of the uncertainty increases as the points are moves closer to the location of the discontinuity. This is due to the sudden change in the output which occurs because of the discontinuity; the increase happens in a much smaller change in the input, resulting in the length scale parameter b decreasing to reflect this lack of smoothness. For example, in the first situation we find that $b = 0.16$ is the optimum, whilst the last example had $b = 0.04$ It is noticeable that, because of these large length scale parameters, we overestimate the uncertainty around the smooth parts of

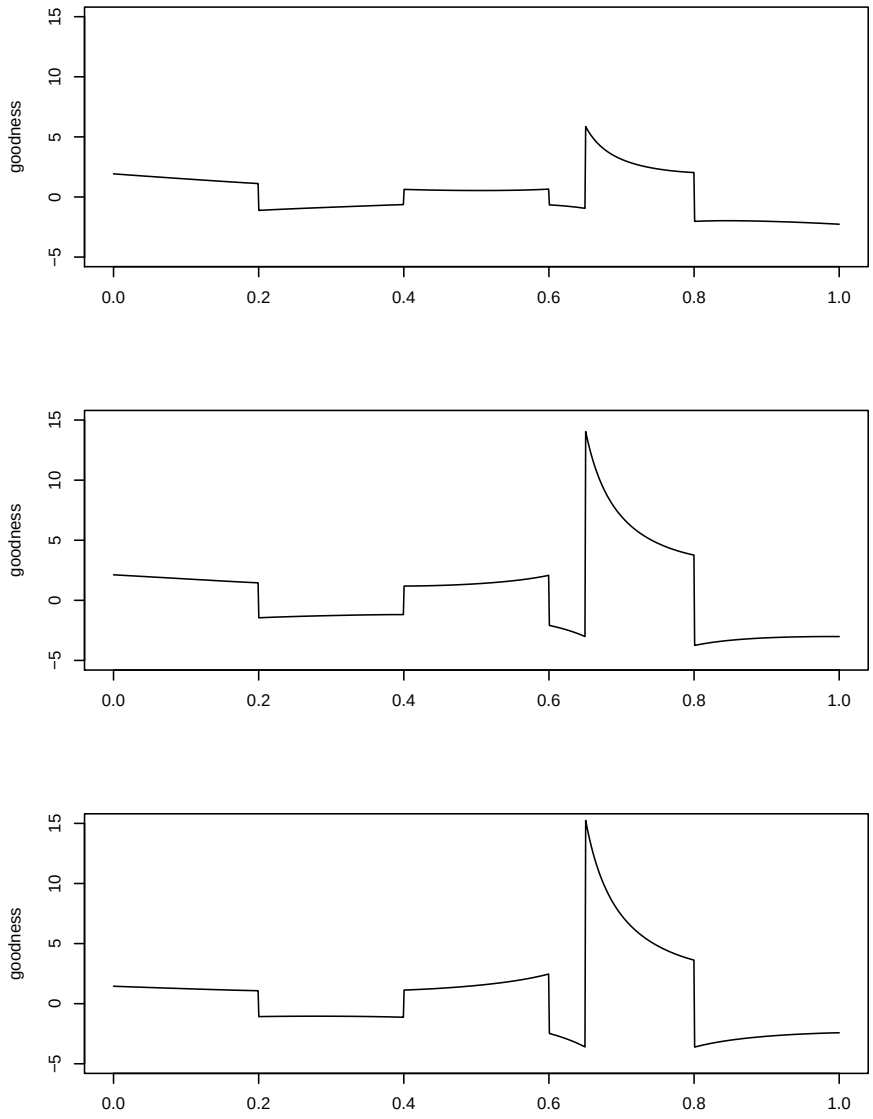


Figure 5.3: The individual prediction errors corresponding to Figure 5.2.

the function where there is little activity.

In the last exploration, we saw the effect on the posterior distribution that the placement of design points had when the function contained a discontinuity. It was seen that one of the main contributions to this posterior distribution is the estimation of the length scale parameter. Alternatively to the last exploration, we could explore the effect that moving design points have on our inference by again looking at the three different locations for (x_4, x_5), but, instead, fixing the length scale rather than optimising it. We can see the posterior distributions when this was done in Figure 5.5. Similar to the previous example, we can immediately notice the increase in the lengths of the credible intervals. In contrast to the previous example, however, we see that the function is still smooth throughout all of the examples. To cope with this smoothness in the latter datasets, the function has to over-smooth around the location of the discontinuity to deal with this

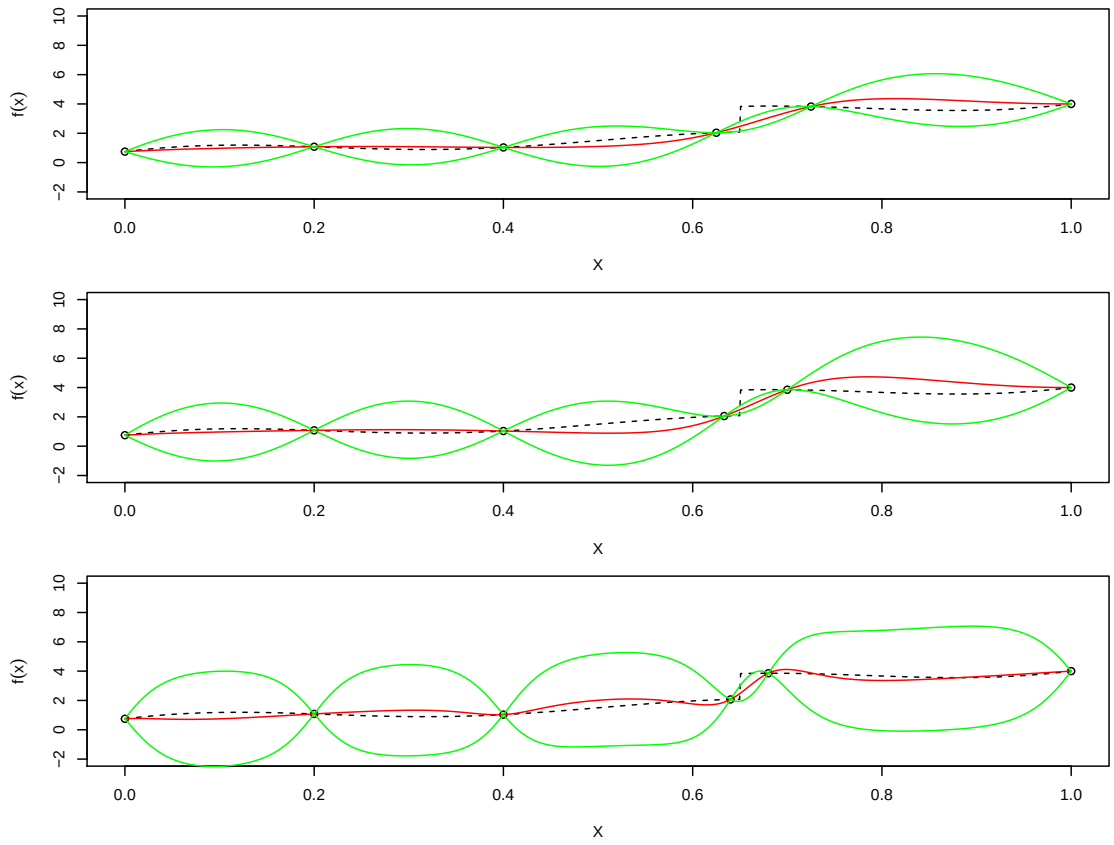


Figure 5.4: The posterior distributions of the function from equation (5.1) with $c = 1.75$, but changing the distance between the two locations closest to the discontinuity, which are x_4 and x_5. We have top: $(x_4, x_5) = (0.625, 0.725)$, middle: $(x_4, x_5) = (0.633, 0.7)$, and bottom: $(x_4, x_5) = (0.64, 0.68)$. The true function is the broken black line, the posterior mean function is the red line, and the central 95 % credible interval can be seen in green.

sharp change in the output whilst retaining the overall smoothness of the model. The closer (x_4, x_5) are to the location of the discontinuity, the more the Gaussian process has to over-smooth, creating poor estimates around the location of the discontinuity.

5.3 Partitioning the input space

There are two key aspects to our method that allow us to model non-smooth spatial processes more accurately: firstly, we partition the input space into separate regions, then, we build an independent statistical model on each of these regions to hence create a statistical model for the full input space. By partitioning the input space into different regions, we should be able to remove (or at least reduce) the effect that the discontinuity has when building our statistical model. Heuristically, the partitioning aspect of the method should partition the input space such that the boundaries of the regions are aligned with the location of the discontinuity. As such, by treating these regions as independent from each other, we are able to use the smoothness of the function, which is hopefully appropriate,

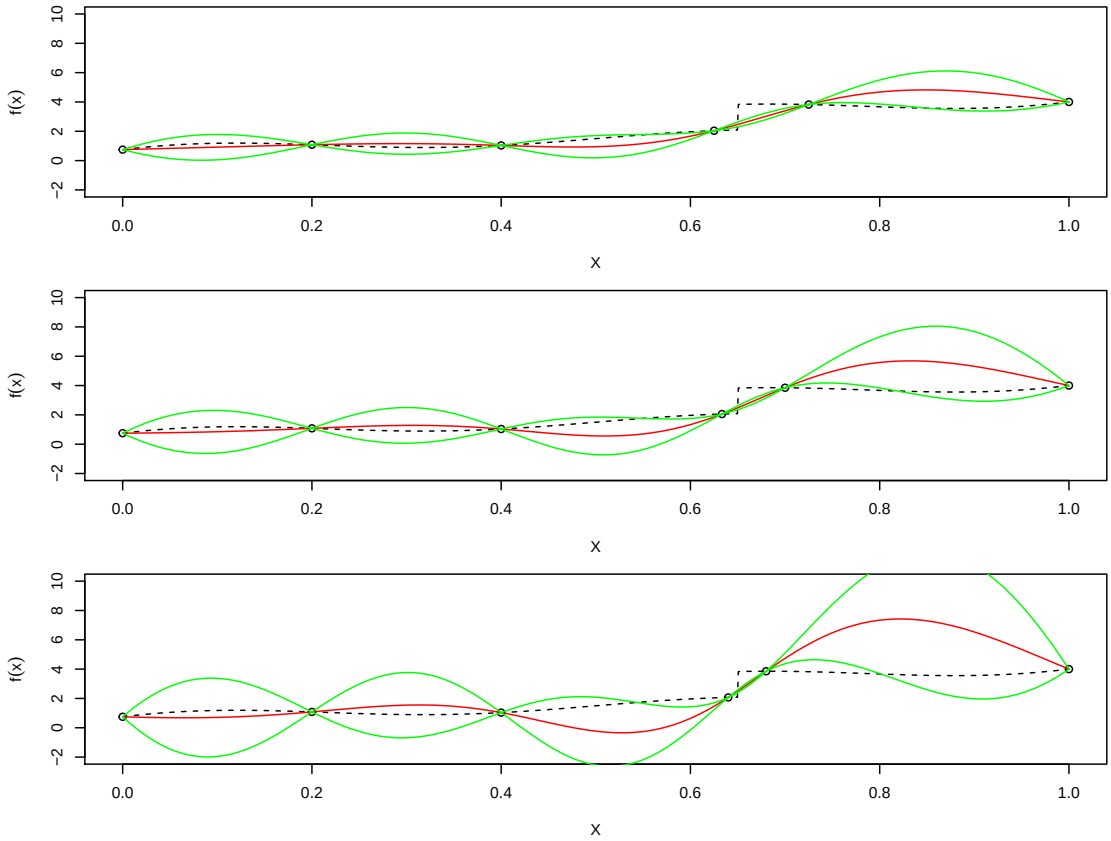


Figure 5.5: The posterior distributions of the function from equation (5.1) with $c = 1.75$, changing the distance between the two locations closest to the discontinuity, which are x_4 and x_5, and fixing the length scale parameter to $b = 1$. We have top: $(x_4, x_5) = (0.625, 0.725)$, middle: $(x_4, x_5) = (0.633, 0.7)$, and bottom: $(x_4, x_5) = (0.64, 0.68)$. The true function is the broken black line, the posterior mean function is the red line, and the central 95 % credible interval can be seen in green.

on either side of the discontinuity. Doing so ensures that we are modelling the parts of the function using only the points that are on the respective side of the discontinuity.

One thing that is therefore important in this method is how we partition the input space. Of course, in one dimension, partitioning the input space is trivial. Consider the example that we looked at in Section 5.1, specifically equation (5.1). We can see in this example that we have a jump discontinuity at $x = 0.65$, and so a natural way of partitioning the input space, given that we know the location of the discontinuity, would be to split the function with points below $x = 0.65$ forming one region and points above $x = 0.65$ forming another region. We could think of this as using a straight line perpendicular to the input axis to partition the space. This concept leads us onto one popular partition method that has been applied previously, which is treed partitioning (Gramacy & Lee 2008) and is used when we have $\dim(\mathbf{x}) \geq 2$.

Treed partitioning is the partitioning of space in which non-overlapping straight lines/slices are used that are parallel to the parameter axis. This type of partitioning is equivalent to

creating a decision tree (Quinlan 1986) in which all of the parameters are continuous. A few examples of what this partition could look like in two dimensions can be seen in Figure 5.6. We can notice from these examples that we always have one more regions than we have partition lines. This method of partitioning is very advantageous if the boundary of the discontinuity is a shape that can be modelled by straight lines parallel to the input axis. This partition method is popular due to its simplicity in use, and also due to its natural and clear interpretation of the regions. To emphasise this point, we can say that for example, for the top left example in Figure 5.6, those points in which $x_1 < 0.4$ and also have $x_2 > 0.2$, belong to the same region.

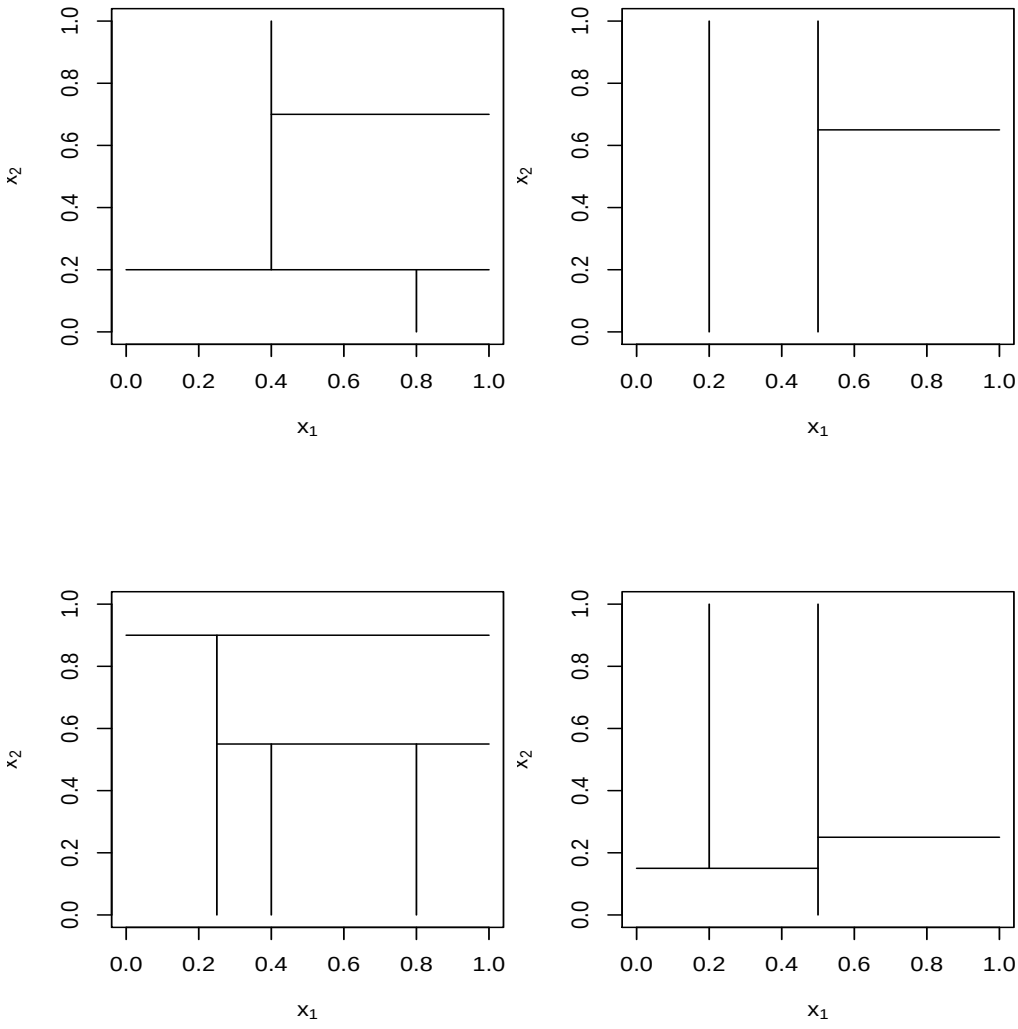


Figure 5.6: Four valid partitions of the unit space in two dimensions using treed partitioning.

The simplicity of the treed partitioning, which allows us to easily report the partition model, can also be thought of as a disadvantage. In its definition, we state that non

overlapping straight lines parallel to the axis are used to create the regions. This set-up is fine when we are considering discontinuities and regions in which their boundaries have straight lines, however, if our regions are not like this then the partitioning can struggle. Consider the situation in which we have a region that has curved boundaries, or is circular, we would find it incredibly difficult to accurately represent the region using straight lines. In theory, it is possible to model this region using straight lines, we would, however, need an infinite number of straight lines to be able to do this, and so, in practice, we will have to hope that a finite number of lines provides a good approximation.

Another partitioning method that could be applied, which may overcome the deficiencies of the Treed partitioning, is Voronoi tessellation (or Voronoi tiling) (Okabe et al. 2000). A standard Voronoi tessellation with k cells (or Voronoi regions) is defined by a set of k centres, $\mathbf{s} = \{\mathbf{s}_1, \ldots, \mathbf{s}_k\}$. An arbitrary point $\mathbf{x} \in \mathcal{X}$ is contained in the cell of the ith centre $\mathbf{s}_i$ if

$$d(\mathbf{x}, \mathbf{s}_i) < d(\mathbf{x}, \mathbf{s}_j) \; \forall \; j \in \{1, \ldots, k\} \backslash i,$$

where $d(\cdot, \cdot)$ is some distance measure, and $\mathcal{X}$ is the space of $\mathbf{x}$. Typically, the distance measure that is used is the Euclidean distance. When a Voronoi tessellation is formed using Euclidean distance as the distance measure, we have the property that, if we have a finite number of unique disjoint centres in finite-dimensional Euclidean space, all of the Voronoi cells are convex polytopes (Gallier 2008), which are higher dimensional polygons such that any two points inside the polytope can be joined using a straight line that is contained the shape. This means, for example, that if we are in two dimensional Euclidean space, all of the resulting cells will be convex polygons. Although in some texts, the areas formed by the Voronoi tessellations are referred to as regions, we will refer to them as cells as to avoid any confusion that may arise later in the thesis in which a region is defined differently than a Voronoi region. Four examples of valid Voronoi tessellations in the two dimensional unit space using a Euclidian distance measure can be seen in Figure 5.7.

We are able to see the flexibility of the Voronoi tessellation in Figure 5.7. It is immediately noticeable that, unlike how the Treed partition is constructed, we are not restricted by the boundary lines being forced to be parallel to the parameter axes. Therefore, we can see how Voronoi tiling would be advantageous over the Treed partitioning if we have region boundaries that are not parallel to the parameter axes.

For our method, we must specify a technique for partitioning the input space. Both of the methods mentioned previously partition the input space to give regions with straight boundaries: treed partitioning does so using non-overlapping lines parallel to the input axes and Voronoi tessellation uses the Euclidean distance from a set of centres to create Voronoi cells. In this chapter, we shall focus on partitioning the input space using Voronoi tessellation due to its flexibility compared to treed partitioning.

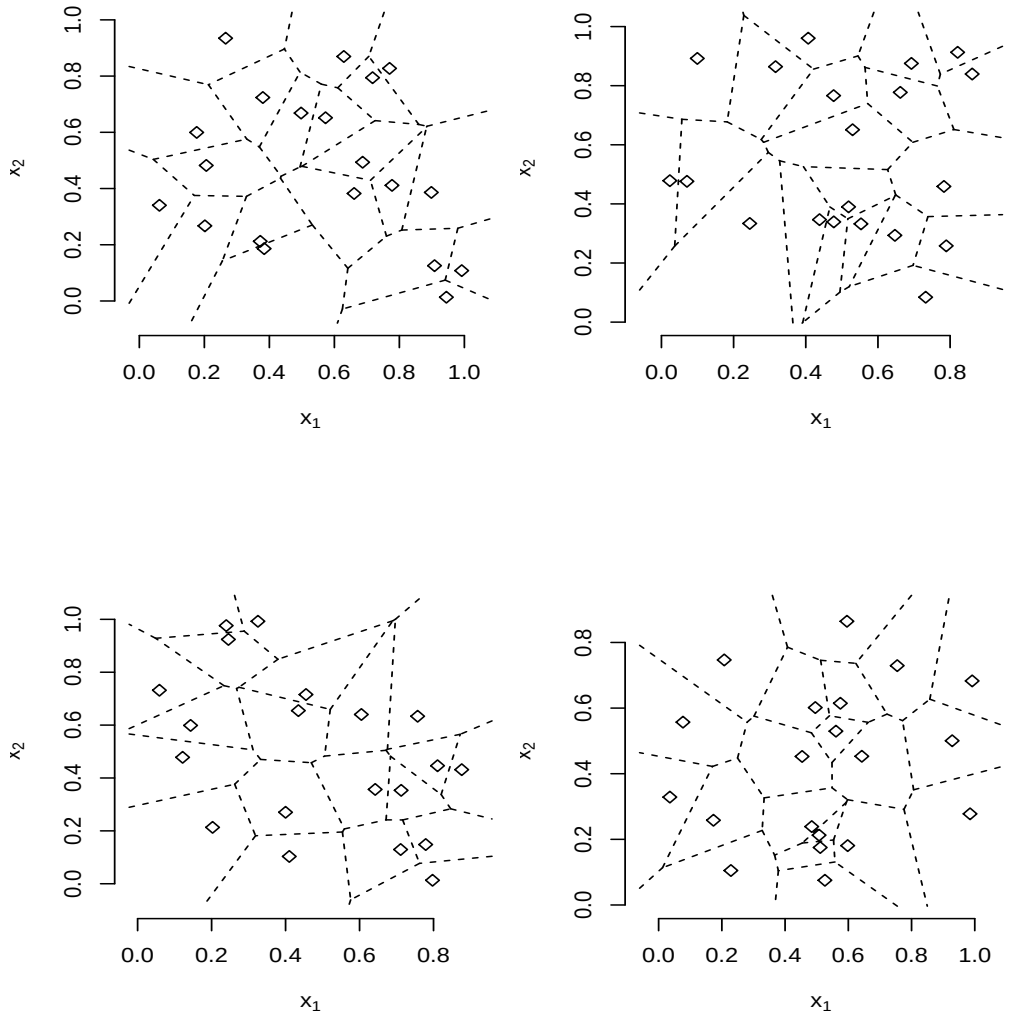


Figure 5.7: Four valid Voronoi tessellation on the unit space. Each of these tessellations has twenty uniformly distributed centres and uses the Euclidean distance as its distance metric.

We allow for discontinuities and changes in the behaviour of $f(.)$ over the input space by partitioning the input space $\mathcal{X}$ into r disjoint regions $\mathbf{R} = \{R_1, \ldots, R_r\}$, where $R_i \subseteq \mathcal{X} \; \forall \; i \in \{1, \ldots, r\}$ and $\bigcup_{i=1}^{r} R_i = \mathcal{X}$. In our method, we allow greater flexibility in our partition by allowing Voronoi cells to join together to create larger, more flexible, and possibly non-convex joint regions. This definition of regions entails that a single region could be made up of multiple Voronoi cells; hence, the earlier decision to call these areas cells to help avoid confusion. An example of this can be seen in Figure 5.8.

This idea of using Voronoi tessellations to partition the input space has been explored before in Kim et al. (2005). The models which are built using Voronoi tessellations in Kim et al. (2005) are a special case of our Voronoi structure where there is no dependence

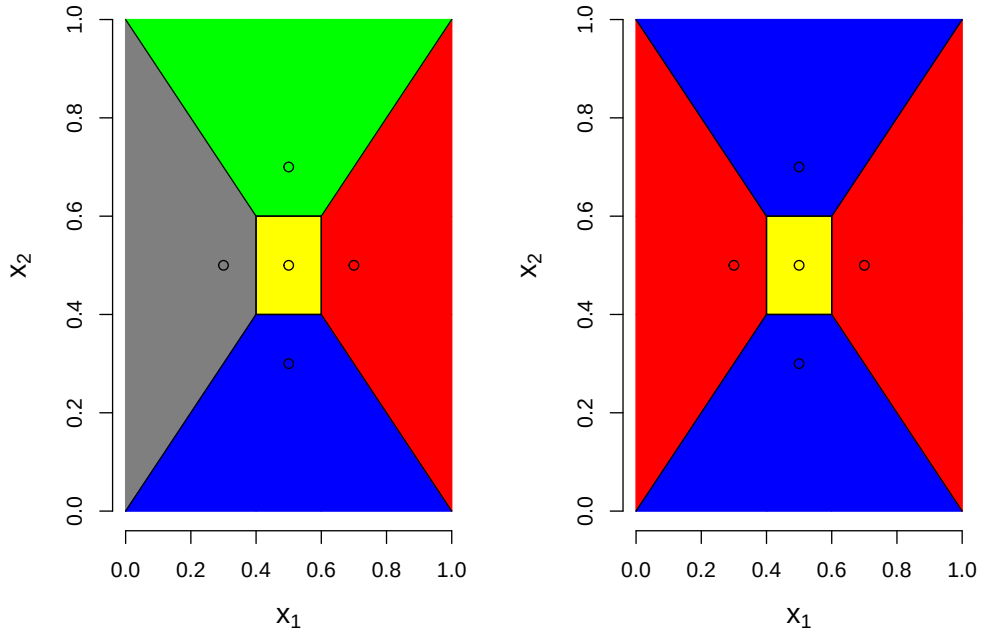


Figure 5.8: Possible setups for a Voronoi tessellation, using a simple two dimensional example with five centres. Left: All cells are independent, forming five regions. Right: Cells have relationships, forming three regions, with a region denoted by a colour.

between cells, which we allow for when the Voronoi cells merge, and severe constraints are applied to the locations of the centres; we hence refer to this method as the standard method. The joining of cells in our tessellation allows more complex regions, such as when one region is surrounded by another, or non-convex shaped regions, without the loss of information that is intrinsic to building regions that can only be a single independent cell. This idea is again highlighted in Figure 5.8, if we had the situation in which the centre tile was a region, and the other region was the remaining space in the unit space, we would have to model the remaining space using four separate regions if we do not consider the joining of cells. Modelling these cells independently has the implication that we are not using all of the points from the function's region to create our estimate and are instead using only a portion in each estimate.

Very importantly, we also look to allow a greater range of models than the standard Voronoi model of Kim et al. (2005) by changing the prior distribution of the centres that defines the cells of the tessellation. In Kim et al. (2005)'s paper, the locations of the centres $\mathbf{s}$ are restricted such that they can only be the locations of the data points, that is, the $\mathbf{x}$ locations in which we have sampled the function f, with a discrete uniform prior used over these locations. This is done so that the potential locations and probability of these locations are easy to track and visualise. They then proceed by fitting separate

independent Gaussian processes on each region. If, for example, we have an input space that is made up of two separate functions, we would ideally want to model this using two disjoint Gaussian processes. This can only be modelled accurately using the standard method if the two functions lie in parts of the input space that can be modelled by two centres at data point locations in $\mathbf{X}$.

We could still attempt to model a setup such as this in the standard method using a larger number of independent centres; however, there are two clear drawbacks to doing so. First, by splitting up a true region into multiple independent cells, we are not using all of the points from the true region to estimate the parameters of the Gaussian process, namely $\boldsymbol{\beta}, b, \sigma_\epsilon^2$ and σ^2. By using a single region, as we allow for in our approach, all points from the region can be utilised simultaneously to gain better estimates of the parameters of interest $(\boldsymbol{\beta}, \sigma^2, \sigma_\epsilon^2, b)$, assuming of course, that we are using the correct spatial partition. Secondly, using a weak prior distribution for the Gaussian process parameters has the constraint that we need at least four points to build a Gaussian process with a defined variance, which could make accurately modelling a function with a discontinuity impossible. That is, if we observe equation (2.11), we can see that using a constant mean for our Gaussian process requires four or more observations of f to produce a valid Student-t distribution, with more complex choices of mean function requiring more observations of f. We may, for example, only have five points sampled in a given region and may not be able to model this region with one cell, making it inadvisable to split this into multiple regions.

We do not require the Voronoi cells to share a vertex to be in the same region, and we do not restrict the centres to be the locations $\mathbf{x}_i$ in which we have observed the function f. However, we note that there are potential identifiability issues here due to the fact that a region in one model which consists of multiple cells joined together can be equivalent to a region in another model consisting of a single cell.

5.4 Voronoi tessellation with joint centres

5.4.1 The prior parameters and likelihood

For our method, we define the collection of tessellation parameters $\mathbf{t} = \{\mathbf{s}, k, r, c\}$ and assign the prior distributions

$$\begin{aligned}
\pi(\mathbf{t}) &= \pi(k, \mathbf{s})\pi(r|k)\pi(c|k), \\
k, \mathbf{s} &\sim \mathrm{PoiPr}(\lambda), \\
r|k &\sim DU(1, k), \\
c|k &\sim DU\left(1, b_k\right),
\end{aligned}$$

where b_k is the number of all possible ordered partitions, the kth Bell number (Aigner 1999), in which a Bell number describes the total number of unique ordered partitions a set can make, $\mathrm{PoiPr}(\lambda)$ is a Poisson process over $\mathcal{X}$ with suitable intensity parameter λ (it is suggested that a few different values for λ are simulated from until a suitable prior is found), we define $\mathcal{C}$ to be a space such that each element of $\mathcal{C}$ is one of the possible relationships between the k centres, then $c \in \mathcal{C}$ is an index of which relationship is used, and $DU(1, k)$ is a discrete uniform distribution on $\{1, \ldots, k\}$. It should be noted that λ is the only hyper-parameter that needs choosing. In practice, we different values of λ should be trialled to find a prior that represents our belief, with larger values of λ incorporating the prior belief that we believe there will be more centres in the function, and smaller values of λ representing the converse and favouring a simpler model. In our experience, we have found that the choice of λ appears to have a negligible effect on our model, with posterior inference essentially the same for a large range of λ values. In all of the examples that have used this model, a λ value of one is used. There are many adjustments that could be made to incorporate prior beliefs about the underlying model. For example, one adjustment that could be made if appropriate is to replace the Poisson process by one that includes a repulsion term, such as a Gibbs process (Illian et al. 2008). Using a repulsion term would have the benefit of additional centres having a localised effect on the model tessellation.

The likelihood of the model is

$$\ell(\mathbf{t}, \mathbf{b}, \boldsymbol{\beta}, \sigma^2; \mathbf{D}) \propto \prod_{i=1}^{r} f_i(\mathbf{y}_i | \mathbf{x}_i, b_i, \sigma_i^2, \beta_i, \mathbf{t}),$$

where $f_i(\mathbf{y}_i | \mathbf{x}_i, b_i, \sigma_i^2, \sigma_\epsilon^2, \beta_i, \mathbf{t})$ is the multivariate Gaussian distribution for outputs $\mathbf{y}_i$ corresponding to inputs $\mathbf{x}_i$ which lie in the ith region, and $\mathbf{D}$ is defined in page five. We can analytically integrate over $\boldsymbol{\beta}$ and $\boldsymbol{\sigma}^2$, using a prior distribution similar to that seen in Section 2.1.2, in which each region has the prior distribution

$$(\boldsymbol{\beta}, \sigma^2) \propto \sigma^{-2}$$

to give us our posterior distribution

$$\pi(\mathbf{b}, \mathbf{t} | \mathbf{D}) \propto \prod_{i=1}^{r} |H_i^T A_i^{-1} H_i|^{-\frac{1}{2}} |A_i|^{-\frac{1}{2}} \Gamma\left(\frac{(n_i - q)}{2}\right) \left(\frac{2}{(n_i - q - 2)\hat{\sigma}_i^2}\right)^{\frac{(n_i - q)}{2}},$$

where n_i is the number of data points in the ith region, $\Gamma(\cdot)$ is the gamma function and H_i, A_i, and $\hat{\sigma}_i^2$ are as defined in Chapter 2, with the subscript i showing that these terms are evaluated using the points that lie in the ith region. If we observe the function with noise, , σ_ϵ^2, similar to equation(2.14), we add this term to our model and treat in the same manor as Chapter 2.

The posterior distribution for $\mathbf{b}$ is analytically intractable, and, to deal with this, we select the parameter using optimisation techniques on the likelihood of $\mathbf{b}_i$, as we similarly do with σ^2_ϵ. We could also take into account our uncertainty in $\mathbf{b}$ by placing a prior distribution on $\mathbf{b}$, such as using the log-normal distribution, and including it within our MCMC method (Haylock 1997). We do not include this as the posterior distribution is sensitive to the choice of prior distribution, and a lack of prior knowledge is often the case for this parameter.

5.4.2 MCMC implementation

We use reversible-jump Markov chain Monte Carlo (RJMCMC) to sample our model parameters (Green 1995). We have two model elements to update that we need to account for in our move types: the set of centres for the tessellation and the relationship between the centres. We are able to account for this update by utilising the 'moves' aspect of an MCMC sampler, as described in Voss (2013). To update the set of centres/cells, we use a similar set up to that seen in Gelman et al. (2013) and Kim et al. (2005), and we add, take away or move a centre: these moves are called *birth*, *death* and *move* respectively. To update the relationship between the centres, we change a single centre to be in a different region (possibly a new region with no other centre); this move is called *change*. This gives us four possible general moves. These four types of proposal are taken to be equally likely during the proposal step. We use an acceptance ratio that is the same as that described in Green (1995), which has the form

$$\alpha = \min\left(1, \text{Likelihood ratio} \times \text{Prior ratio} \times \text{Proposal ratio}\right).$$

Due to the set-up of the moves, we find that the acceptance ratio simplifies to the ratio of the posterior of the proposed model to that of the existing model. As we cannot have a death when we have one centre, and we also cannot change the relationship of the centre, we only propose birth and move steps in that situation. To maintain reversibility here, which is required for our sampler to converge to the posterior distribution, when a birth step is proposed, we multiply the acceptance ratio by $1/2$, and, conversely, we multiply the acceptance ratio by 2 when we have two centres and we propose a death step. These multiplications are called $\mathrm{adjustments}$, and we set the $\mathrm{adjustment}$ to 1 in all other cases. The choice of mixing parameter, Σ_p, should be selected to improve the number of times a proposal is accepted, with large values leading to a very small number of proposals being accepted and a small value leading to a large number being accepted and hence the need for thinning. Pseudo-code detailing the RJMCMC steps using the prior distribution from Section 5.4 is given in Algorithm 2.

In Figures 5.9 - 5.12, we show illustrations of the four possible move types of our RJMCMC for a made-up example. In these figures, if two centres are in the same region,

Algorithm 2 The RJMCMC implementation of the Joint centre Voronoi Gaussian process

Begin with a random valid model setup $\mathbf{t}_0$;
for $i = 1, \ldots, n_s$ **do**
  Propose *Birth*, *Death*, *Move* or *Change* with equal probability;
  if *Birth* proposed **then**
    $\mathbf{s}_{(k+1)} = P; P \sim \mathrm{U}_k \mathcal{X}$;
    Update $\mathbf{c}$ such that $\mathbf{s}_{(k+1)}$ is related to another region or independent;
  else if *Death* proposed **then**
    Remove one element of $\mathbf{s}$ at random;
    Remove the chosen point's relationship from $\mathbf{c}$;
  else if *Move* proposed **then**
    Select an element of $\mathbf{s}$ at random, $\mathbf{s}_j$ say;
    Propose a new centre $P \sim N_k(\mathbf{s}_j, \Sigma_p)$, with Σ_p tuned for mixing;
    Set $\mathbf{s}_j = P$;
  else if *Change* proposed **then**
    Select an element of $\mathbf{s}$ at random, $\mathbf{s}_j$ say;
    Change $\mathbf{s}_j$'s relationship in $\mathbf{c}$ s.t. it is independent or related to a different region;
  end if
  Update the current model $\mathbf{t}_i$ to obtain proposed model $\mathbf{t}_p$;
  Fit an independent Gaussian process to each region in $\mathbf{t}_p$;
  Choose b and σ_ϵ^2 to be the values that maximise the likelihood from equation (5.3);
  Calculate the likelihood of the proposed model $\pi(\mathbf{b}, \mathbf{t}_p, \sigma_\epsilon^2 | \mathbf{D})$;
  Generate $U_i \sim U[0, 1]$;
  if $U_i \leq \frac{\pi(\mathbf{b}, \mathbf{t}_p, \sigma_\epsilon^2 | \mathbf{D})}{\pi(\mathbf{b}, \mathbf{t}_{i-1}, \sigma_\epsilon^2 | \mathbf{D})} \times \mathrm{Adjustment}$ **then**
    $\mathbf{t}_i = \mathbf{t}_p$;
  else
    $\mathbf{t}_i = \mathbf{t}_{i-1}$;
  end if
end for

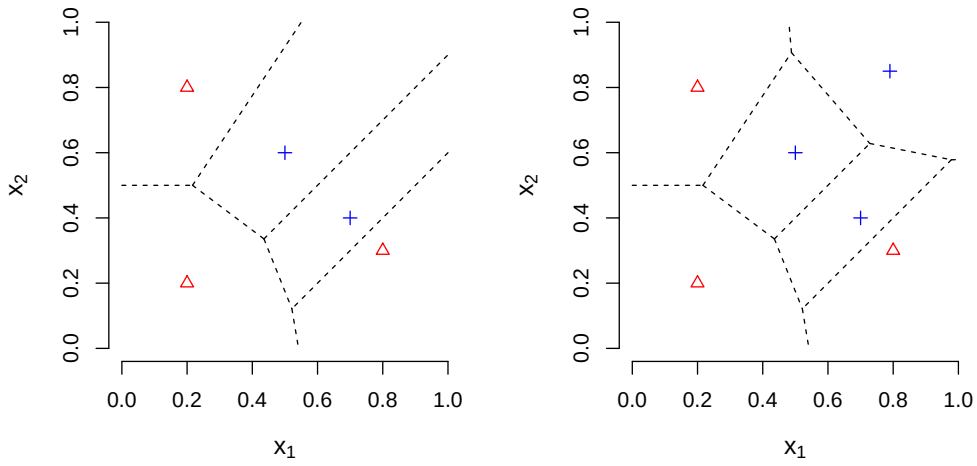


Figure 5.9: An example of a Birth move proposal for our RJMCMC. On the left we have the current Voronoi tessellation, in which we have region one which has centres denoted by red triangles and region two which has centres denoted by blue crosses. On the right we have the proposed model if we were to have a birth and we had $s_{k+1} = (0.79, 0.85)^T$ that was related to region two.

then this is shown by their centres having the same colour and shape. For the moves, we start with the same current model, in which we have $x_1 = (0.2, 0.2, 0.7, 0.8, 0.5)^T$, and $x_2 = (0.2, 0.8, 0.4, 0.3, 0.6)^T$, using the unit square as the space for x. We then show possible valid models that could be proposed through each respective move type. In Figure 5.9 we have show the case in which the new centre joins to an existing region; it should also be noted that this centre forming a region on its own (an independent region) is also possible. Similarly, in Figure 5.12, as opposed to forming its own region, the centre that was selected could also have joined region two.

After the RJMCMC update of the tessellation, we are then able to fit a statistical model to each region, as mentioned earlier, our model of choice is the independent Gaussian processes. We can then use the Gaussian process model on each region to make predictions at points in that region. It is worth noting that the sampling of the posterior distribution is sensitive to the starting state of the sampler, with many local maximums typically occurring in the highly complex model space. To ensure that we have convergence to the true posterior distribution and adequate mixing, it is recommended that a range of starting states are used. These chains should then be observed to check that the sampler is not

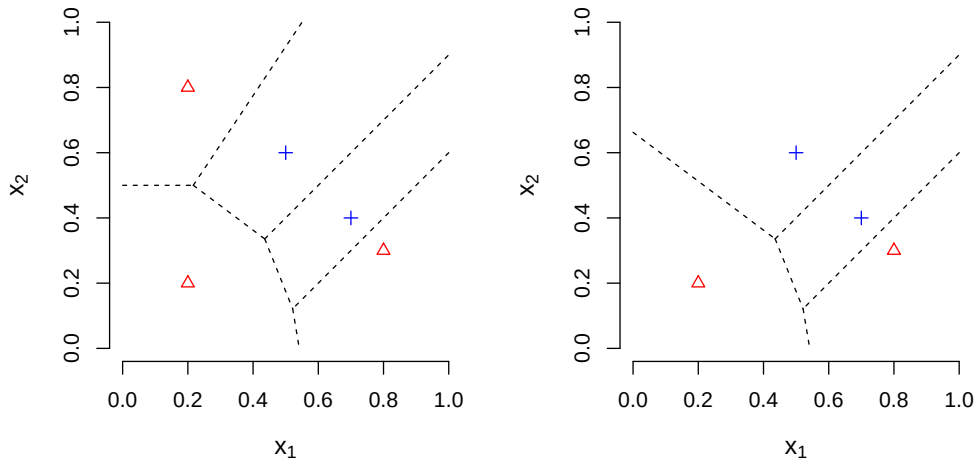


Figure 5.10: An example of a Death move proposal for our RJMCMC. On the left we have the current Voronoi tessellation, in which we have region one which has centres denoted by red triangles and region two which has centres denoted by blue crosses. On the right we have the proposed model if we were to have a death move and we removed $s_k = (0.2, 0.8)^T$.

continuously sampling near a part of low probability space due to the existence of a local maximum. It is suggested that those chains that do so are removed from our posterior distribution; we could, for example, select those five chains that have observed the largest posterior pdf value to continue with our posterior sample and remove the samples from all other chains.

5.5 Adaptive sampling to identify discontinuities

In some applications, it may be possible to gather additional data at new training points $\mathbf{x}^*$. In many cases, this is costly and/or time consuming, so these values of $\mathbf{x}^*$ must be chosen with care. In particular, we may wish to sample points such that we estimate any discontinuities or borders between regions more accurately. When there is the existence of a discontinuity or a sharp change in the output of the function, the prediction is often poor around the location that it occurs (see Figure 5.3). Being able to model these borders and the location of any discontinuity could, hence, potentially have a huge impact on future inference. Having more information around the discontinuity will not only help

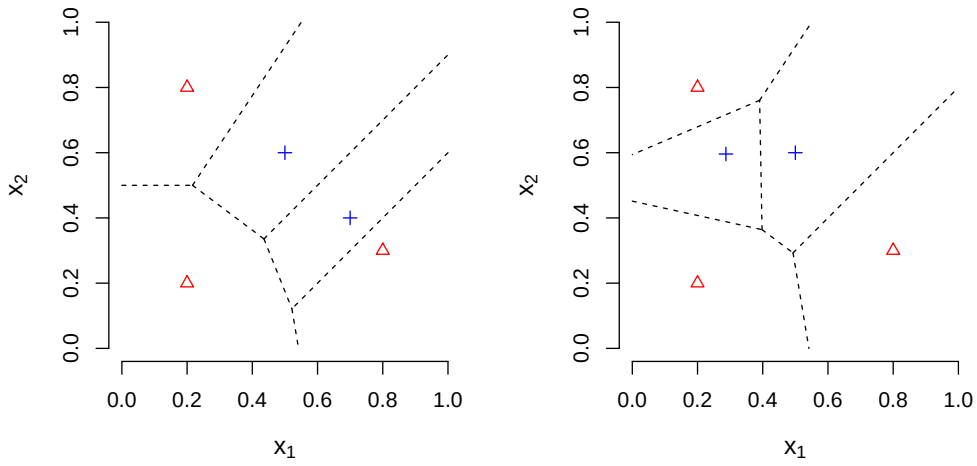


Figure 5.11: An example of a Move proposal for our RJMCMC. On the left we have the current Voronoi tessellation, in which we have region one which has centres denoted by red triangles and region two which has centres denoted by blue crosses. On the right we have the proposed model if we were to have a Move and we change $s = (0.7, 0.4)$ to $s = (0.287, 0.596)^T$.

us predict outcome values at unobserved locations with more accuracy, but will also supplement the understanding we have about where the discontinuities are occurring, which is often of practical interest. Existing sampling methods such as space filling algorithms, that is, selecting the location to sample such that we maximise our coverage of the parameter space, and largest uncertainty samplers (Santner et al. 2003), in which we select the location of our sample such that we choose the location that has the largest pointwise posterior variance, are not tailored to this objective.

The (approximate) MAP model is found by looking at which tessellation in our posterior sample has the largest likelihood value. Heuristically, we could think of this MAP model as our best estimate of the location of the boundaries of the discontinuity, in light of the data that we have observed. We then propose the following sampling method to help estimate these boundaries: We use our MAP model, and sample points on the boundary of the region that we want to sample from in this model, as this is a good estimate of the boundary of the discontinuity. Any point on the boundary of the region would hence be a sensible candidate choice for our boundary sample. There are an infinite number of positions that we could sample on the boundary, and so we should attempt to maximise

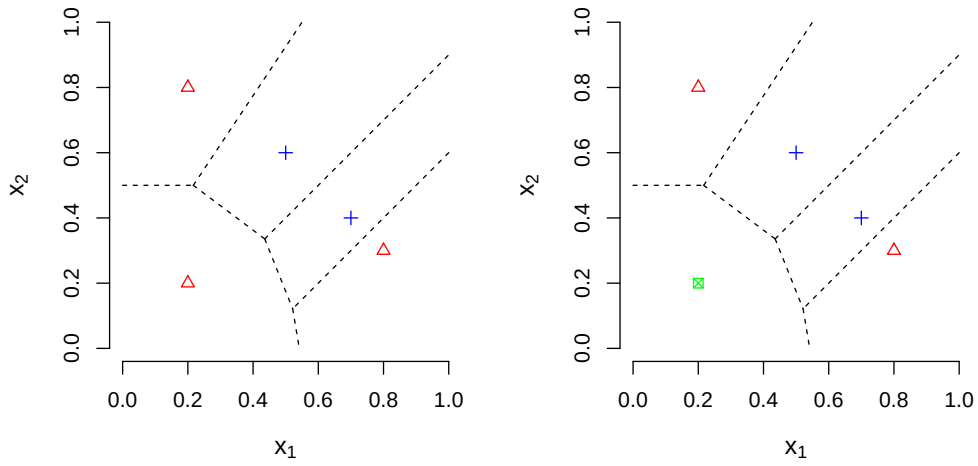


Figure 5.12: An example of a Change move proposal for our RJMCMC. On the left we have the current Voronoi tessellation, in which we have region one which has centres denoted by red triangles and region two which has centres denoted by blue crosses. On the right we have the proposed model if we were to change the relationship of $s = (0.2, 0.2)^T$ so that it belongs to its own independent region, denoted by a green crossed square.

the information we get from each sample. To do so, we iteratively choose points on the boundary that are furthest from all existing design points to try to attain some of the properties that are established for space filling designs (e.g. Pronzato & Müller 2012). The algorithm for this sampling method is given in Algorithm 3. We can see a simple example of this kind of sampling in the top panel of Figure 5.13.

We may note that it is trivial to extend this sampling method to sample any generic region or multiple regions. For example, if we are interested in a specific boundary between two regions, we simply look at the boundary separating the two regions in question, and we can choose a point to sample from the points that appear on both region boundaries. Of course, the method in Algorithm 3 is not the only sampling method that we could use that is tailored to our objective. A change could be made to the algorithm if we are able to double the number of points that we can sample. Instead of sampling at a point $\mathbf{x}^*$ when sampling the boundary of region i, we could look at the line that interpolates $\mathbf{x}^*$ and the centre of its corresponding cell $\mathbf{s}_i$, and sample two points on this line at distances $||\mathbf{x}_j^* - \mathbf{s}_i|| \pm \epsilon$ from the centre. That is, rather than sampling at the point $\mathbf{x}^*$, we sample at points $\mathbf{x}^* \pm \epsilon|\mathbf{s}_i - \mathbf{x}^*|$, where $\epsilon << 1$. This adaptation should, in theory, sample just

inside and just outside of the discontinuity if ϵ is chosen suitably. A visualisation of this method can be seen in the bottom panel of Figure 5.13.

It would also be straightforward to combine this sampling approach with sampling points in regions of highest posterior uncertainty if we wished to improve both boundary detection and function estimates. An example of this would be, if we had enough resources to sample a further n_p points, we could sample n_p^* points using the boundary sampler, in which $n_p > n_p^*$, and sample the remaining $(n_p - n_p^*)$ points at those locations with the largest uncertainty.

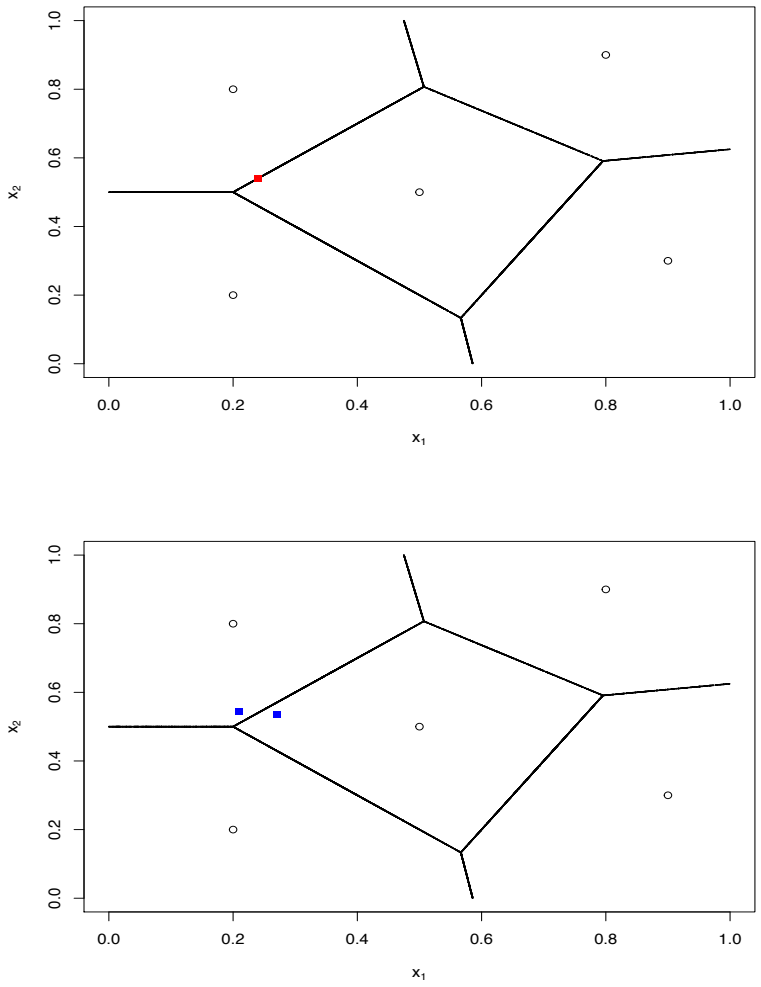


Figure 5.13: An example of using the boundary sampler when we have five centres. The centre locations are denoted by circles, the boundaries of the Voronoi tessellation cells are the black lines. In the top figure, we see the new proposed point in red when we use the original sampling method; in the bottom figure, we see the new proposed points in blue when $\epsilon = 0.1$ for the adaptation.

Algorithm 3 The boundary sampling method

Require: $n_p > 0$ - number of points to sample;

Require: $\mathbf{X}$ - the n locations where we have observed data;

Implement the method from Section 5.4 to gain a posterior sample of models;

Find the MAP model from these posterior samples;

Identify the region on whose border that we want to sample;

Choose a candidate set of n^* points, $\tilde{\mathbf{x}}$, to sample at by looking at the boundary of this region;

for $k = 1, \ldots, n_p$ **do**

Select the point in $\tilde{\mathbf{x}}$ such that

$$\max_{i \in 1,\ldots,n^*} \min_{j \in 1,\ldots,n} d(\tilde{\mathbf{x}}_i, \mathbf{x}_j),$$

that is, which point i in $\tilde{\mathbf{x}}$ maximises the minimum distance to all points in $\mathbf{X}$;

$\tilde{\mathbf{x}}_i$ will be our kth point to sample;

Add $\tilde{\mathbf{x}}_i$ to $\mathbf{X}$ and update $\mathbf{X}$ and n;

end for

We sample at locations $\{\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+n_p}\}$.

5.6 Simulation studies

5.6.1 A diamond-shaped discontinuity

To initially test our modelling approach, we apply it to a deterministic test function, which is shown in Figure 5.14a. Our example has a discontinuity defined by straight lines, but these are not parallel to the parameter axes . The function is defined by

$$\eta_1(\mathbf{x}) = \begin{cases} \sin(x_1) + \cos(x_2) & \text{for } \mathbf{x} \in T, \\ \sin(x_1) + \cos(x_2) + 10 & \text{else,} \end{cases}$$

where $T = \{\mathbf{x} : x_2 - x_1 \leq 0.2 \cap x_2 - x_1 \geq -0.2 \cap x_2 + x_1 \geq 0.8 \cap x_2 + x_1 \leq 1.2\}$. It can be seen in Figure 5.14a that we have a square discontinuity in the middle of the unit space that has been rotated by 45°

We evaluate the function at 80 different inputs for our design points, the location of which are chosen using a Latin hyper-square design with a maximin criterion to get a good, even coverage of the input space (see Section 2.3). One thing of interest here is to look at the estimated surface of our model and to see how this compares to the true surface. Due to the nature of the posterior samples, any mean surface (or kriging surface) that we attain from a single sample would be conditional on the tessellation $\mathbf{t}_i$ for that sample. As such, the mean surface of a single sample from our posterior sample will

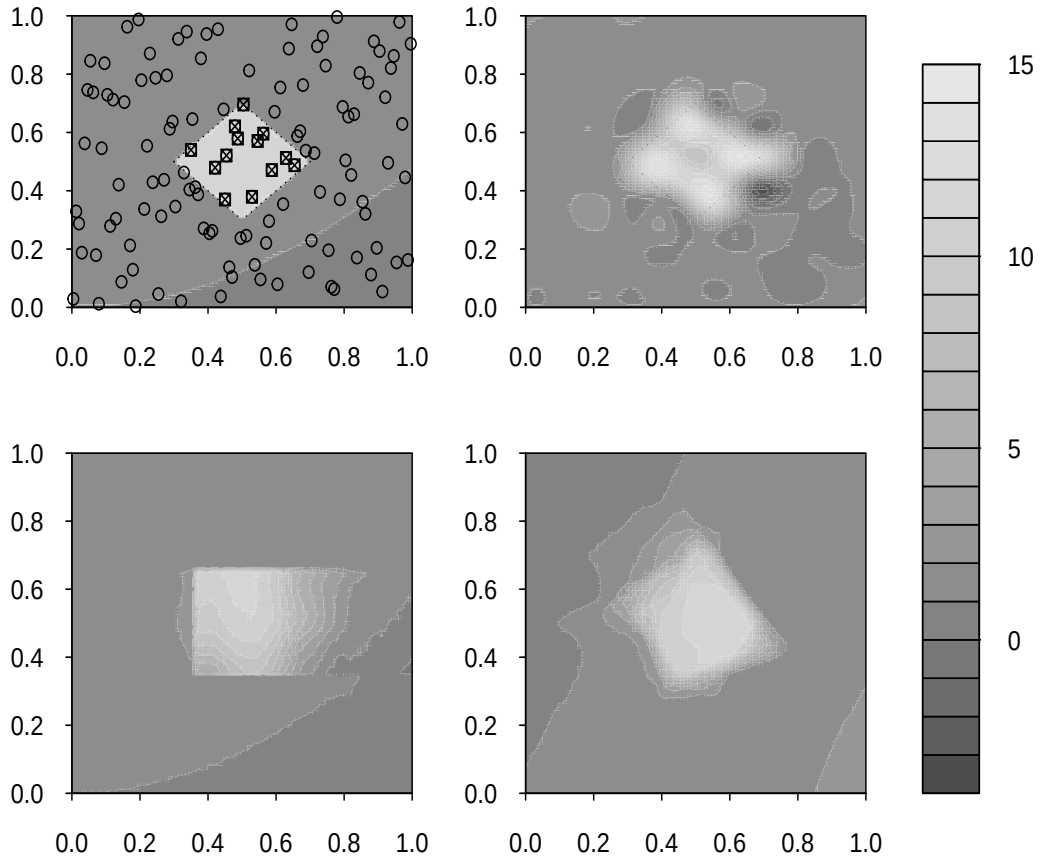


Figure 5.14: Top left (a): The true diamond test function, the design points that lie within the discontinuity are square otherwise they are shown as a circle; Top right: The standard Gaussian process mean surface; Bottom left: The TGP integrated surface; Bottom right: The integrated surface of our method.

be a piecewise Gaussian process. We can numerically integrate over $\mathbf{t}$ via Monte Carlo methods using the posterior sample, and, hence, have an integrated mean surface that is not conditional on the tessellation parameter; that is $E_{\mathbf{t}}\left(E_{\eta(.)|\mathbf{t}}(.)\right)$. This will be referred to as the *integrated surface*, whilst the surface of a single sample will be referred to as a *mean surface* henceforth to avoid confusion between the two. The use of the integrated surface will allow us to compare the accuracy of the model compared to other models that create a single surface as its prediction. To create the integrated surface for our analysis, we find the value of the mean surface for each of the posterior tessellation samples at 10,000 points (using an equispaced grid of 100×100 points), and find the mean of these points over the samples.

We compare different modelling methods to ours using the mean squared error (MSE) of the integrated surface for each method. The MSE is defined as

$$MSE(y) = \frac{1}{n}\sum_i (y_i - \hat{y}_i)^2,$$

where $\hat{y}_i$ is our estimate of y_i.

We find that the MSE of our method (MSE $= 1.84$) is smaller than that of both the Treed GP (MSE $= 1.98$) and the standard GP (MSE $= 2.04$). The MSE of our method compared to the others suggests that our approach is more representative of the true surface in this case. The integrated surfaces of all of the methods can be seen in Figure 5.14. We also note that our modelling method performs better than the convolution based Gaussian process (MSE $= 2.13$) from the R package 'convoSPAT' based on the paper by Risser & Calder (2015*b*), which involves the use of a spatially varying smoothness process within our covariance process, allowing the variance and smoothness terms to vary over the input space. That is, rather than keeping the length scale term b and the variance term σ^2 fixed, we allow it to vary over the parameter space $\mathcal{X}$, turning the two terms into functions that depend on the value of $\mathbf{x}$.

We show in Figure 5.15 examples of the tessellations in the posterior distribution for this example, so that we can observe the types of models that are proposed and accepted for our model. We also consider the performance of the adaptive sampler for this example. Following Algorithm 2, we obtained the MAP model from our posterior sample, which is shown in Figure 5.16. The MAP model has 14 cells divided into two regions, with one region containing 12 of these cells and the other region containing just two. The region with two cells, which is the region that contains all of the points from the discontinuity, is the region whose boundary we will sample on. To do this, we implement the sampler from Algorithm 3, using 2,000 candidate points on the boundary and selecting five of these points to evaluate and include in our training data.

	1	2	3	4	5	6	7
Original points	0	0.294	0.505	0.118	0.047	0.023	0.014
After sampler	0	0.595	0.280	0.060	0.021	0.031	0.006

Table 5.2: The posterior probability for the number of regions for the diamond example in Section 5.6.1 before and after the sampler was used.

We can see in Figure 5.16 that two of the points we have chosen to sample lie very close to the true discontinuity, and, around those areas, we should have a much better understanding about the location of the boundary. We will also reduce the uncertainty about the mean function around the other three points that have been sampled although these points do not lie as close to the boundary as the two previously mentioned. We compare our sampling method to two existing methods of selecting new design points: using a Sobol sequence (Giunta et al. 2003), which recursively selects the point in space that is furthest from the existing points, and selecting the points in $\mathcal{X}$ that have the largest posterior variance (see Chapter 2).

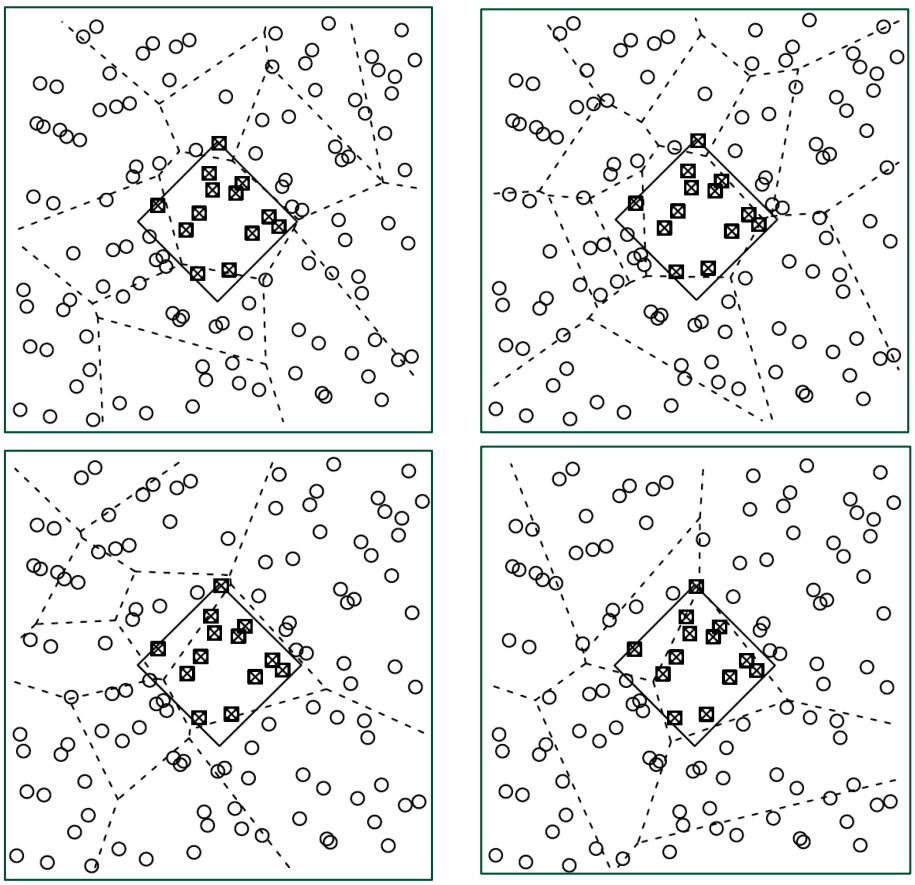


Figure 5.15: An example of four tessellations from the posterior distribution of the diamond example. The circles represent those data points that lie outside of the region T, and the crossed squares are those data points that lie inside.

We find that our sampling algorithm produces the smallest MSE (MSE=1.352) using the method in Algorithm 3 when an additional five points (Figure 5.16) are sampled compared to using Sobel (MSE=1.511) and the largest posterior uncertainty (MSE=1.392). In Table 5.2, we can see that the most probable number of regions is not equal to the true number of regions when only the original points are considered. The most probable number of regions is three. The posterior probability is larger for three regions as opposed to two is due to the Gaussian process's ability to model a gradient, which would occur when a small number of points from both sides of the discontinuity are considered a region. However, the true number of regions becomes the most probable when the additional new points are added. In fact, as we further sample more points, we become more confident that the number of regions is two $\big(\Pr(r = 2) = 0.89$ when we sample an additional ten points (Figure 5.17) using the sampler$\big)$.

5.6.2 A discontinuity with curved boundaries

The second test function is shown in Figure 5.18a. This example is a particularly difficult one as to truly represent a circular boundary using Voronoi tessellation we would need an

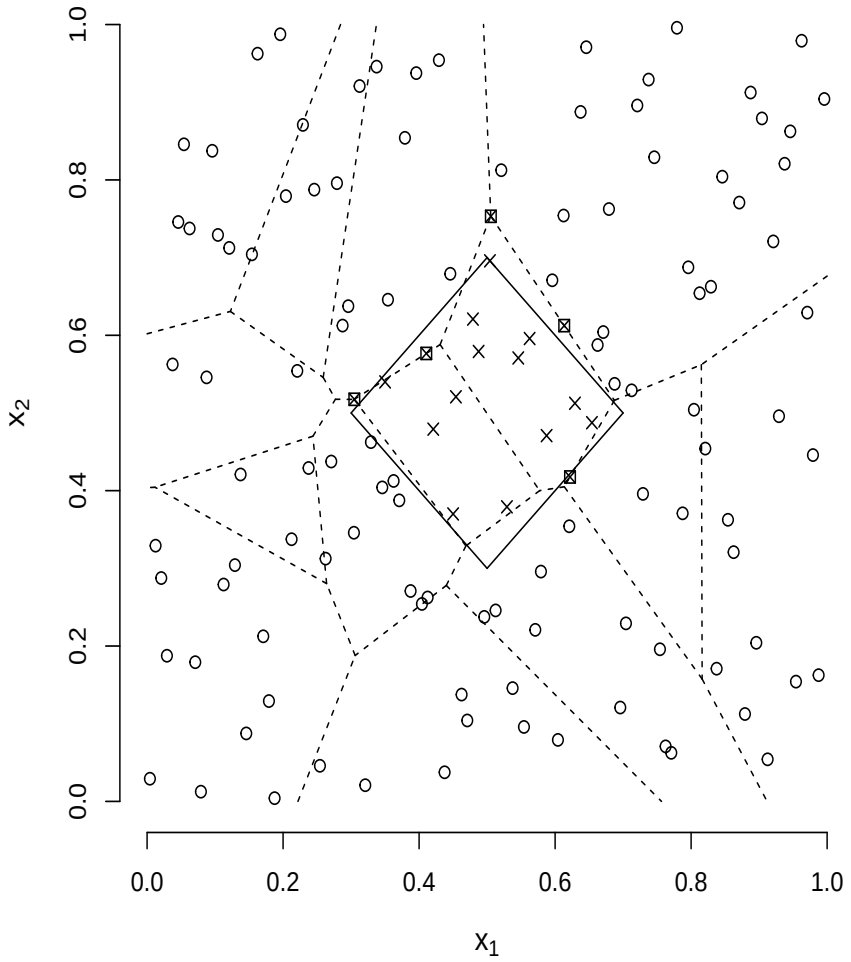


Figure 5.16: The MAP tessellation of the diamond example. The two cells that contain all of the points that lie within T form one region and the the remaining cells correspond to the other region. The design points that lie outside the discontinuity are denoted by circles, the design points that lie within T are denoted by crosses and the new points selected by our algorithm to sample are denoted by squares containing crosses.

infinite number of centres. This test function is defined as

$$\eta_2(\mathbf{x}) = \begin{cases} 10 + x_1^2 + 5x_2^2 + 3\cos(10x_1^2 + 5x_2^2) & \text{for } \mathbf{x} \in L, \\ x_1^2 + 5x_2^2 + 3\cos(10x_1^2 + 5x_2^2) & \text{else}, \end{cases} \tag{5.3}$$

where

$$L = \big\{\mathbf{x} : \{x_1 \in [0.25, 0.6] \cap x_2 \in [0.3, 0.6]\} \cup \{(x_1 - 0.25)^2 + (x_2 - 0.6)^2 \leq 0.15^2\} \\ \cup \{(x_1 - 0.6)^2 + (x_2 - 0.6)^2 \leq 0.15^2\} \cup \{(x_1 - 0.4125)^2 + (x_2 - 0.3)^2 \leq 0.175^2\}\big\}.$$

We evaluate the function at 70 different design points chosen using a Latin hypersquare design with a maximin criterion to again get a good even coverage of the input space (Johnson et al. 1990).

The integrated surface we obtain for this case on application of our method can be seen

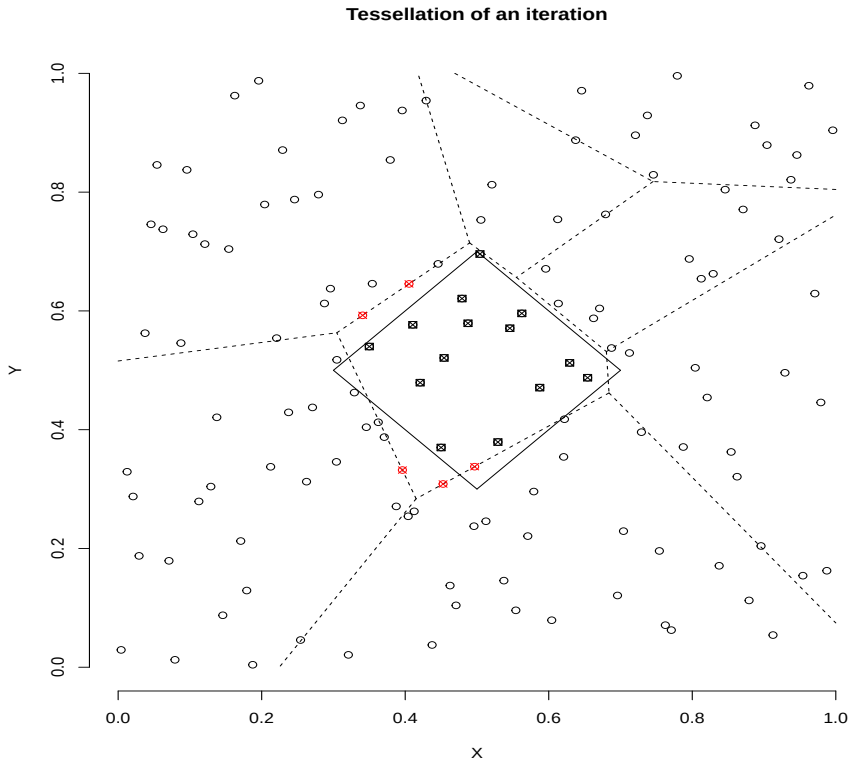
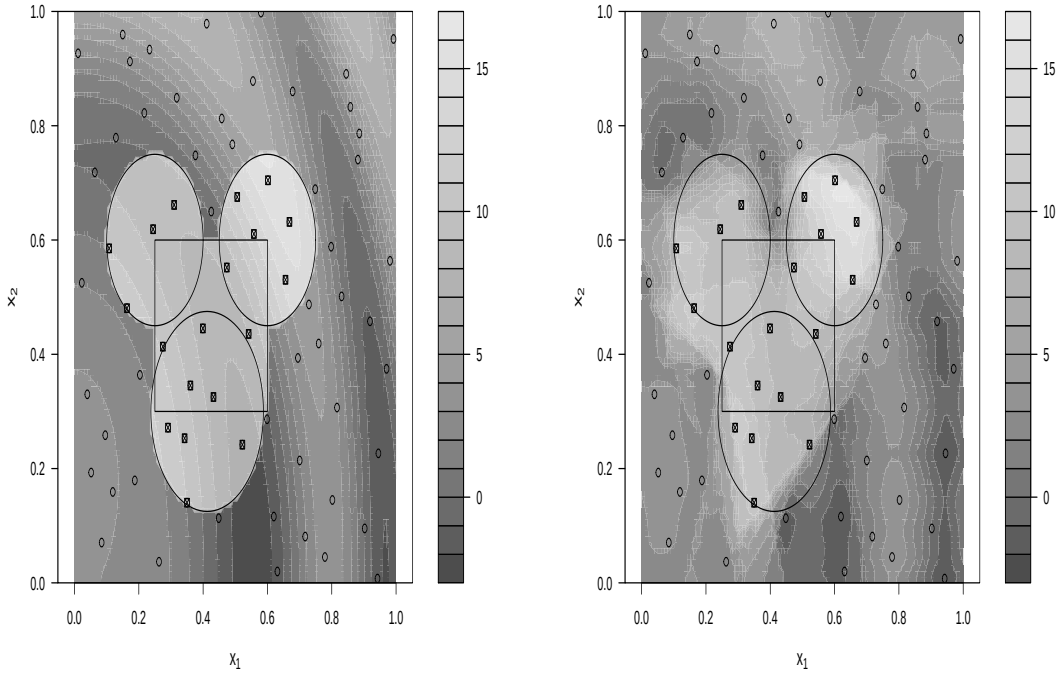


Figure 5.17: The MAP tessellation of the diamond example. The two cells that contain all of the points that lie within T form one region and the the remaining cells correspond to the other region. The design points that lie outside T are denoted by circles, the design points that lie within the discontinuity are denoted by black squares containing crosses and the new points selected by our algorithm to sample are denoted by red squares containing crosses.

in Figure 5.18b. We can see that the method has performed as well as can be expected when considering the data we have used to train it. The feature of a gap in the discontinuity between the top two circles appears to have been captured well in the integrated surface. Of course, we can measure the performance of the method by looking at the mean squared error of the integrated surface in the same way as the example in Section 5.6.1. Similar to which was seen in Section 5.6.1, the results show that our method (MSE=4.498) has a smaller MSE than treed GP (MSE=6.886) and the standard GP (MSE=6.473). This reduction in MSE again suggests that our approach is more representative of the true surface than these other estimates.

In the treed Gaussian process method, the input space is partitioned using non-overlapping straight lines parallel to the parameter axis and independent Gaussian processes are built

for each of the regions. We can see from the shape of L in Figure 5.18a that we would need a large number of these partitions to be able to get a good approximation for the the true shape of the discontinuity. A similar argument follows when we consider the shape of T in the simulated example of Section 5.6.1. The extremely large MSE values for the standard GP is due to the fact that a standard GP is inappropriate for both of these functions as the smoothness assumption is clearly being violated. As a result, the mean function must over-smooth to ensure that the function intersects the training points exactly leading to poor estimates around the discontinuity. This is similar to the problem that was described in Section 5.2, but made more difficult due to the increase in dimension of the parameter $\mathbf{x}$.



(a) The true function.

(b) The integrated surface.

Figure 5.18: Filled contour plots of the true function from equation (5.3) and the integrated surface of our method.

5.7 Applications of the method to real datasets

5.7.1 Cloud modelling

We illustrate our method on output from a complex numerical cloud-resolving model, the System for Atmospheric Modelling (SAM) (Khairoutdinov & Randall 2003). The model is used to simulate the development of shallow nocturnal marine stratocumulus clouds for 12 hours over a domain of size 40 km x 40 km x 1.5 km height, given changes in the initial conditions described through the perturbation of six key parameters (which we refer to as

$\mathbf{x}$). These simulations are an updated version (with longer run-time and updated radiation scheme) of the nocturnal marine stratocumulus simulations (set 2) described in detail in Feingold et al. (2016), and we focus here on the average predicted cloud coverage fraction over the domain in the final hour of the simulations, y.

Shallow clouds are very important to the climate system as they reflect solar energy to space and hence cool the planet, offsetting some of the greenhouse gas warming. These clouds are particularly sensitive to aerosol concentrations in the atmosphere and meteorological conditions, where small changes in temperature and humidity profiles can impact strongly on whether clouds form or not, and how thick/reflective they are. It is essential to understand how changes in aerosol and meteorological conditions can affect shallow clouds in order to improve their representation in climate models. Currently, large-scale climate model representations of shallow clouds are poor as they form and develop on smaller scales than the large grids used, yet how they are represented can have a strong influence on predictions of climate sensitivity (That is, the magnitude of warming for a prescribed increase in CO2).

Initial investigations of these SAM model simulations and expert opinion has suggested that the model potentially produces two different forms of cloud behaviour (open and closed cell behaviour) over the six-dimensional parameter space. Hence, the underlying model function that we want to reproduce with our modelling approach is potentially made up of a single function with two regimes, and will likely contain a discontinuity in y as the model behaviour moves between these regimes. As such, there is also interest in knowing about the location of any discontinuity/change in regime in order to explore where and why this phenomenon occurs.

We have 105 training points available from the simulations to build our model of the cloud coverage fraction, y, where the input combinations were chosen to cover the 6-d parameter space using a space-filling maximin Latin hypercube design. Scatter plots of the outputs against the individual inputs can be seen in Figure 5.19, in which we can see no immediately obvious way of splitting the data. We see from Figure 5.19 that most areas of the parameter space output consistent values of y at around 0.9, however some other areas have much smaller values of y, from 0.2 to 0.4, highlighting the potential existence of two regions. The plot of the output against the aerosol concentration (x_6) input suggests that high values of this parameter are very likely to yield large cloud coverage values; however, there is no clear way to differentiate low values.

One interesting aspect that we have discovered from our posterior sample is that the MAP model obtained is one that contains two regions. The posterior distribution for the number of regions is shown in Table 5.3. The belief that there are two underlying regimes (cloud behaviours) is further strengthened by these posterior probabilities, in which we see that two regions $\big(\Pr(r = 2|\mathbf{D}) = 0.667\big)$ is the most probable, despite the fact that

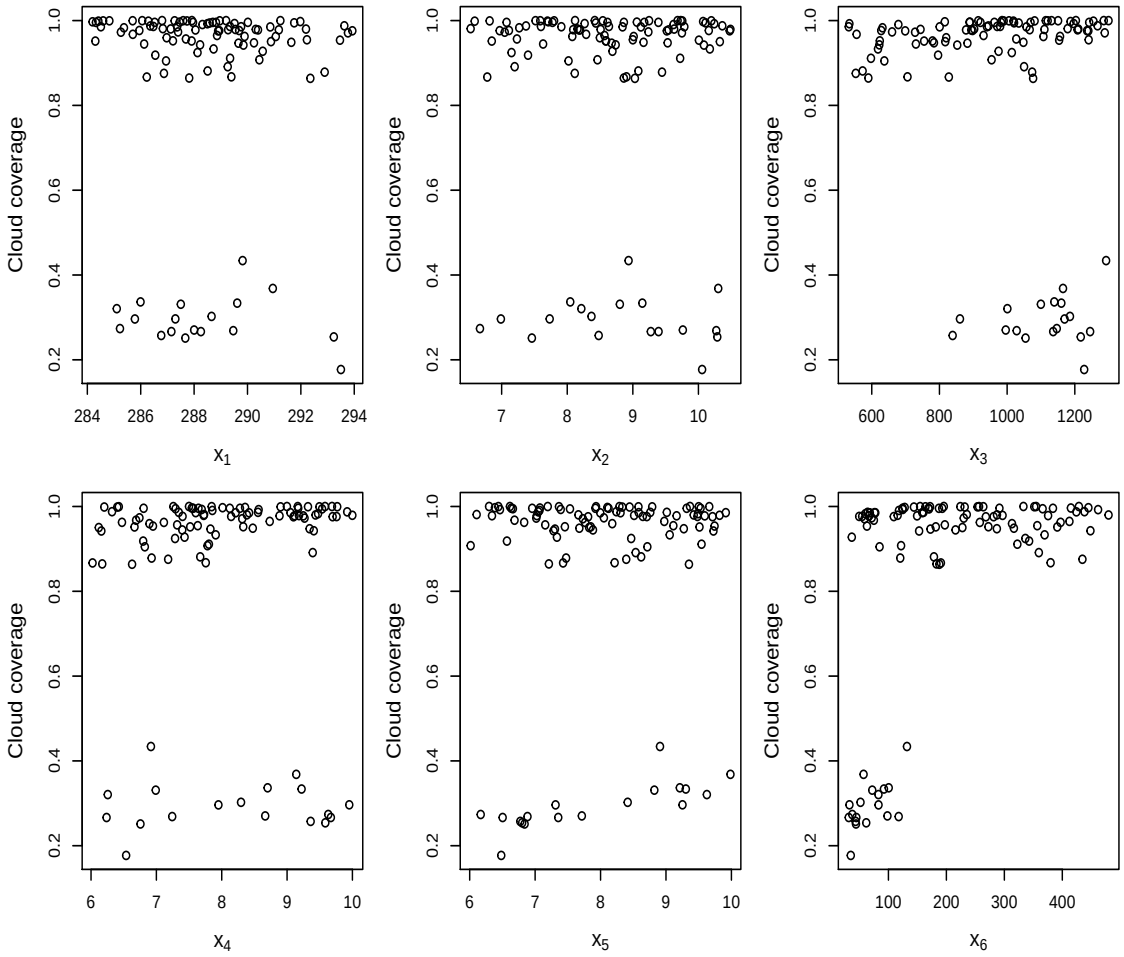


Figure 5.19: The scatterplots of each input plotted against the output for the original 105 cloud coverage data points.

no prior knowledge of this was incorporated.

Number of regions	1	2	3
Probability	0.102	0.667	0.231

Table 5.3: The posterior probability distribution for the number of regions in the cloud model.

To measure how well our method is performing compared to other methods, 35 further simulations (with different input configurations to the training simulations) were run through the computer simulator, and these were used for validation, following the advice of Bastos & O'Hagan (2009). Our method performs better at predicting these validation points (MSE $= 0.016$) than the Treed GP (MSE $= 0.032$) and the standard Gaussian process (MSE $= 0.025$). To gain 1000 thinned samples (using one in every ten samples) after a burn-in of 1000 samples, our method takes around 805 seconds, whilst the most popular alternative, the TGP, takes around 11.6 seconds (for the same number of thinned samples and burn-in). This direct comparison in time, however, is rather misleading as the

methods are implemented in different programming languages (R for our method and Fortran for TGP). Following this, we refitted our model using all 140 simulations as training points, and, due to the modeller's desire to understand more about the parameter location in which the change in regime occurs, the sampler method from Section 5.5 was implemented. An additional 25 parameter combinations were selected by the sampler, chosen using a candidate set of 170,000 points sampled on the boundary of the smaller (low cloud fraction) region, and these simulations were run.

As with our initial model, our final posterior sample (after incorporating the extra information from these new simulations in our training data set) revealed that the MAP model is one which has two regions, and that two regions are most probable $\big(\Pr(r = 2|\mathbf{D}) = 0.87\big)$. An interesting inference we aim to draw from our posterior sample is to visualise the shape of the two regions and the discontinuity between them. A huge challenge when dealing with data of dimension $d > 3$ however is the visualisation of results. Our MAP model has 18 Voronoi cells corresponding to one region and 87 corresponding to the other region. The region with 18 cells corresponds to low cloud fraction output and will be referred to as the smaller region. The region with 87 cells corresponds to high cloud fraction output and will be referred to as the larger region.

In Figure 5.20, we attempt to visualise the shape of the boundary of these regions of cloud behaviour. We used ten equispaced points in each input dimension to create a grid of 1,000,000 equispaced points over the six dimensions, and noted which points lie in each region of the MAP model. To aid with visualisation, we perform a dimension reduction technique. The technique we apply here is a 2-d averaging scheme, which is as follows: There are 15 possible pairwise combinations of our input variables $\big\{(X_1, X_2), (X_1, X_3), \cdots\big\}$, which are each assigned 100 equally spaced points in 2-d. For each of these points, we have 10,000 possible combinations that the other inputs can take (due to our grid) and so we compute the proportion of these 10,000 points that lie within the smaller region. In Figure 5.20, we use a grey scale to represent this proportion, with a white (black) block meaning that all of the points lie within the larger (smaller) of the two regions, and so correspond to areas of high (low) cloud fraction output.

We can see from Figure 5.20 that the smaller (low cloud fraction) region does indeed appear to have a complex shape in the parameter space as we initially suspected. We can observe that most of the parameter space results in the high values of aerosol concentration. It can also be seen that for many of the parameters, we do not notice any notable patterns that help us decide whether a parameter will belong to the high or low cloud fraction group. In particular however, an interesting aspect of this region can be seen in x_6, the aerosol concentration. It appears that smaller values of aerosol concentration are much more likely to be attributed to the smaller region corresponding to low cloud fraction. This observation is supported by the MAP model that was seen when a TGP

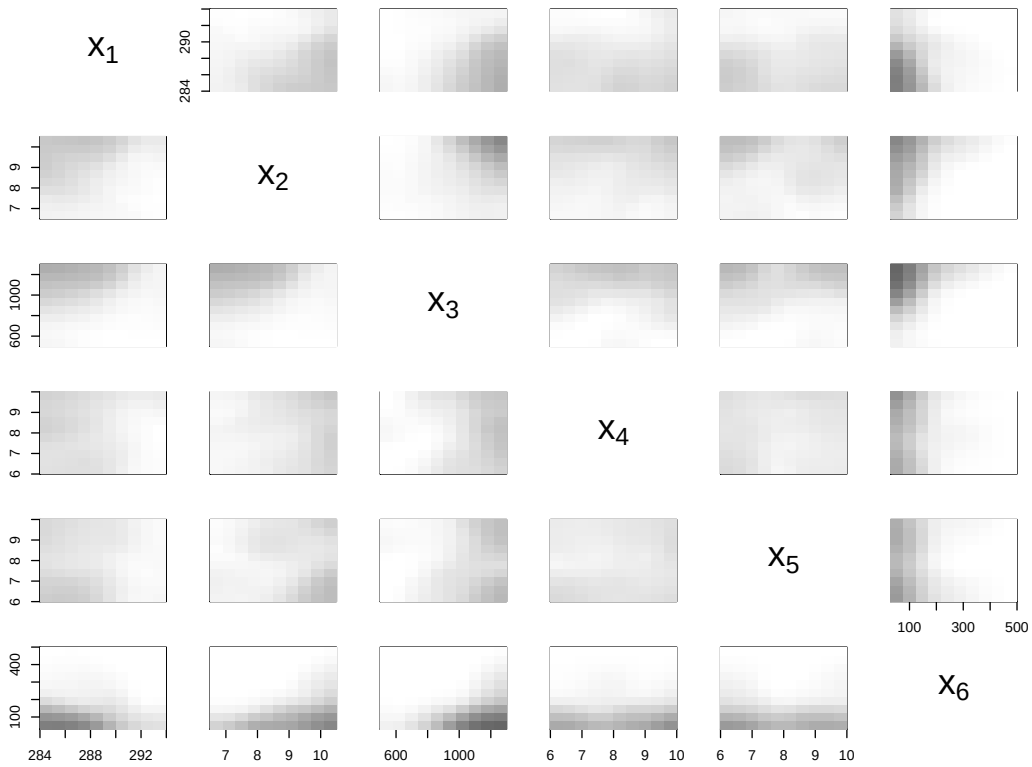


Figure 5.20: The 'averaged' proportion of points that lie within the smaller region for each of the 2-d projections based on the MAP model of all of our points. A white pixel representing that there are no points that lie within the smaller region and a black pixel representing that all of the points lie within the smaller region.

was attempted, with the MAP model in that situation splitting the range of the aerosol concentration input variable at 117.7 cm^{-3}. Figure 5.20 also indicates a reason why the TGP performed poorly compared to our method; in the $(X_1, X_6), (X_2, X_6)$ and (X_3, X_6) projections, we see that the region appears to have a curved boundary, which the TGP will find near impossible to model with straight lines.

Our results here show that by using our described modelling approach, we are able to more clearly and accurately capture and represent the discontinuity that corresponds to the sharp change in cloud behaviour over the six-dimensional parameter space of the cloud model initial conditions. This is a significant step of importance to the cloud modelling community, as this enables the identification of the key initial conditions under which these changes in behaviour may occur. We are also able to determine the sensitivity of the cloud fraction output to the co-varying initial conditions. Full exploration of this may ultimately lead to improvements in the way the shallow cloud coverage is represented in climate models.

5.7.2 USA ammonia levels data

We also apply our method to data on recorded ammonia (NH_4) levels at locations across the USA, obtained from the National Atmospheric Deposition Program (National Atmospheric Deposition Program 2007), which can be seen in Figure 5.21. The NH_4 was measured at 250 locations of interest in the USA, with the two points in the bottom right corresponding to the United States Virgin Islands and Puerto Rico, which are included within the analysis. On plotting the data in Figure 5.21, we have found that there is a drastic change in the output for certain areas of the USA. In other locations, however, the output does not change as drastically, suggesting that we may have heterogeneity. As this is real observed data which is observed with error, as opposed to a deterministic computer output as in the previous examples within the chapter, the error term (or nugget effect) σ_ϵ^2 is included in the covariance function our model.

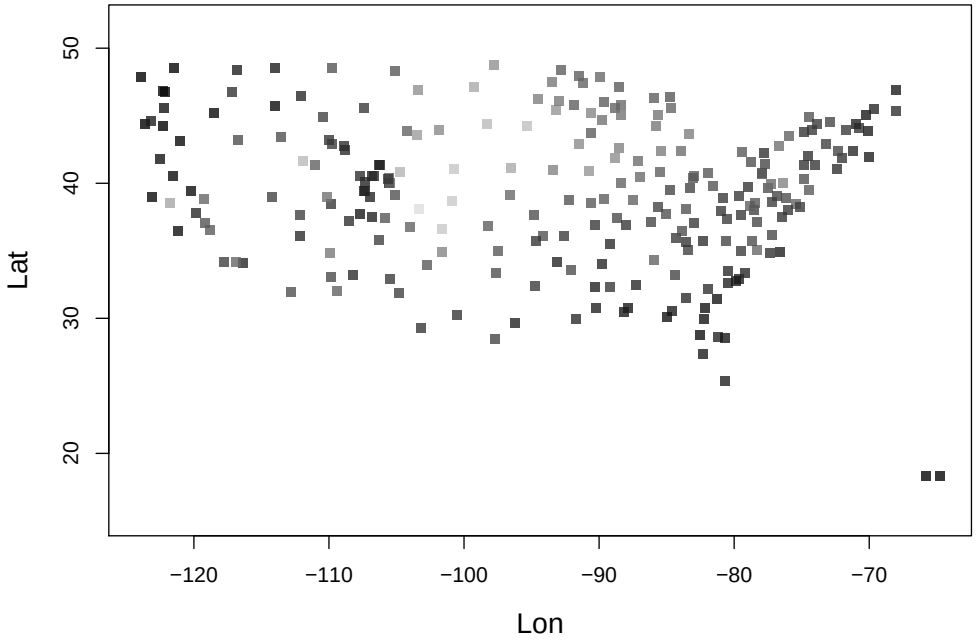


Figure 5.21: The design points and output of the USA ammonia data.

The integrated surface that we obtain for this example, via application of our modelling approach, is shown in Figure 5.22. This surface suggests that the north-central region of the USA has higher levels of ammonia compared to the rest of the country. We also notice that the north western region of the USA has much lower levels than the rest of the country. We can see that the sharp change that occurs around the central part of the USA is captured by our method's integrated surface. Figure 5.23 shows our posterior distribution for the number of regions of different behaviour in NH_4 over the USA. We see that we have a bell shaped distribution that peaks at eight regions with an elongated

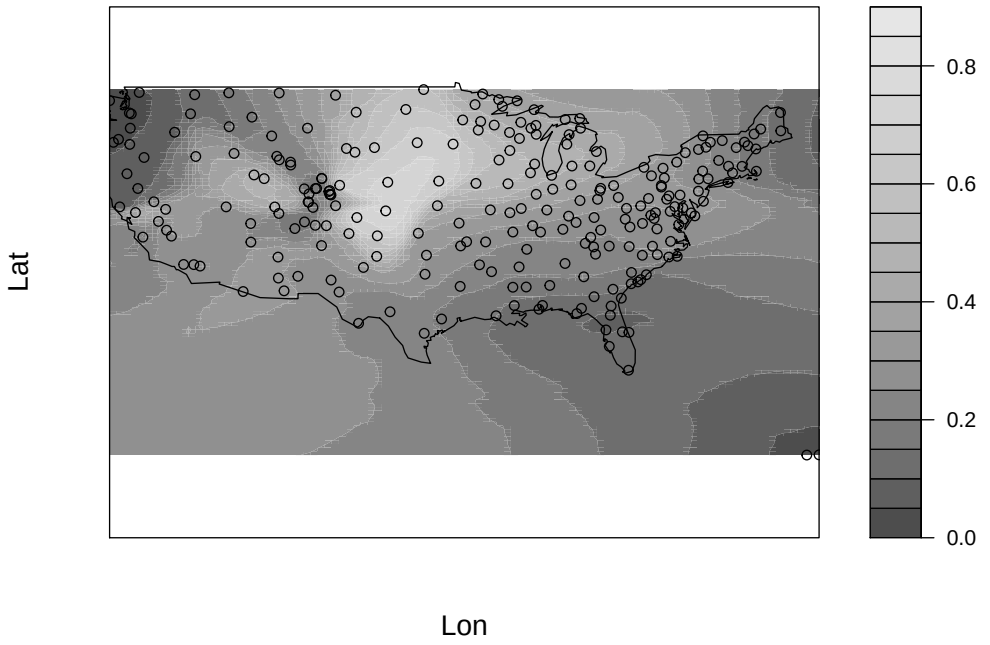


Figure 5.22: The integrated surface of our method for the USA ammonia data.

tail towards the larger values, showing that there are most likely eight different regimes over the spatial area. As with the previous examples in the chapter, we test our method against the TGP and the standard Gaussian process modelling approaches. To do this, we use cross validation in which we randomly omit 50 training points (20% of the total data), and then use these as validation points on a model trained using only the remaining training points. Here, we again found that our method has a lower MSE (MSE $= 0.0057$) than both the standard Gaussian process (MSE $= 0.0084$) and the treed Gaussian process (MSE $= 0.0059$). The integrated surface of the cross-validation data can be seen in Figure 5.25. We can see that the integrated surface for the cross-validation data is very similar to that of the full collection of points, seen in Figure 5.22, which aligns with the MSE results that we have seen for the cross validation.

In Figure 5.24, the approximate MAP tessellation can be seen using all of the USA data. In the MAP model, it can be seen that the input space has been partitioned into six regions. Very noticeable is the north of the USA, which, as we can see from Figures 5.22 and 5.21, is the area in which we see a sharp change in the output, is partitioned into multiple different regions. This reflects the intuition that we would have from the data regarding this area. We can also see that towards the east of America, where we see data that is very similar to each other and low NH4 readings, that this large area is mainly modelled using one Gaussian process.

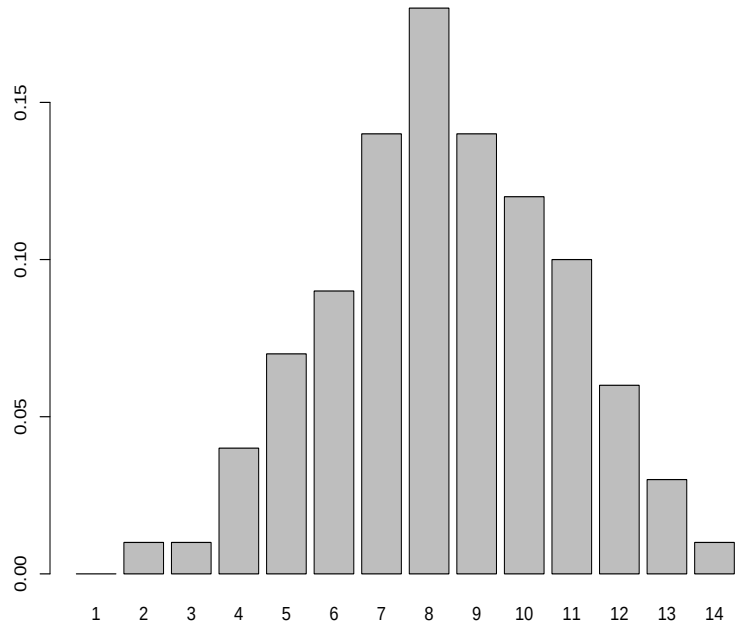


Figure 5.23: The posterior distribution for the number of regions for the USA ammonia level data, in which darker shades represent outputs close to 0 and lighter shades are outputs close to 1.

5.8 Conclusions

In this chapter, a new method was developed that allows us to model an unknown multi-dimensional function f which contains discontinuities. Further to this, a sampling scheme was introduced that selected new design point locations, with the objective of better defining the location of the discontinuity. This was done by sampling at our 'best guess' of the boundary of the discontinuity. It was shown through simulations and practical examples that our model performed better than some of the popular alternative models for these situations in terms of prediction.

As was seen, our model performs particularly well when we had a moderate number of dimensions for our input. The idea of joining together tiles in our Voronoi tessellation allowed us to model the unknown function f well when the shape of the discontinuity was a non-convex shape. Of course, our choice of partitioning method for the input space was the use of Voronoi tessellations. It is easy to see that other partition methods, such as treed partitioning could potentially be improved for situations in which the shape of the discontinuity is non-convex by allowing the any of the partitions that it creates to merge and form larger regions. In our method, a Gaussian process was used as the model for each of the resulting regions. We are, however, free to choose any regression model for the regions; the Gaussian process was selected in our method as the unknown function was was assumed to be smooth in the separate regions and we did not expect to have a

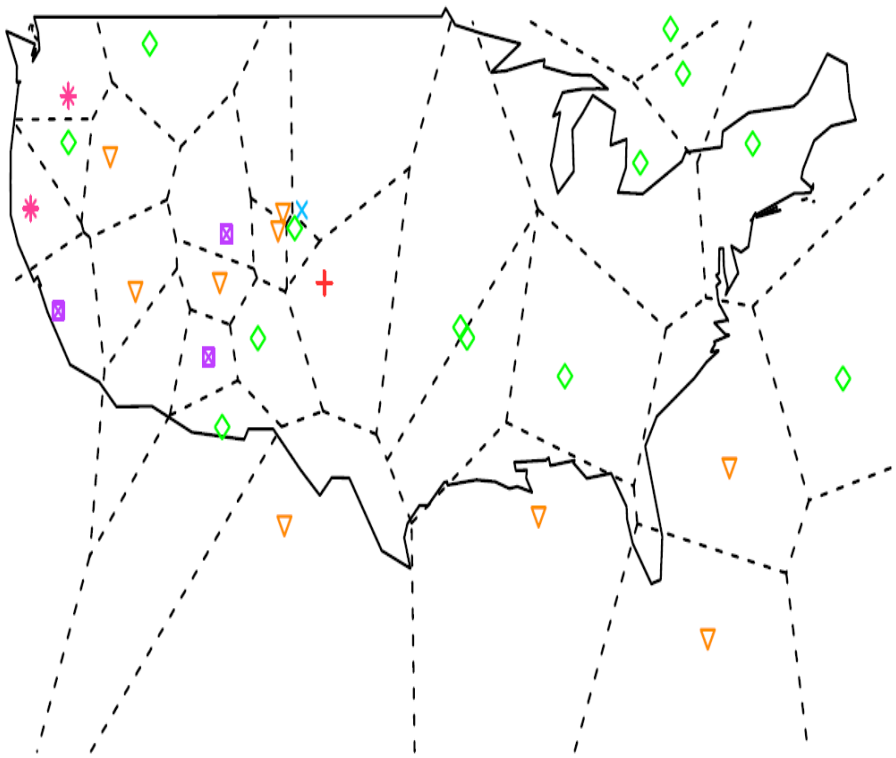


Figure 5.24: The tessellation for the maximum a posterei model for our method using all of the USA data. Tiles that have matching colours and shapes for their centre belong to the same region.

large amount of data in each region.

For situations in which we do have a large number of design points, a different model may be more appropriate due to the problem that a Gaussian process when we have large amount of data. To use the Gaussian process, we must build the covariance matrix A, with the storing and inversion of this matrix being difficult if we have a large number of data points.

In the next chapter, we look at situations in which we have a large number of data-points and the unknown one-dimensional parameter function contains a discontinuity. In particular, emphasis in the next chapter is placed on accurately representing our uncertainty in the unknown function f. This is achieved through a combination of the Gaussian process and wavelet shrinkage.

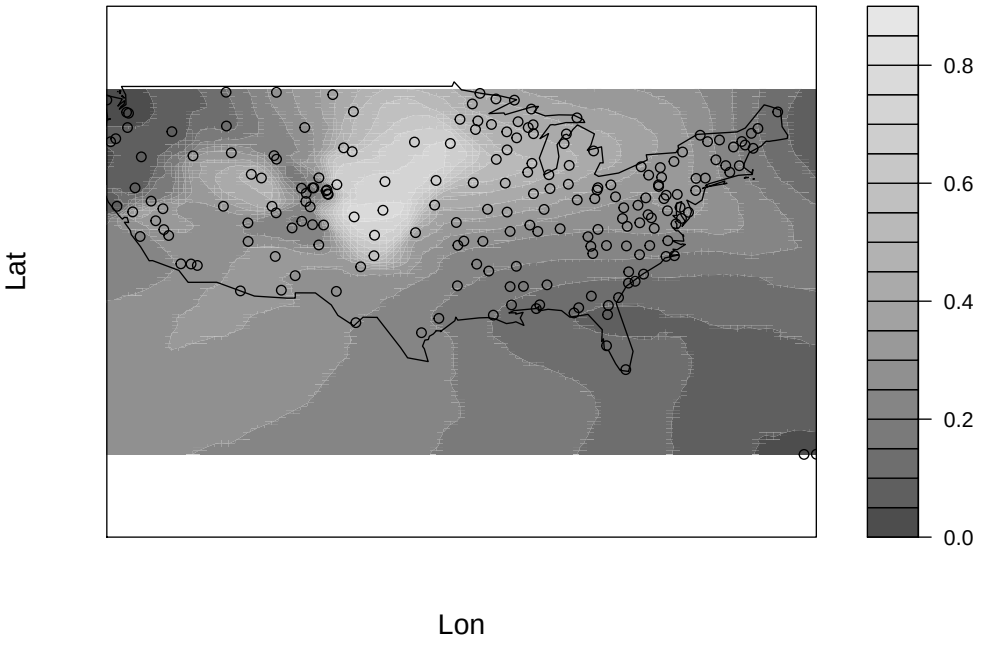


Figure 5.25: The integrated surface of our method for the points that were used for the cross validation in the USA ammonia data.

Chapter 6

Uncertainty quantification in wavelet shrinkage

6.1 Introduction

Often, we are faced with a situation in which we are interested in estimating a function g. In many of these situations, we are able to make observations, y, of the function, such that we have

$$y(x) = g(x) + e(x), \tag{6.1}$$

where e is a distribution with mean zero. In most practical cases, we do not know the true value of e at any location x, nor do we know the true values of the parameters of its distribution (or even its distribution). Hence, we must use some kind of modelling technique to estimate the function g, using the information provided by the y values that we observe.

There is an array of parametric and non-parametric techniques that can be used to model the function g, such as the Gaussian process discussed in Chapter 2, smoothing splines (De Boor et al. 1978), or, simply, linear regression. Many of these methods, however, rely on a number of assumptions being made about the properties of the function g, and the noise element of the observations, e. These assumptions are known to have a large affect on our inference of the function g, limiting our range of possible functions (Heaton & Silverman 2008). For example, if we were to use simple linear regression, we know that our function must have the same gradient for all of the parameter space, and, so, we have ruled out all possible functions that do not have this property. It seems natural to want to limit the amount of assumptions that we make when we are modelling our function, and rule out as little functions as possible to allow our model to be adaptable to different types of function. This is where the wavelet methodology, seen in Chapter 3, is advantageous over other methods, in that we do not make any assumptions, and are therefore able to model functions that contain phenomena such as discontinuities well (Heaton & Silver-

man 2008). Using simply a Gaussian process, for example, in this situation in which we have a jump discontinuity, leads to a poor estimation of the function f, as we can show in Figure 6.1. We can see that not only do we have a poor estimation immediately around the location of the discontinuity, but also the existence of a discontinuity can lead to poor estimates to points far from the location of the discontinuity (due to Gibbs phenomenon). To be able to use wavelet methods, we again need data that is one-dimensional, dyadic, and equally spaced.

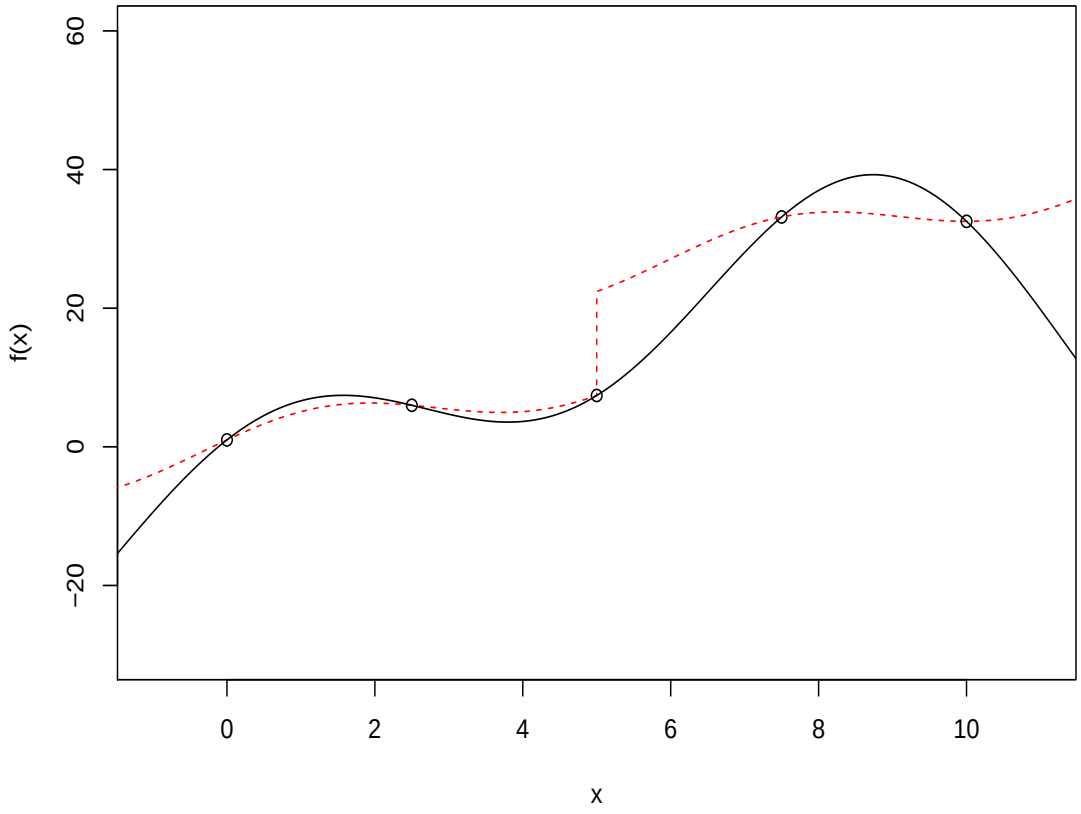


Figure 6.1: The mean of a Gaussian process with a Gaussian covariance function fitted to a function with a discontinuity

Using Bayesian modelling methods, we can estimate the function g, and we are also able to find a full probabilistic distribution that encapsulates our beliefs about this function. This information allows us to explore our knowledge of the function after we have observed it at selected locations, which can be used in decision processes that are based on our beliefs, and guide future explorations. Wavelet shrinkage, as briefly described in Section 3.8, is a modelling tool that can be used to find a posterior distribution for the function that we are trying to estimate. Many papers have been written on this methodology, such as the introductory paper, Donoho & Johnstone (1994), and the introduction

of Bayesian methods to the idea in Chipman et al. (1997). These papers provide us with scenarios in which these methods are useful, mainly, when we have functions that are suited to the assumptions that are made by the methods. That is, when we are using our method on function that are periodic or symmetric. The suited functions that we are referring to are those types of functions that are used within the named papers that do not pose us additional problems, such as issues with boundary conditions. In those papers, we can see that, when we do not have to consider any boundary condition problems, such as wrongly assuming the function is periodic or symmetric, the method seems to perform well. We know, however, that these ideal situations do not always arise, certainly in real world applications, and we see functions that do not have these properties. One main feature that ensures that we do not have any of these nice properties, is the existence of a long term trend in the function.

Papers have been written to try to tackle the problem of the existence of long term trends in the underlying function when attempting to use wavelet shrinkage. Three notable papers are Lee & Oh (2004), Oh & Lee (2005), and Oh & Kim (2008), in which they try to tackle this problem using a linear model to attempt to remove the long term trend, before performing Bayesian shrinkage methods. Lee & Oh (2004) introduces the idea of the method, suggesting selection of the parameters of the linear model using methods such as Stein's unbiased risk (Stein 1981) and Bayesian information criterion (e.g. Weakliem 1999). Oh & Lee (2005) suggests a local linear regression to improve the linear model if we have a trend that is difficult to model towards the boundaries. Oh & Kim (2008) discusses the advantage that is gained when different methods are used to select the parameters of the linear model, such as using the integrated likelihood of the model parameters or empirical Bayes methods, in which we select the hyper-parameters for the prior distributions of our model by using the data we have observed (hence breaking the Bayes paradigm). These papers use a point estimate to report their results, often reporting the median of the shrinkage method, found by selecting the median of the posterior distribution of the wavelet coefficients and using the estimate of the long term trend to report their estimate of the function.

As discussed previously, it is often useful to report distributions, as opposed to single point estimates, as it can be much more informative when considering our uncertainty in the estimation of the function. In this chapter, we attempt to improve on the existing methods by, not only looking at the posterior distribution of the function that is created from the wavelet shrinkage part of the method, but by also considering the uncertainty that we have in estimating the long term trend within the function, or the parameters of the estimate of our long term trend. We also consider the improvement that can be made to the estimation of our uncertainty, and the prediction, through the use of a more complex function for the long term trend, with a Gaussian process our tool of choice.

6.2 General setup of the method

If we have a function, such as that in equation (6.1), and we observe data from this function, we can consider the model in its most basic form, in which e is independent Gaussian noise. Using W as our matrix representation of the discrete wavelet transformation of choice (from Chapter 3), along with equispaced and dyadic realisations of this function, we can see that we have

$$d^* = d + \epsilon,$$

where $d^* = Wy$, $d = Wg$, and $\epsilon = We$.

We know from Chapter 3 that the objective of wavelet shrinkage is to remove the noise e that we observe with the observations to find an estimate of the true function. It is known that the wavelet decomposition provides a sparse representation of data, and, hence, most of the information/features of the function g are encapsulated in a small number of the d coefficients (Nason 2010). We therefore want to use this feature and work within the wavelet domain to find our estimate of the function g. Heuristically, we do this by removing those d^* coefficients that are simply noise, which should have small values, and retain those d^* that are large and attributed to those coefficients that contain the main features of the function g. If we call our estimate of d, after attempting to remove ϵ from d^*, $\hat{d}$, then our estimate of the function g is therefore $\hat{g} = W^T\hat{d}$.

As we have alluded to before, it is crucial that the coefficients are not tainted by boundary conditions. Within the papers mentioned in Section 6.1, for example in Lee & Oh (2004), it can be seen that, when these boundary conditions are present due to the existence of a long term trend, subsequent predictions are poor due to the coefficient on the boundaries very large values. Their solution is to decompose the underlying function into two parts, a long term polynomial function, and a wavelet function. We can set-up this method by again assuming the simplest model, in which we have

$$y_i = f(x_i) + e_i,$$

where e_i is an independent Gaussian distribution with variance σ_e^2. Now, we can decompose the function $f(\cdot)$ so that we have

$$f(x) = f_P(x) + f_W(x),$$

where $f_P(\cdot)$ is a function to model the long term trend, a polynomial function for example, and $f_W(\cdot)$ is the wavelet function. We could also think of f_P and f_W as the global and local fit, respectively. In Lee & Oh (2004), the heuristic reasoning is that we estimate f_P, with $\hat{f}_P$ say, using this estimate to obtain

$$\hat{e}_i = y_i - \hat{f}_P(x_i). \tag{6.2}$$

We then hope that $\hat{e}$ is a function with periodicity or symmetry (at least approximately), and, hence, wavelet smoothing can be performed without the large coefficient(s) that we observe when a long term trend is present. The values of e_i are then used to estimate f_W, with our final estimate of the underlying function

$$\hat{f}(x_i) = \hat{f}_P(x_i) + \hat{f}_W(x_i).$$

The previous papers do not consider the full uncertainty about f, and, instead, use the point estimate that has been described in this section. To gain a full distribution for f, we must consider the uncertainty that we have about both f_P and f_W.

6.2.1 The f_P function

In Lee & Oh (2004) and Oh & Kim (2008), we have

$$f_P = \sum_{l=0}^{d} \alpha_l x^l, \tag{6.3}$$

and so the estimator $\hat{f}_P$ has the form

$$\hat{f}_P = \sum_{l=0}^{d} \hat{\alpha}_l x^l.$$

Lee & Oh (2004) and Oh & Kim (2008) propose numerous method to estimate the various parameters involved in this model. Lee & Oh (2004) uses methods such as Bayesian information criterion (BIC) to decide on the value of d, the number of polynomial terms in f_p, whilst Oh & Kim (2008) introduces methods such as using an integrated likelihood to estimate d. All of the methods in these papers involve using a point estimate $\hat{\alpha}$ when estimating the values of α, which forms our estimate for f_p. Oh & Kim (2008) also introduce a method that takes the uncertainty of d into more consideration than the other methods, placing a discrete uniform prior distribution on d, ranging from one to a user defined value, d_{max}, and using the posterior probabilities $\Pr(d|\mathbf{y})$ to form the estimate

$$\hat{f}_P = \sum_{j=1}^{d_{\max}} \Pr(d = j|\mathbf{y}) \sum_{l=0}^{j} \hat{\alpha}_{jl} x^l,$$

for a chosen value of $d_{\max}$, and $\hat{\alpha}_{jl}$ is the MAP estimate of the lth linear regression coefficient when we include j coefficients in our regression.

To allow us to account for the uncertainties in the estimate of the function, the distributions of the regression coefficients should be utilised. To quantify our uncertainty in this aspect of the function, one solution that we propose is to assign f_P a Gaussian process prior. Using a Gaussian process prior provides a natural method for quantifying the uncertainty in the function and provides a full distribution for it. Alternatively, we could

keep the function form for f_P seen in equation (6.3), but properly take into consideration the uncertainty that we have in the parameters d and α that form our uncertainty in the function. This can be done by placing prior distributions on these parameters. Of course, other functions could be used for f_P, for example splines, however we do not explore these other choices of functions.

6.2.2 The f_W function

As mentioned earlier, we must also consider the contribution that f_W has to our belief in the function f. Previous papers have focused on point estimates, using the posterior median values of d_{jk} to obtain estimates, $\hat{d}_{jk}$, which are then used in the wavelet reconstruction to give us $\hat{f}_W$, our estimate of f_W. We know, however, that a distribution for f_W can be found using these wavelet shrinkage methods, as is seen in Barber et al. (2002), Johnstone & Silverman (2005), and Davison & Mastropietro (2009). Barber et al. (2002) introduces a sampling method to estimate the required univariate posterior cumulative density function (CDF) values of the function f_W for the case in which we use a univariate Gaussian distribution for the non-zero mixture of our wavelet coefficients d_{jk}. Davison & Mastropietro (2009) uses a saddlepoint approximation to find estimates for the CDF values of both the Gaussian case, and the Laplace probability density function (PDF) case, which is used in Johnstone & Silverman (2005).

Our setup for the prior distribution of any coefficient d_{jk} is

$$\pi(d_{jk}) = (1 - \omega_j)\delta(0) + \omega_j \gamma_a(d_{jk}),$$

where $\delta(0)$ is a Dirac delta at zero, $\gamma_{a_j}(\cdot)$ is a symmetric density with level dependent parameter a_j, and ω_j is a level dependent parameter that encodes our belief in the probability that d_{jk} is non-zero. We can see that this mixture distribution encodes the belief that we believe that the coefficient is either exactly zero, or, it is some non-zero value which follows a symmetric distribution. In early papers, the Gaussian density was used as the choice of symmetric density (Abramovich et al. 1998) due to its simplicity. In later papers, however, such as in Johnstone & Silverman (2005), the idea of using a symmetric density with tails that decayed slower than the Gaussian was explored. One density that was proposed was the Laplace density, which has the form

$$\pi(x|a,\mu) = \begin{cases} \frac{1}{2a}\exp\left\{\frac{x-\mu}{a}\right\} & \text{if } x < \mu, \\ \frac{1}{2a}\exp\left\{\frac{\mu-x}{a}\right\} & \text{else.} \end{cases} \tag{6.4}$$

We discuss the derivation for finding the posterior distributions of the wavelet coefficients d_{jk} in the next section.

6.3 Derivation of posterior distributions and the choice of priors

As discussed in the previous section, we are attempting to find the posterior distribution of f_W, and, to do so:

1. We assign prior distributions to our wavelet coefficients.
2. Observe data.
3. Find the posterior distribution of the coefficients given the data.
4. Apply an inverse wavelet transformation to these coefficients.

We can consider the posterior distribution for the case in which we use the Gaussian PDF for γ, the symmetric non-zero part of our prior. For our derivations, we need to define our data and the likelihood that we will use. We observe a function g at n locations $\mathbf{x} = \{x_1, \ldots, x_n\}$, which are dyadic and equally spaced with error to give us

$$\mathbf{y} = \{y_1 = g(x_1) + \epsilon_{y_1},\ \ldots, y_n = g(x_n) + \epsilon_{y_n}\},$$

where $\epsilon_i \sim N(0, \sigma_\epsilon^2)$, and we define $\mathbf{g} = \{g(x_1), \ldots, g(x_n)\}$.

We can then use a wavelet decomposition, W, on the data $\mathbf{y}$ to give us our data $\mathbf{d}^*$

$$\mathbf{d}^* = W\mathbf{y},$$

where $\mathbf{d}^* \sim N_n(W\mathbf{g}, W\Sigma_\epsilon W^T)$, $d_i^* \sim N(W_i\mathbf{g}, W_i\Sigma_\epsilon W_i^T)$, and W_i is the row of the wavelet transformation used to calculate the ith coefficient in $\mathbf{d}$. If we have independence in $\epsilon_i\ \forall\ i \in \{1, \ldots, n\}$, we have $\mathbf{d}^* \sim N_n(W\mathbf{g}, \sigma_\epsilon^2 I_n)$. For ease of notation, we define $\zeta_i^2 = W_i\Sigma_\epsilon W_i^T$, and use the relation that $W\mathbf{g} = \mathbf{d}$. We can then define our likelihood for a wavelet coefficient to be

$$d_{jk}^* | d_{jk} \sim N(d_{jk}, \zeta_i^2). \tag{6.5}$$

Using Bayes rule, we are able to find the posterior distribution of the wavelet coefficient d_{jk}, and the calculations for these posterior distributions can be seen in Sections A.1.1 and A.1.2 for the Gaussian and Laplace PDF respectively.

6.4 Selecting the parameters for our prior distribution

Crucial to this method, we must decide the values of the parameters in our prior distribution. It is easy to consider that the larger our value of ω_j, the stronger our belief is that the true value of d_{jk} is non-zero. We also need to decide on the values of ζ^2, which characterises the level of noise that we observe our wavelet coefficient with, and should

be estimated by Σ_ϵ and a_j, which characterises the speed of convergence in density to zero, and, hence, how likely 'large' coefficient values are.

When we are using a wavelet shrinkage method, we are almost always considering more than one wavelet coefficient. Typically, the more data that we have, the larger number of wavelet coefficients that we consider in our shrinkage. When we have a very large number of these wavelet coefficients, it is infeasible (or at least impractical) to have a representative prior distribution that we have correctly elicited for all of the considered coefficients. This is not only due to the large number of coefficients and parameters that we have to consider, but the lack of substantive prior knowledge on these coefficients, with a lack of research into eliciting due to these problems. To have an informative belief about a specific coefficient at a specific level seems very unlikely without having seen any data, although, of course, if time allows and we have the knowledge, we should incorporate our belief representatively.

With this in mind, a decision still needs to be made on the values of the parameters in our prior distribution. In the recent literature, when a Bayesian shrinkage is utilised, empirical Bayesian methods are utilised. Notably, Abramovich et al. (1998) first advocated a standardised selection of the parameters of our prior distribution, exploring a relationship with the choice of parameters and the resulting properties in Besov space, as well as its regularity properties. Johnstone et al. (2004) then began the exploration into emperical Bayesian techniques, selecting the values of the parameters using a likelihood method that they introduce, which is

$$\ell(\omega_j, a_j; X_i) = \sum_{i=1}^{n} \log\{(1-\omega_j)\phi(X_i) + \omega_j g_{a_j}(X_i)\}, \tag{6.6}$$

where $\phi(\cdot)$ is the PDF of the standard Gaussian distribution, and $g(\cdot)$ is the convolution of the symmetric distribution γ_{a_j} and the Gaussian distribution. The paper suggests selecting ω_j by maximising equation (6.6), using the d_{jk} values as our X_i terms, and replacing i in the equation with k. The value of a_j can also be estimated using a similar method, maximising the maximum likelihood with this parameter included in equation (6.6).

The estimation of the parameter ζ_j^2 can be done in multiple ways. Donoho & Johnstone (1994) suggest that the Maximum Absolute Deviation (MAD) estimator, a non-Bayesian method, should be used to estimate the parameter. The MAD estimator is defined as

$$\sigma_{\text{MAD}}(\mathbf{x}) = \text{med}(|1.4826\mathbf{x} - \text{med}(1.4826\mathbf{x})|), \tag{6.7}$$

where $\text{med}(\cdot)$ is an operator that finds the median of a set of numbers. We can see from equation (6.7) that we are calculating the median absolute distance of the points from the median value. The number 1.4826 is used as a correction such that when the MAD estimator is used on Gaussian data, the standard deviation is estimated unbiasedly (Nason

2010). Their paper advocates the use of it on the finest level coefficients, as this level tends to contain mainly random noise, and then use this estimate for all levels. This is done as attempting to estimate the noise term from subsequent coarser levels is more difficult due to the prevalence of the functions coefficients generally.

Johnstone & Silverman (1997) also advocated the use of the MAD estimator, equation (6.7), but estimating ζ_j^2 by level, using the corresponding level's wavelet coefficients. They argue that, when faced with noise that has a correlation structure, using an estimate based on that level's coefficients provides a better posterior estimate. Other methods for estimating ζ_j^2 have also been used, such as including the parameter in our maximum likelihood approach from equation (6.6) (Peck 2010).

Other decisions must be made when deciding on the parameters and form of our prior, which are: the coarsest level that we will implement the wavelet shrinkage on, and our choice of smooth PDF in our prior distribution. In Section 6.6, we show results for both choices of PDF for comparison. In the literature, there is little written on the choice of coarsest level, often referred to as the primary resolution level, j_0 (Nason 2010); however, it can have a large impact on our results. Barber & Nason (2003) explored the effect that the primary resolution level had on the Absolute Mean Squared Error of different shrinkage methods and concluded that most methods were insensitive to the choice of j_0. The paper, however, did not consider the effect that the choice of j_0 has on the posterior variance of our prediction. Explorations into this suggested that using a j_0 that is very small can artificially inflate the posterior variance, which may happen due to the difficulty in estimating the parameters ω_j, a_j, and ζ_j^2 when we have a very small number of coefficients to do so with.

6.5 Our algorithm

Using the explorations and considerations from the previous sections, we are able to describe the procedure in full and create an algorithm that others can follow to implement our method. In the algorithm, we detail a 'preset' method that we recommend, however, as described in the previous sections, if it is appropriate, prior knowledge should be used to aid choices of the prior and their parameters.

A Gaussian process is used for our prior distribution of f_P. As the data that we will be using will typically be noisy, our Gaussian process will have a noise term in the covariance function. The wavelet function's prior distribution for its coefficients require a number of decisions. The symmetric PDF that is used, and is a recommended choice in practice, is the Laplace PDF, from Section A.1.2. The parameters ω_j and a_j are estimated by maximising the likelihood method in equation (6.6). To deal with the potentially correlated noise, we use the MAD estimate for ζ_j^2, using those observed coefficient in level

j. As a pre-set, $j_0 = 4$ is used, which allows a total number of at least 16 coefficients to be used to calculate our parameters a_j, ω_j, and ζ_j^2.

To sample from the distribution of f, we first find the posterior distribution of $f_P|\mathbf{y}$, which, as we know from Chapter 2, will be a multivariate Gaussian for any finite number of locations. We can make a single posterior functional draw from the distribution $f_P(X)|\mathbf{y}$, and subtract this from the data $\mathbf{y}$, labelling the resulting data $\tilde{e}_i$ (similar to equation (6.2)). We can make a posterior functional draw by making a random pointwise draw from the posterior distribution $f_P(X)|\mathbf{y}$ at every X location. The new data, $\tilde{e}_i$, is then decomposed using our wavelet decomposition, and this data is used to find the posterior distribution of our wavelet coefficients. Making a single posterior draw from each wavelet coefficient, an inverse wavelet transformation is used to find our draw from $f_W(X)|\mathbf{y}, f_P$. The draws from $f_P(X)|\mathbf{y}$ and $f_W(X)|\mathbf{y}, f_P$ are then summed to give us our posterior draw of $f(X)|(y)$. The procedure as a whole is detailed in Algorithm 4.

6.6 Simulated examples

To show the utility of the method that we have developed, it is implemented on a few test functions. We first explore an example, which is similar to some of the examples that we have explored in the previous chapters. We use the method, as well as a few variants of the method, on a function which has an overall trend with a discontinuity. We then use an example that was seen in Lee & Oh (2004), in which a wavelet test function that is popular in the literature, the Doppler function (Donoho & Johnstone 1994), is offset by a polynomial function. Our method is tested using a variety of prior set-ups, using both the Gaussian PDF and the Laplace PDF, as well as exploring the possibility of using a non-decimated wavelet decomposition (Section 3.7) in place of the decimated wavelet decomposition. Also compared, is the use of the wavelet function without the additional long term trend function f_P, the Gaussian process without a wavelet term, the use of a polynomial term for f_P, similar to that seen in Lee & Oh (2004), and the adaptation to the linear model that was suggested in Section 6.2.1 in which we consider the uncertainty in the linear model.

To make these comparisons, three metrics are used. As explained in Section 6.1, the method that we are using will give us a posterior distribution for the function. Measuring how well the function is performing using just a single metric, such as Mean Squared Error (MSE), which measures how well a single point estimate does compared to the true function, is insufficient. Although MSE does provide a useful metric, in that it provides us with a measure of how close our 'best guess' is to the true function, further metrics should be explored that also provides information on how well the other features of the posterior distribution are performing. To do so, we consider two further metrics. One

Algorithm 4 The new uncertainty method

Input: $y_1, \ldots, y_n \quad \in \mathbb{R}^d$ - observed data.

$x_1, \ldots, x_n$ - data locations.

n_s - number of posterior samples.

Output: $g_1(x)|\boldsymbol{y}, \ldots, g_{n_s}(x)|\boldsymbol{y}$ - posterior samples of g

We have our data $y_i = g(x_i) + \epsilon_i$, $\epsilon_i \sim N(0, \sigma_\epsilon^2)$.

Split our function into $g(x) = f_P(x) + f_w(x)$.

Use a Gaussian process prior for $f_P(x)$.

Update in light of $\mathbf{y}$ to get $f_P(\cdot)|\boldsymbol{y} \sim N(\mu^*, \Sigma^*)$ (see Chapter 2).

Decompose f_W using a wavelet decomposition

$$f_w(x) = \sum_{j=0}^{J-1} \sum_{k=0}^{2^j} d_{j,k}\psi_{j,k}.$$

Place at a prior distribution on the $d_{j,k}$, such that we have

$$\pi(d_{j,k}) = w_j\phi(d_{j,k}; 0, \tau_j^2) + (1 - w_j)\delta(0).$$

Estimate $f_w(x)$ whilst also taking into account our uncertainty in $f_P(x)$.

for (i in $1 : n_s$) **do**

Take a posterior functional draw from $f_P(x)|\boldsymbol{y}$, call it $f_{P_i}(x)$.

Find our sample data, which is $\hat{e}_i = \boldsymbol{y} - f_{P_i}(x)$.

Perform a wavelet decomposition to get $\boldsymbol{d}_i^* = W\hat{\mathbf{e}}_i$.

Take a posterior sample from each $d_{jk}|d_{j,k}^*$ (Section A.1.2), call these d_{jk_i}.

Get a posterior sample for $f_{w_i}(x)|\boldsymbol{y} = \sum_{j=0}^{J-1}\sum_{k=0}^{2^j} d_{j,k_i}\psi_{j,k}$.

Our posterior sample is $g_i(x)|\boldsymbol{y} = f_{w_i}(x)|\boldsymbol{y} + f_{P_i}(x)$.

end for

metric that we consider, is the coverage of our method, which is defined as

$$\text{Cv}_\alpha(\mathbf{X}) = \sum_{i=1}^{n} \frac{\mathbb{1}_{C_\alpha}(X_i)}{n},$$

where $\alpha \in (0,1)$, and C_α is a region containing $\alpha\%$ of the distribution of X_i. This metric will describe how accurate the point-wise uncertainty that we have in our posterior distribution matches reality.

Another metric that will be used to assess the posterior distribution is the 'confidence band' of the prediction. For a single set of variables, $\mathbf{X}$, we can observe whether all observations of the variables are contained within their respective $\alpha\%$ range of their distribution. The confidence band is defined as

$$\text{CB}_\alpha(\{\mathbf{X_1}, \ldots, \mathbf{X}_{n_x}\}) = \frac{1}{n_x} \sum_{i=1}^{n_x} \mathbb{1}_{B_\alpha}(\mathbf{X}_i),$$

where n_x is the number of data sets we are using, and B_α is the multidimensional space that encapsulates $\alpha\%$ of the distribution of $\mathbf{X}_i$. This metric again describes the accuracy of our posterior distribution, but, unlike the coverage which looks at each variable in $\mathbf{X}$ individually, looks at the joint distribution of the $\mathbf{X}$. Hence, for this metric, we need the $\alpha\%$ confidence interval to contain the truth for all of the variables in X to produce a value that is not zero. The confidence band looks at multiple data sets and tells us the proportion of those datasets in which all of the observations are contained point-wise within their respective $\alpha\%$ range of their distribution.

6.6.1 A function with a trend and a discontinuity

For this simulation, we use the test function

$$f_1(x) = \begin{cases} 0.02x + 0.0004x^2 & x \leq 70, \\ 0.02x + 0.0004x^2 + 5 & \text{else.} \end{cases}$$

We realise this function at 256 equispaced points between 0 and 100 with error $\epsilon \sim N(0, 0.4)$, such that we observe

$$y(x) = f_1(x) + \epsilon \qquad \epsilon \sim N(0, 0.4).$$

For the analysis, we will create 100 different datasets, which will all have the same x locations. For all of the wavelet fits, we will be using a level dependent noise term. This is done to account for any correlation and dependency in the data; using a level dependent noise term reduces the effect that this correlation has on our estimate of the coefficient parameters. As opposed to calculating the noise term using the finest level coefficients, we use the coefficients that are on the respective level to find the noise term for that level,

hence making it level dependent. In exploratory analysis, it appeared that the wavelet looked more sensible when this was done, as opposed to making the noise term the same for all levels.

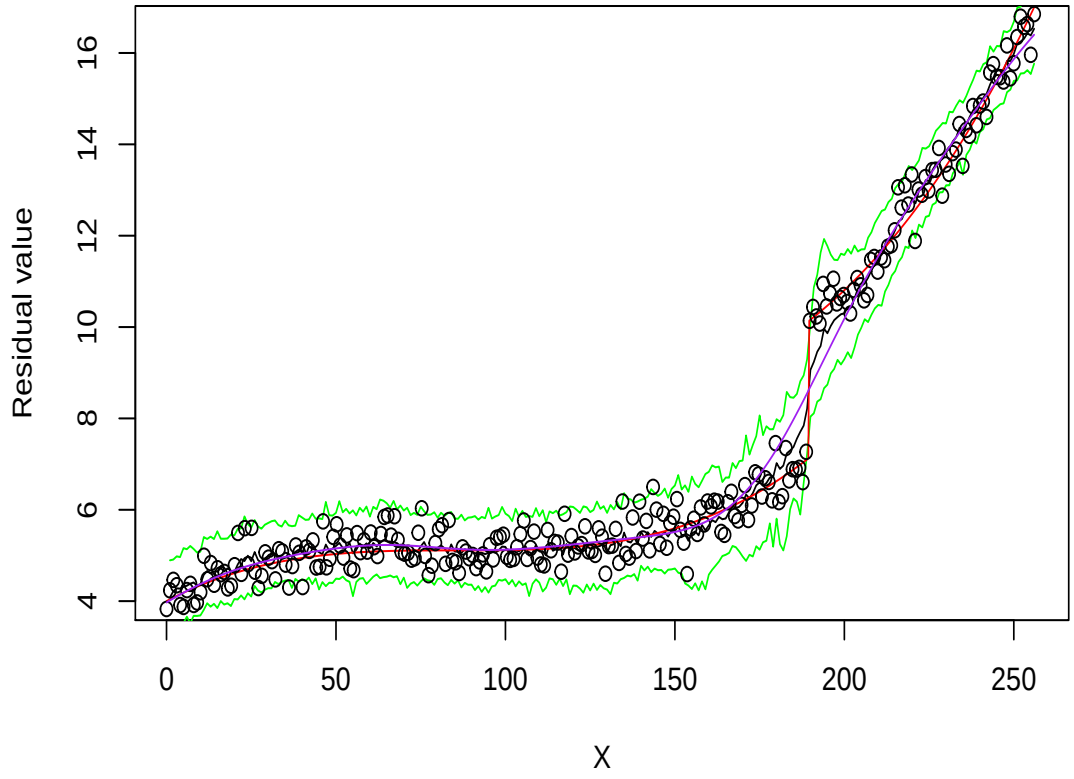


Figure 6.2: An example of a posterior distribution of f from a dataset in Section 6.6.1. We make 256 observations (represented as circles) with a SNR$= 5$, with the true function shown in red, the Gaussian process mean in blue, and our method's mean in black. We also show the point-wise 95% central confidence interval in green.

We can see a table of result in Table 6.1, in which we use different performance measures to test some of our methods against existing methods. We show each possible combination for: the long term trend (Using a Gaussian process, a linear trend and no trend), whether we consider the uncertainty around this long term trend ('u' meaning that we consider the uncertainty and 'n.u' meaning that we do not and simply take its MLE or mean, similar to the methods seen previously in Lee & Oh (2004)), our choice of symmetric prior ('N' denoting Normal and 'L' denoting Laplace), and we also consider which wavelet transformation we use ('D' representing a decimated wavelet transformation and 'ND' representing a non-decimated wavelet transformation).

Also shown, are pictures of both the coverage vs x-location in the subsequent pages.

For comparison, on each page we show the coverage when uncertainty in the long term trend is considered on top of the coverage when it is not considered for each of the methods (Figures 6.3 – 6.10). When we do not have a long term trend, we stack the Normal symmetric PDF on top of the Laplace symmetric PDF for the same type of wavelet decomposition (Figures 6.11 and 6.12). An example of the posterior distribution of f can be seen in Figure 6.2, in which we also show the mean of the Gaussian process to show the visual improvement that can be made through the use of our method when compared to the use of just the Gaussian process.

From Table 6.1, key results have been highlighted with bold numbers, we can see that the MSE when taking into consideration the uncertainty is comparable to that in which we do not take in to consideration the uncertainty. The Non-Decimated and the Decimated wavelet decompositions are comparable for the Gaussian case. We can see that the non-decimated wavelet transformation generally performs better in terms of MSE compared to the decimated transformation in the Laplace case, however, the converse is true when we consider the accuracy of the coverage (a higher coverage with a smaller width of uncertainty) and the confidence bands. In terms of computational time, we can see that, generally, the non-decimated transformation requires more time to implement the method than the decimated transformation. It is also easy to see that the method in which we consider no long term trend adjustment (no f_P) performs poorly compared to our other methods when considering MSE. The same can also be said when we do not attempt to use an f_W function, using only a Gaussian process. For the latter case, we can see that the coverage is unrepresentative. The coverage plots show us that all method performs the poorest around the location of the discontinuity, as would be expected.

6.6.2 The Doppler with a trend

The Doppler test function, that was used in Lee & Oh (2004), has the form

$$f_2(x) = (x(1-x))^{0.5} \sin\left(2\pi \frac{1.05}{(x+0.05)}\right) + 3(x-0.6)^2. \tag{6.8}$$

The original test function was seen in Donoho and Johnstone (1994), which is the first part of equation (6.8). A graphical representation of a wavelet decomposition can be seen in Figure 6.15 for a sample of data from the test function with the linear trend removed. The figure shows the ability of the wavelet decomposition to detect different frequencies at different locations. Lee & Oh (2004) showed how they could deal with the boundary conditions that posed a challenge when the function was not periodic, or in their case, offset by some quadratic term. We can see that the Doppler function has been offset by an addition of $3(x-0.6)^2$, which ensures that we have problems when it comes to these boundary conditions. An example of a dataset from this function can be seen in Figure

Func	WT	Sym	MSE(s.e)	Coverage (width)	Conf band (width)	Mean time
GP(u)	D	N	**0.049** (0.016)	**0.912** (0.614)	0.036 (0.874)	**56.1**
GP(u)	D	L	**0.090** (0.017)	0.982 (1.437)	0.416 (0.832)	122.0
GP(u)	ND	N	0.050 (0.015)	**0.741** (0.362)	0 (0.149)	54.4
GP(u)	ND	L	0.069 (0.014)	0.883 (0.483)	0 (0.373)	**313.2**
GP(n.u)	D	N	**0.049** (0.017)	0.890 (0.580)	0.013 (0.836)	55.4
GP(n.u)	D	L	**0.089** (0.109)	0.971 (1.286)	0.414 (0.829)	127.8
GP(n.u)	ND	N	0.050 (0.014)	0.472 (0.207)	0 (0.147)	54.2
GP(n.u)	ND	L	0.063 (0.009)	0.691 (0.297)	0 (0.178)	345.3
L(u)	D	N	**0.182** (0.020)	0.520 (0.567)	0 (0.985)	52.2
L(u)	D	L	0.063 (0.017)	0.975 (1.311)	0.375 (2.661)	121.8
L(u)	ND	N	0.192 (0.020)	0.372 (0.288)	0 (0.480)	53.2
L(u)	ND	L	0.075 (0.010)	0.912 (0.546)	0 (0.898)	343.7
L(n.u)	D	N	**0.184** (0.020)	**0.475** (0.504)	0 (0.475)	52.1
L(n.u)	D	L	0.071 (0.051)	0.967 (1.251)	0.375 (2.565)	121.5
L(n.u)	ND	N	0.194 (0.025)	0.299 (0.212)	0 (0.398)	53.2
L(n.u)	ND	L	0.062 (0.023)	0.762 (0.333)	0 (0.746)	343.5
/	D	N	**4.324** (0.083)	**0.171** (0.747)	0 (1.428)	49.4
/	D	L	0.153 (0.031)	0.979 (1.916)	0.488 (4.057)	171.3
/	ND	N	2.694 (0.087)	0.134 (0.301)	0 (0.645)	53.5
/	ND	L	0.106 (0.015)	0.905 (0.749)	0 (2.052)	363.3
GP(u)	/	/	**0.114** (0.014)	**0.738** (0.145)	0 (0.147)	2.3

Table 6.1: A table of the results of the various methods for prediction and uncertainty quantification. In the Func column, we have a Gaussian process (GP), a Linear model (Lin), and no underlying model. The brackets represents whether uncertainty in this long term trend function was taken into account. WT represents the type of wavelet transformation used: D – Decimated, ND – Non-decimated. Sym represents the prior PDF used for the symmetric part of our prior on the wavelet coefficients: N – Normal PDF, L – Laplace PDF. The interval width is that of the central 95% credible width.

6.14. Their suggested method was to first try to estimate the underlying long term trend using a polynomial function. As we can see from equation(6.8), they should be able to do this extremely well.

Similar to the analysis performed in their paper, for these methods, 1,000 datasets were simulated, and, similar to the previous section, we looked at three measures to see how well they performed. In Figure 6.14, we can see what the test function looks like, and we can also see the type of data that we are using, with an example set of data shown. For

each of the 1,000 datasets, 512 equally spaced points between 0 and 100 were inputted into the test function to create our output. We can see from the figure an example of the dataset that is used in our analysis; we have simulated these datasets such that the SNR is 5, which is described as a 'medium' SNR in Lee & Oh (2004). The results of this can be seen in Table 6.2. We can see an example of the posterior distribution of f for one of our simulated datasets in Figure 6.13, in which, as expected, the estimation of the function is poor for low values of x, and improves as it increases. It should also be noted that, due to the Gaussian process that we have also used, the long term trend appears to have been found by the posterior distribution.

Func	WT	Sym	MSE(s.e)	Coverage (width)	Conf band (width)
GP(u)	D	N	**0.039** (0.004)	**0.342** (0.184)	0.000 (0.299)
GP(u)	D	L	0.033 (0.013)	**1.000** (1.527)	1.000 (2.891)
GP(u)	ND	N	0.041 (0.004)	0.129 (0.074)	0.000 (0.114)
GP(u)	ND	L	0.035 (0.002)	0.994(1.551)	1.000 (2.958)
GP(n.u)	D	N	**0.039** (0.004)	0.309 (0.167)	0.000 (0.275)
GP(n.u)	D	L	0.037 (0.024)	0.999 (1.535)	1.000 (2.899)
GP(n.u)	ND	N	0.041 (0.004)	0.089 (0.047)	0.000 (0.085)
GP(n.u)	ND	L	0.035 (0.005)	**0.972** (1.254)	0.980 (2.479)
L(u)	D	N	**0.039** (0.001)	0.341 (0.170)	0.000 (0.279)
L(u)	D	L	0.032 (0.014)	1.000 (2.413)	1.000 (3.056)
L(u)	ND	N	0.019 (0.021)	0.112 (0.064)	0.000 (0.047)
L(u)	ND	L	0.033 (0.010)	0.939 (0.721)	0.000 (1.291)
/	D	N	**0.053** (0.010)	0.283 (0.212)	0.000 (0.351)
/	D	L	1.099 (1.485)	1.000 (4.637)	1.000 (7.195)
/	ND	N	0.055 (0.002)	0.079 (0.054)	0.000 (0.092)
/	ND	L	0.048 (0.001)	1.000 (2.221)	1.000 (3.824)

Table 6.2: A table of the results of the various methods for prediction and uncertainty quantification. In the Func column, we have a Gaussian process (GP), a Linear model (Lin), and no underlying model. The brackets represents whether uncertainty in this long term trend function was taken into account. WT represents the type of wavelet transformation used: D – Decimated, ND – Non-decimated. Sym represents the prior PDF used for the symmetric part of our prior on the wavelet coefficients: N – Normal PDF, L – Laplace PDF. The interval width is that of the central 95% credible width.

From Table 6.2, in which some key results have been highlighted in bold, we can see that the normal PDF performs poorly for all of the methods when we consider the accuracy of the coverage and the confidence bands. We can see that all of the methods that use the

f_P term perform best when considering the MSE, with the use of wavelet shrinkage on its own performing poorly on this metric. We can again see that, generally, the Laplace PDF tends to perform better than when a normal PDF is used in terms of MSE. We can also observe that the normal prior performs poorly with respect to coverage and confidence band, with these metrics showing that the normal prior results in overconfidence in our estimation of the function. The Laplace PDF, on the other hand, tends to underestimate our confidence in the estimation of the function and produces confidence intervals that are large. The use of the non-decimated decomposition with a Laplace PDF seems to produce the most accurate results for both coverage and the confidence band.

The coverage plots, seen in Figures 6.17 to 6.24, shows that the Gaussian PDF struggles to correctly encapsulate the uncertainty throughout the input space, showing an oscillation in their coverage plots. The plots also show that the Laplace PDF tends to overestimate our uncertainty, with a large variance responsible for the coverages close to one seen in Figures 6.18, 6.22, and 6.24.

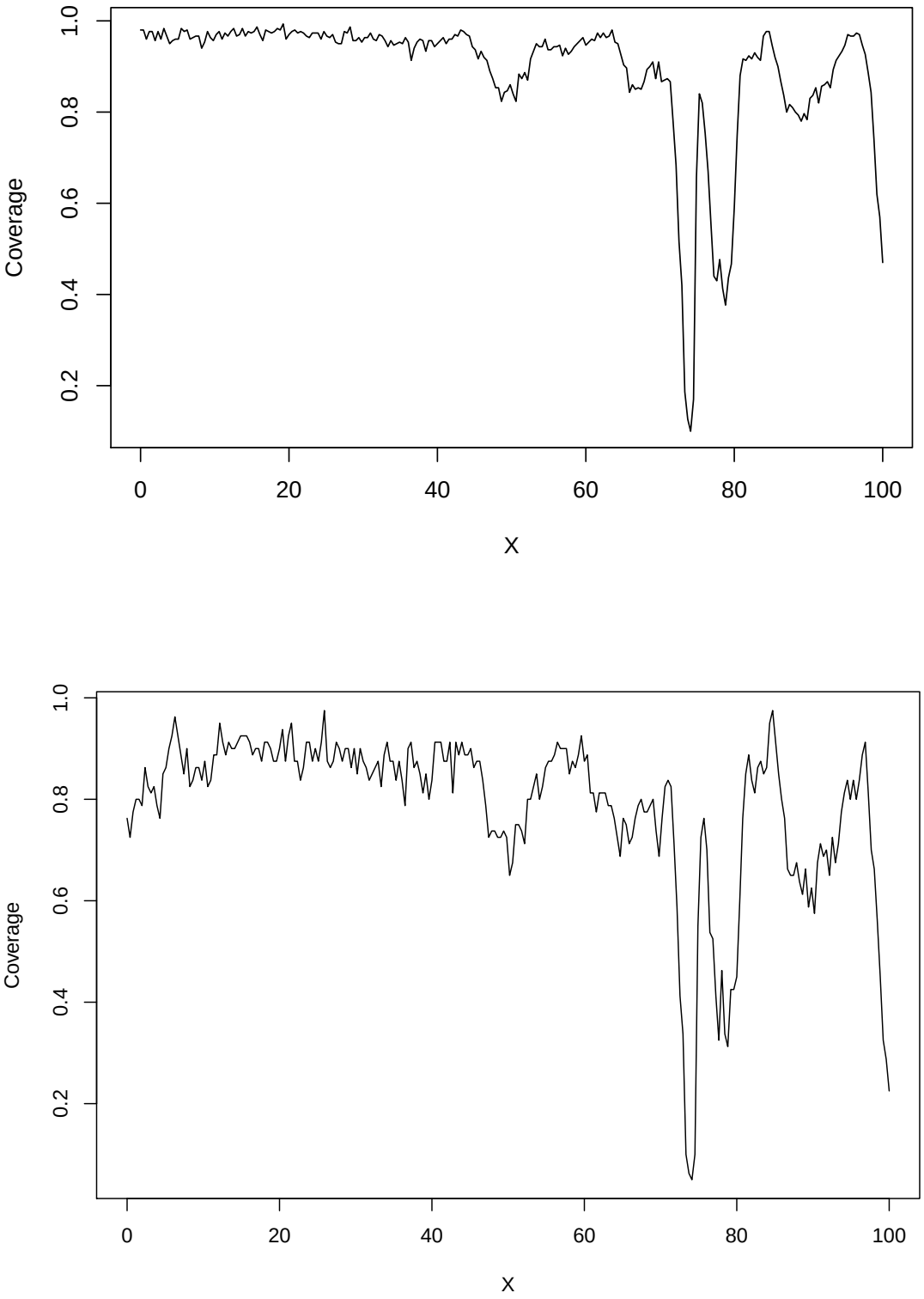


Figure 6.3: Coverage vs x, when we use the GP with uncertainty (top) and just its mean (bottom) as the long term trend; using a decimated wavelet decomposition with the normal PDF as the symmetric PDF in the prior

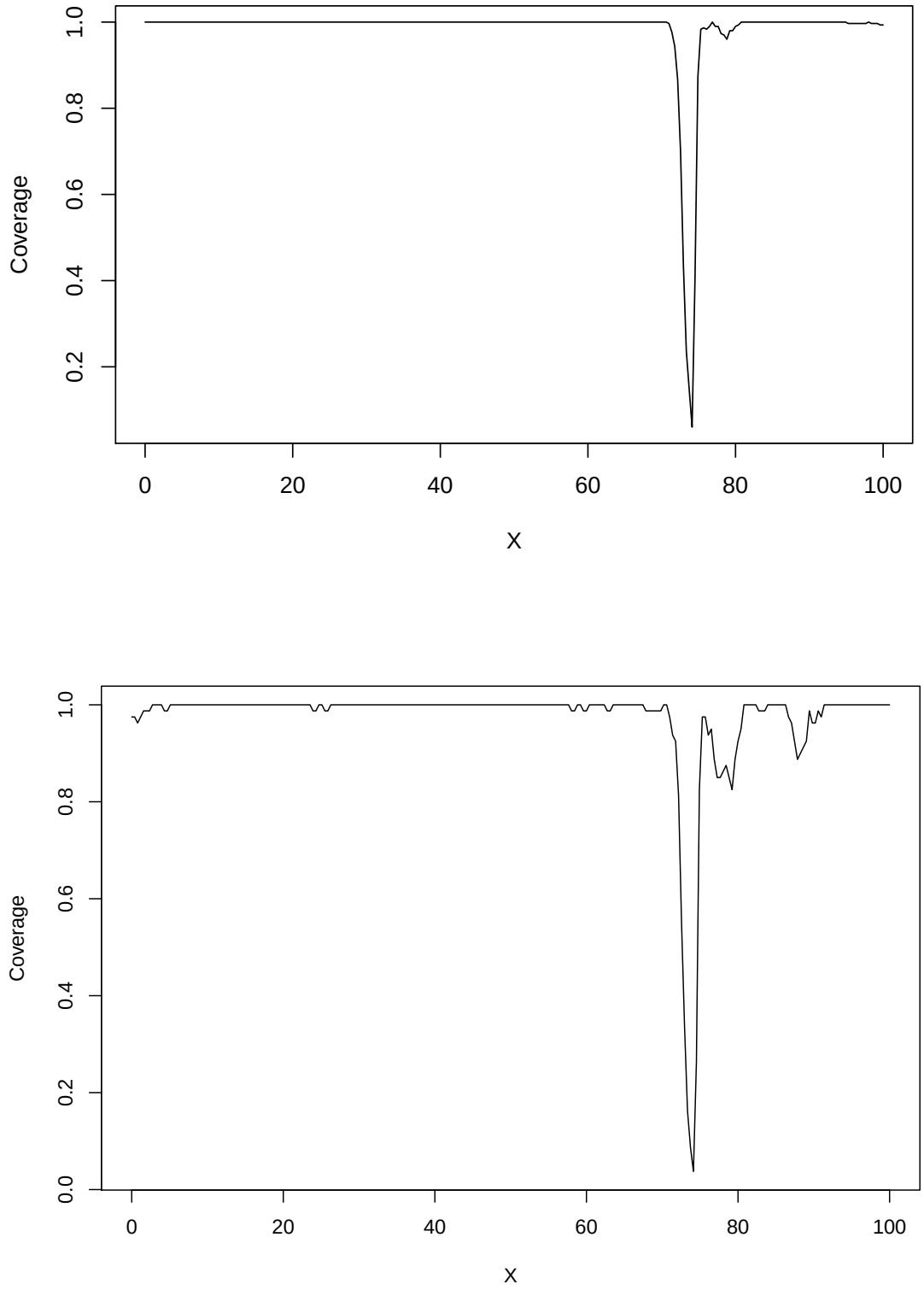


Figure 6.4: Coverage vs x, when we use the GP with uncertainty (top) and just its mean (bottom) as the long term trend; using a decimated wavelet decomposition with the Laplace PDF as the symmetric PDF in the prior

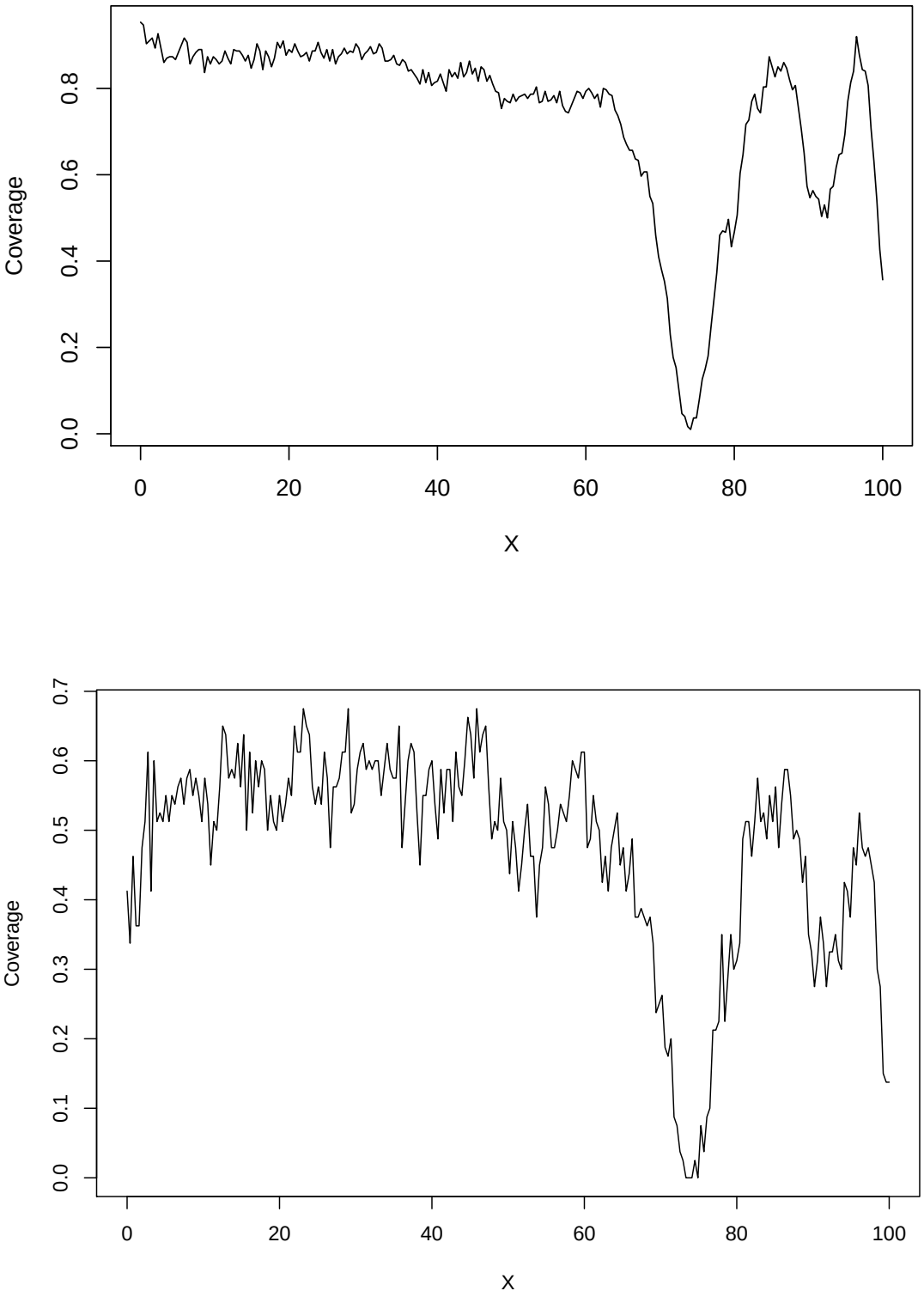


Figure 6.5: Coverage vs x, when we use the GP with uncertainty (top) and just its mean (bottom) as the long term trend; using a non-decimated wavelet decomposition with the Normal PDF as the symmetric PDF in the prior

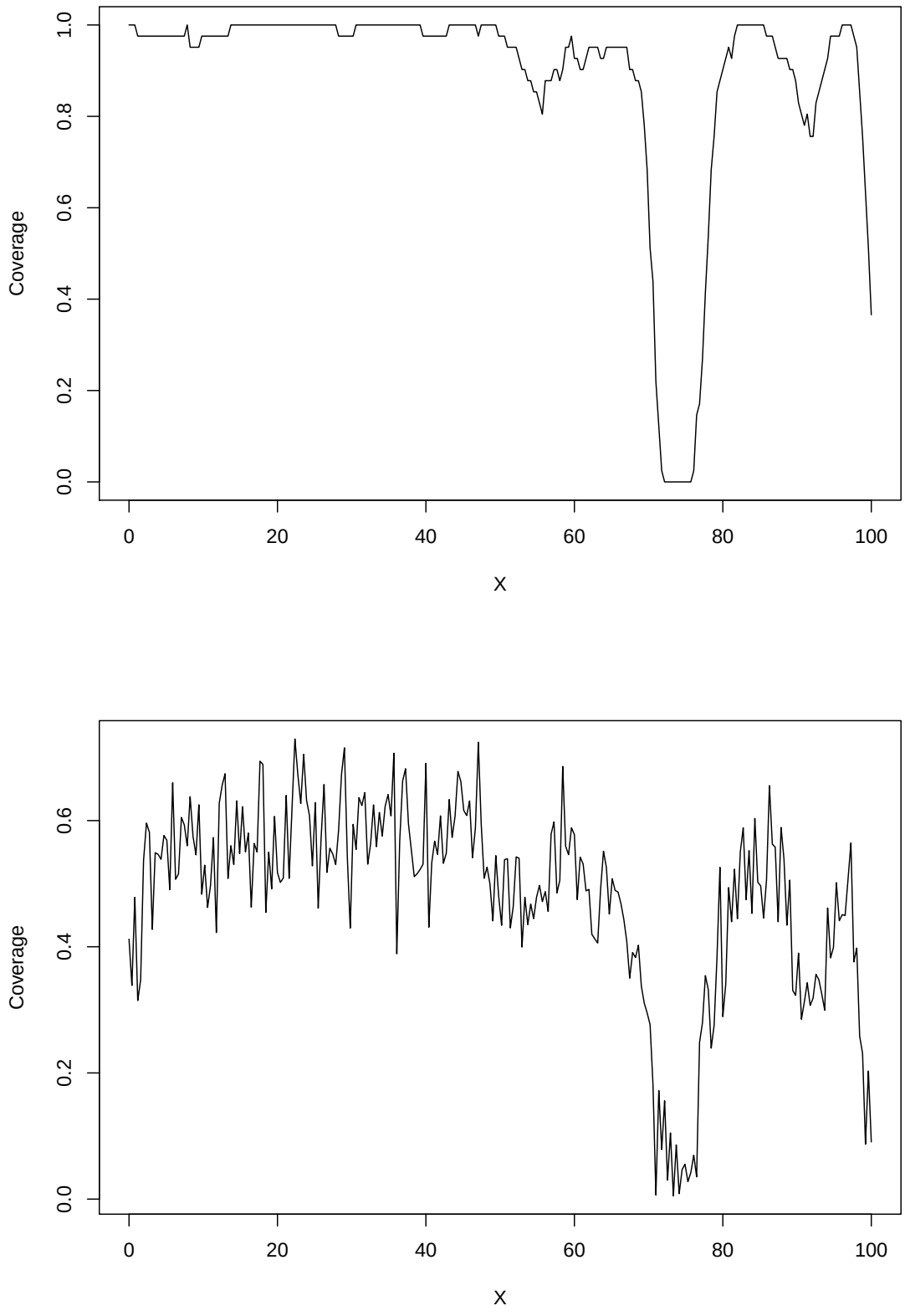


Figure 6.6: Coverage vs x, when we use the GP with uncertainty (top) and just its mean (bottom) as the long term trend; using a non-decimated wavelet decomposition with the Laplace PDF as the symmetric PDF in the prior

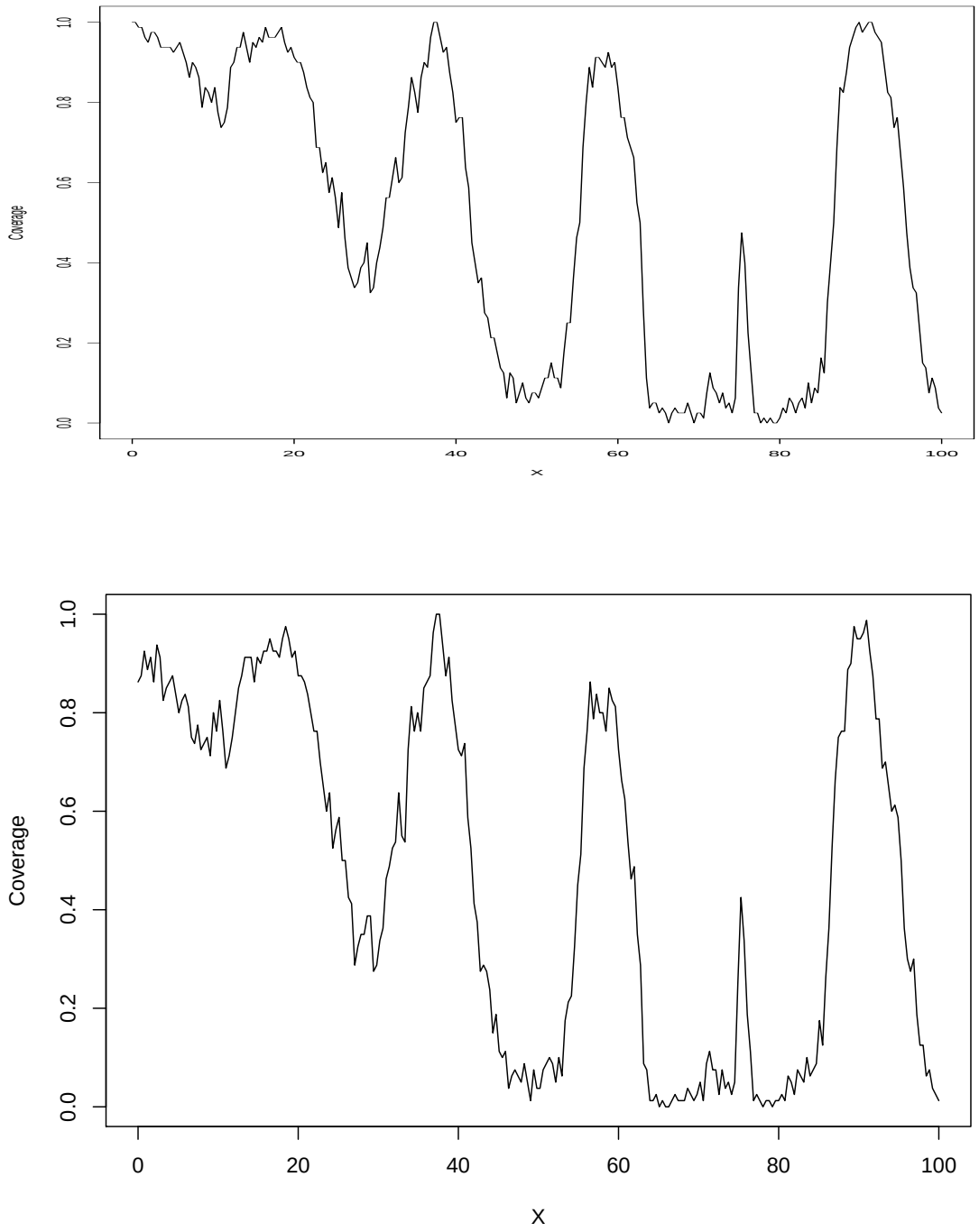


Figure 6.7: Coverage vs x, when we use a linear model with uncertainty (top) and just its mean (bottom) as the long term trend; using a decimated wavelet decomposition with the Normal PDF as the symmetric PDF in the prior

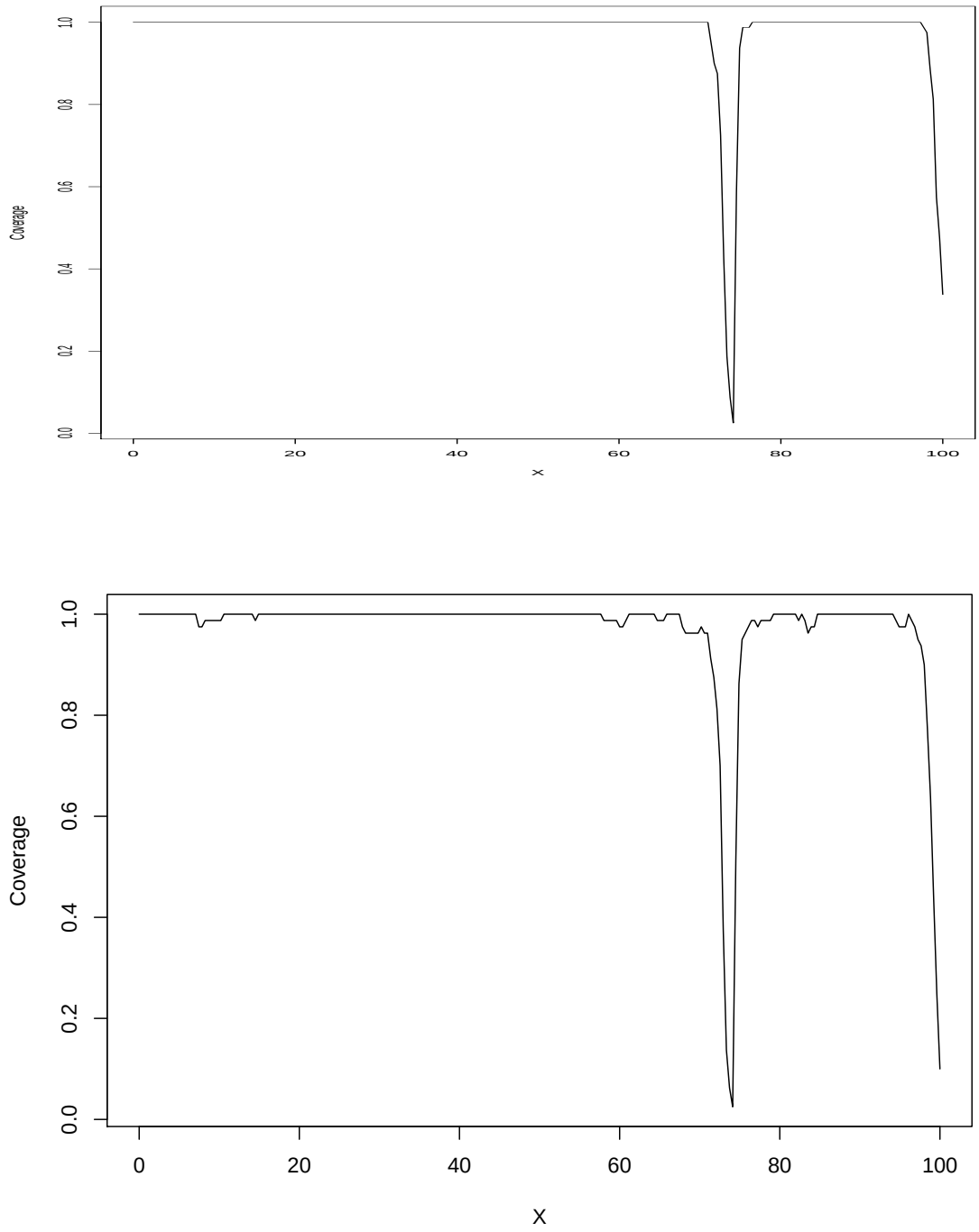


Figure 6.8: Coverage vs x, when we use a linear model with uncertainty (top) and just its mean (bottom) as the long term trend; using a decimated wavelet decomposition with the Laplace PDF as the symmetric PDF in the prior

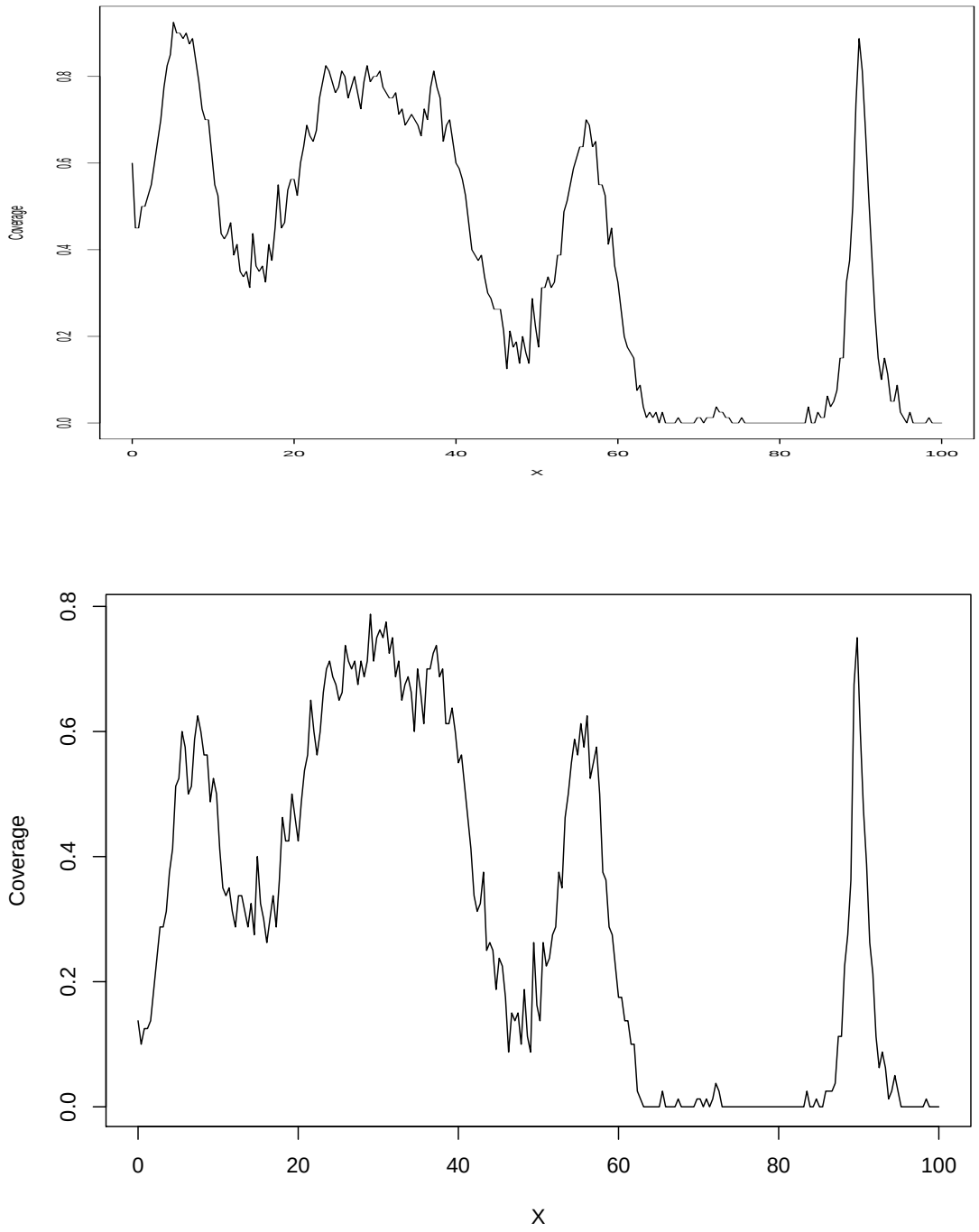


Figure 6.9: Coverage vs x, when we use a linear model with uncertainty (top) and just its mean (bottom) as the long term trend; using a non-decimated wavelet decomposition with the Normal PDF as the symmetric PDF in the prior

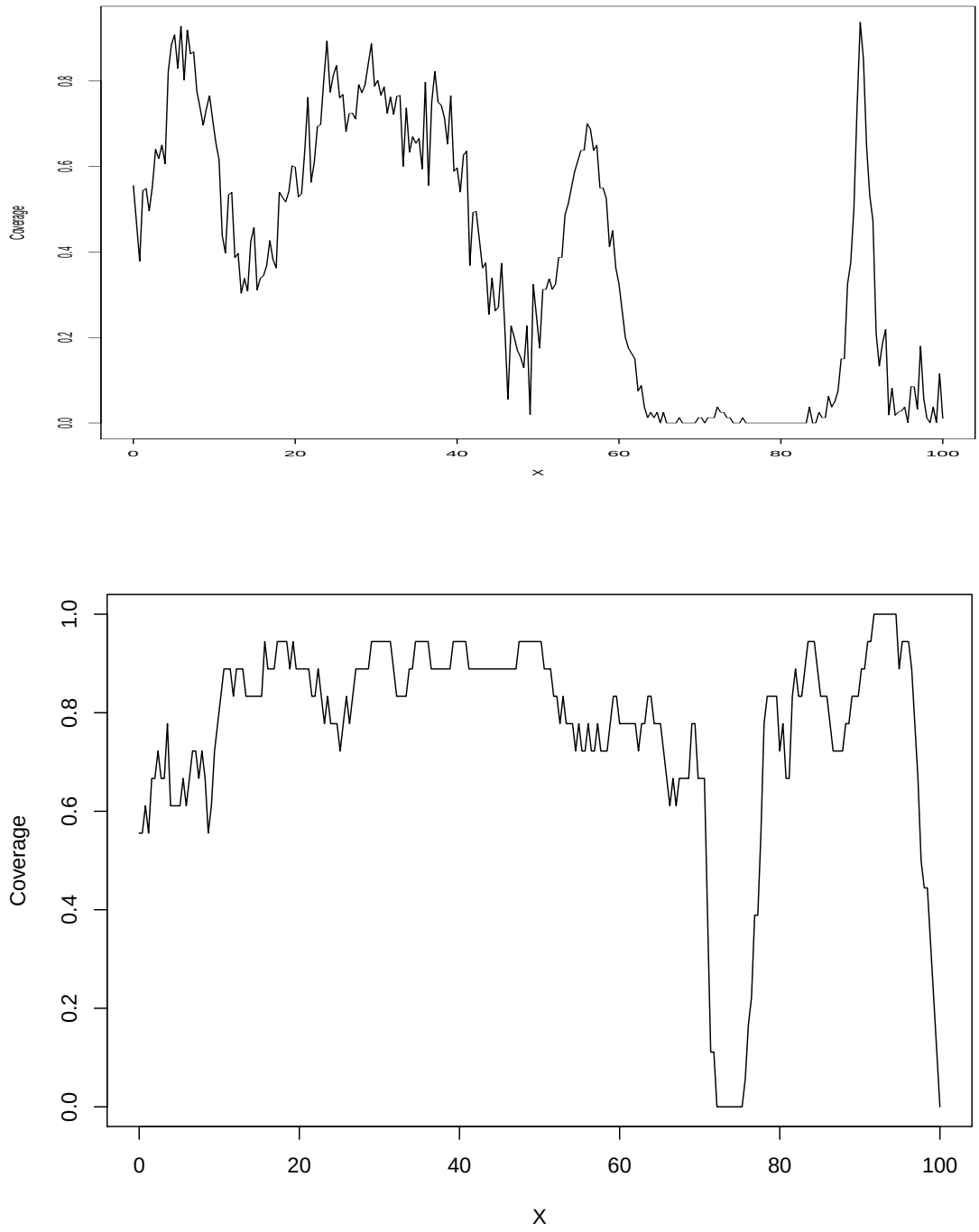


Figure 6.10: Coverage vs x, when we use a linear model with uncertainty (top) and just its mean (bottom) as the long term trend; using a non-decimated wavelet decomposition with the Laplace PDF as the symmetric PDF in the prior

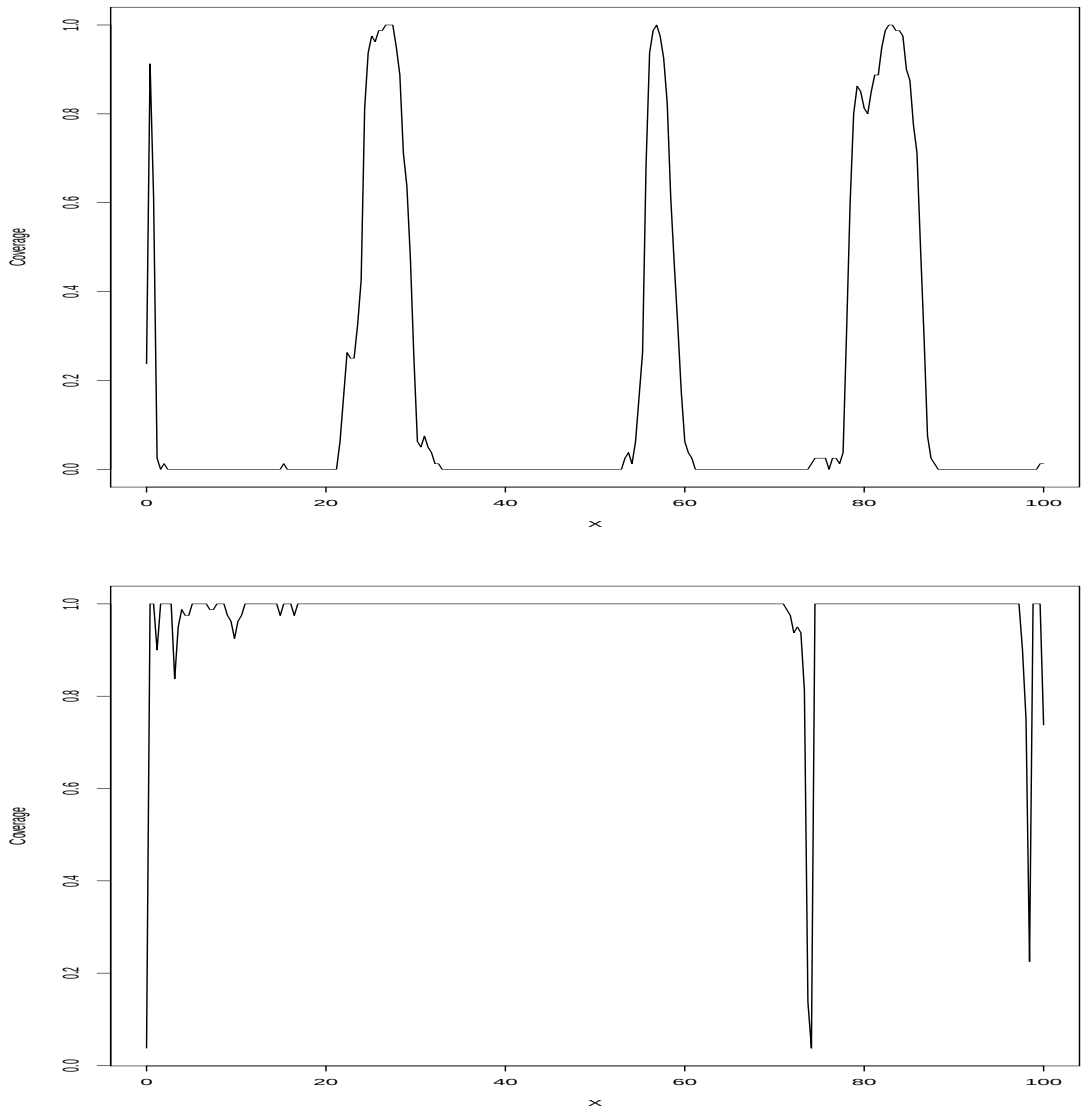


Figure 6.11: Coverage vs x, no long term trend is used here; using a decimated wavelet decomposition with the Normal(top) PDF and Laplace(bottom) PDF as the symmetric PDF in the prior

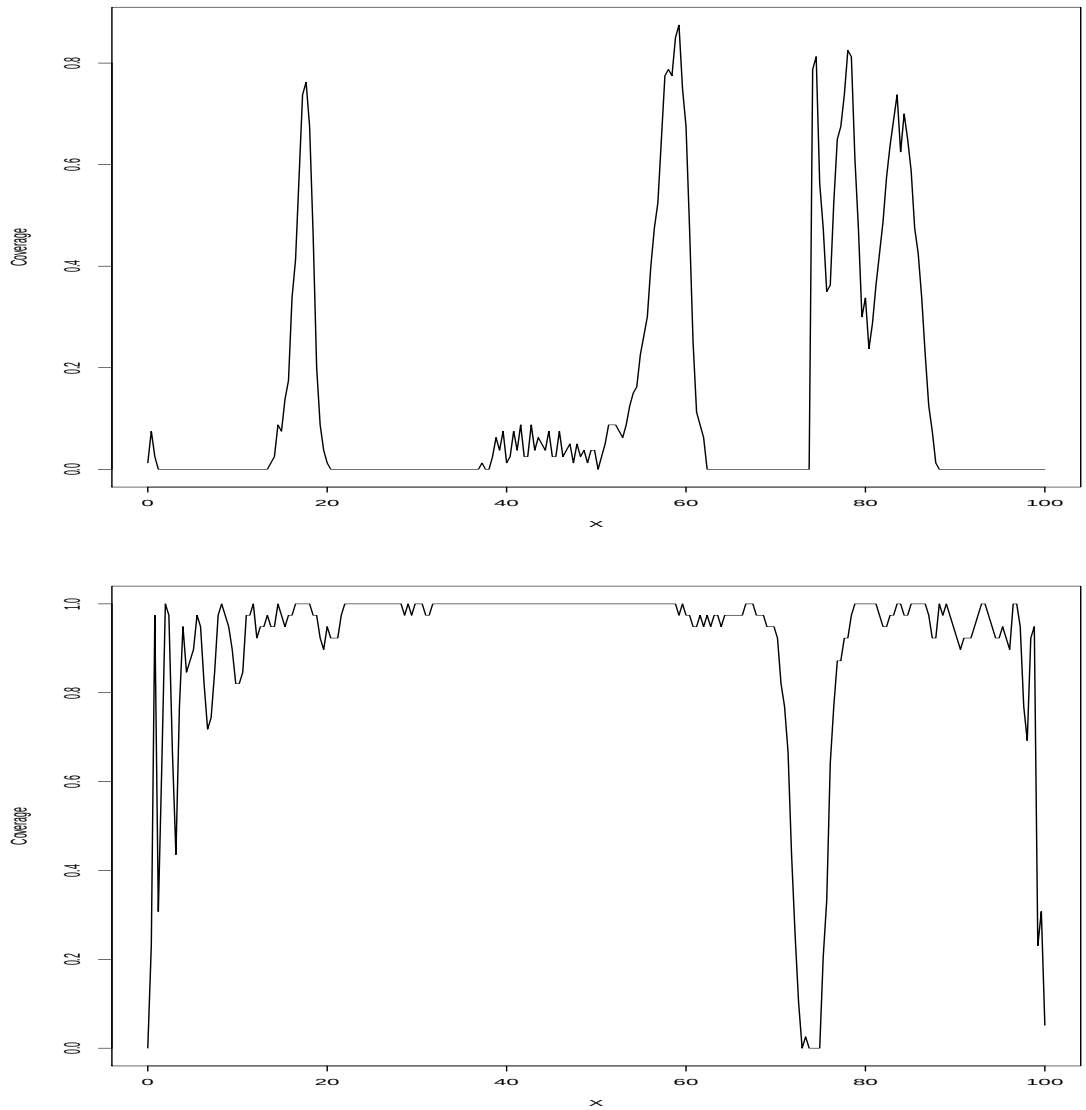


Figure 6.12: Coverage vs x, no long term trend is used here; using a non-decimated wavelet decomposition with the Normal(top) PDF and Laplace(bottom) PDF as the symmetric PDF in the prior

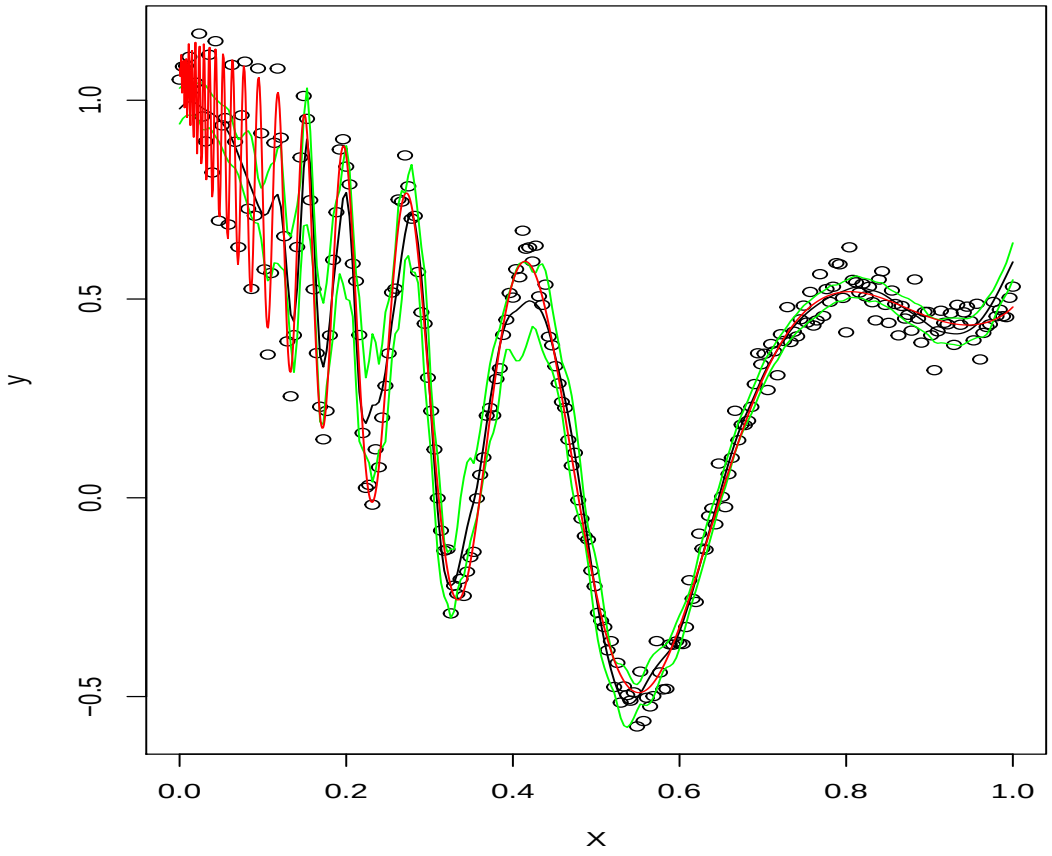


Figure 6.13: An example of a posterior distribution of f from a dataset in Section 6.6.2. We make 256 observations (represented as circles) with a SNR$= 5$, with the true function shown in red, and our method's mean in black. We also show the point-wise 95% central confidence interval in green.

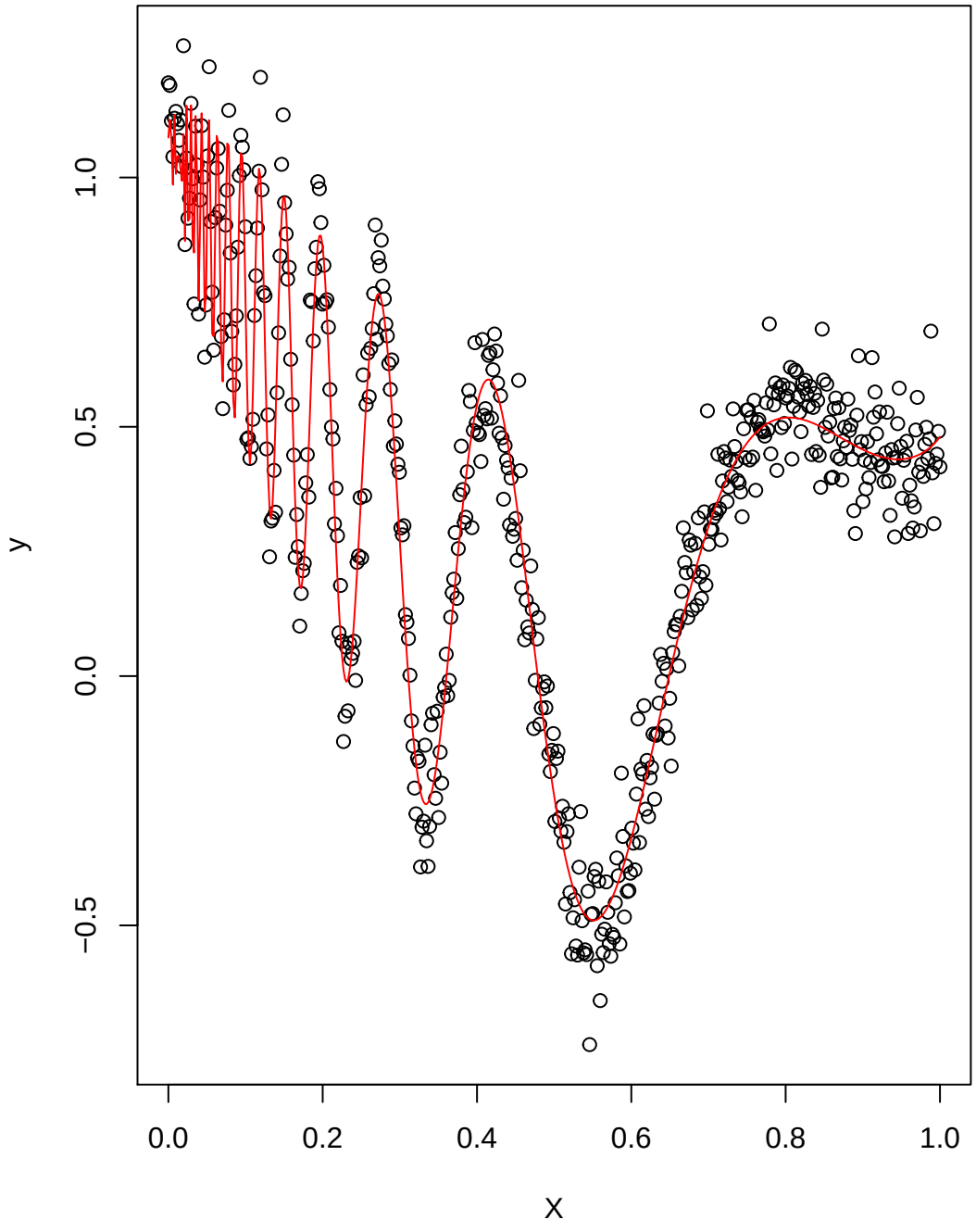


Figure 6.14: An example of a Doppler test function dataset, in red we have the true test function, and the points show a realisation of this test function with a SNR of 5.

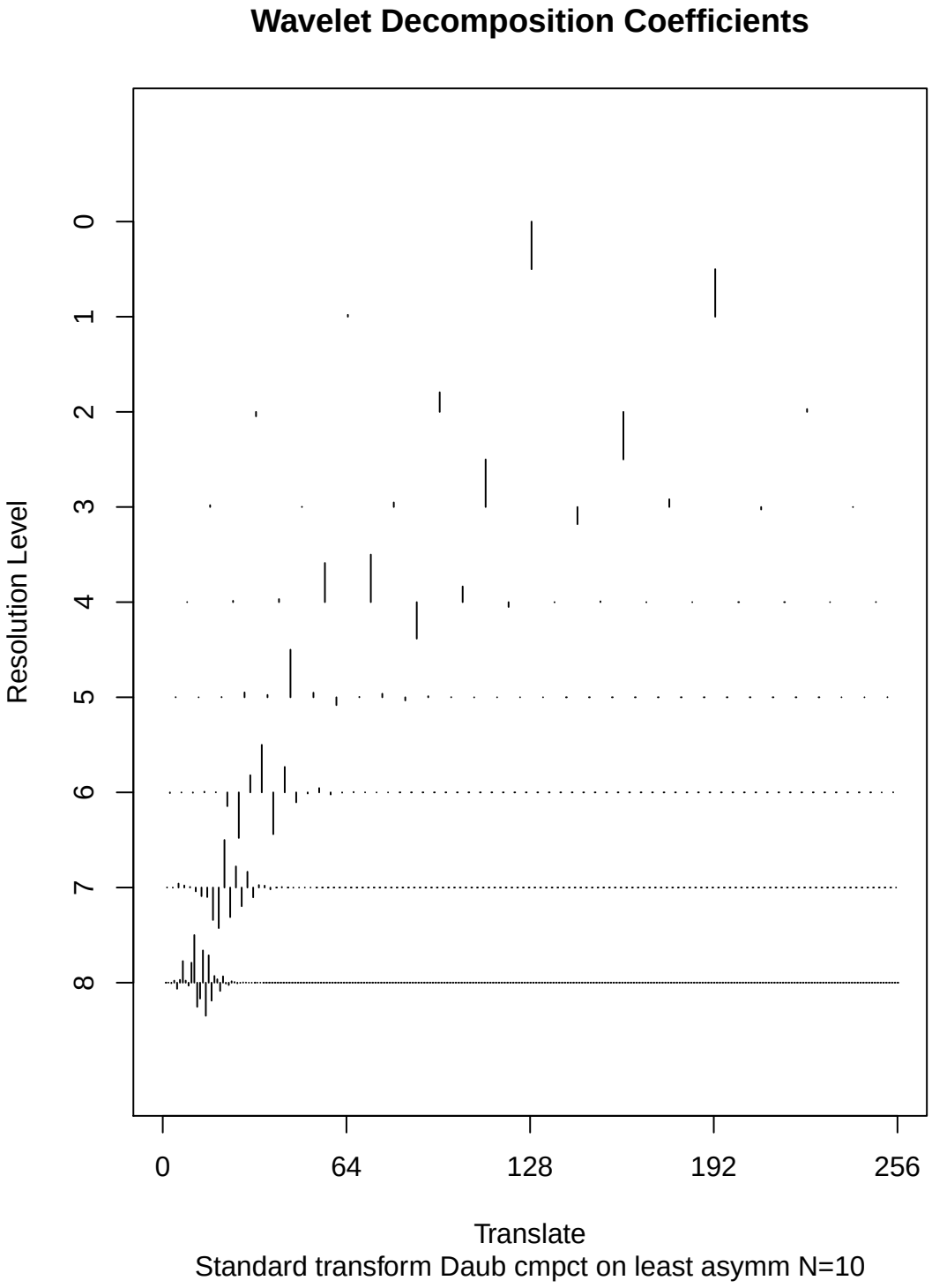


Figure 6.15: The wavelet decomposition of the Doppler test function from equation (6.8), where we have removed the quadratic term. The coefficients are scaled by level.

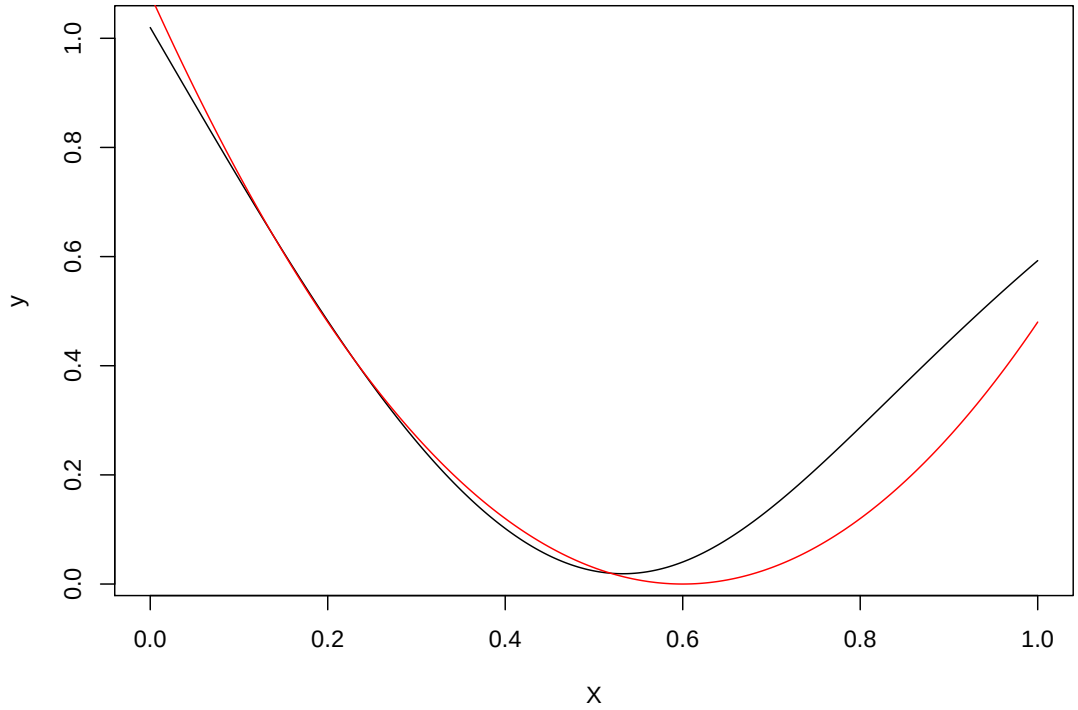
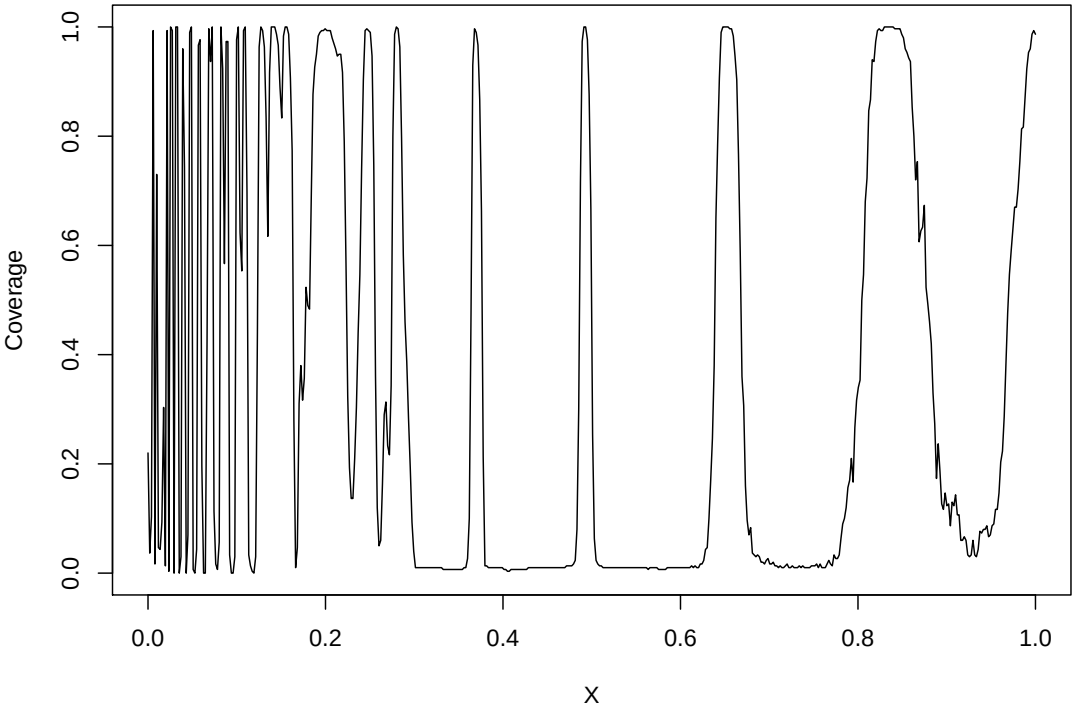


Figure 6.16: The posterior Mean of the Gaussian process (of the first dataset) when we fit one to the Doppler test function which involves a quadratic offset. In red, we have the true offset, in black,we have the Gaussian process mean.



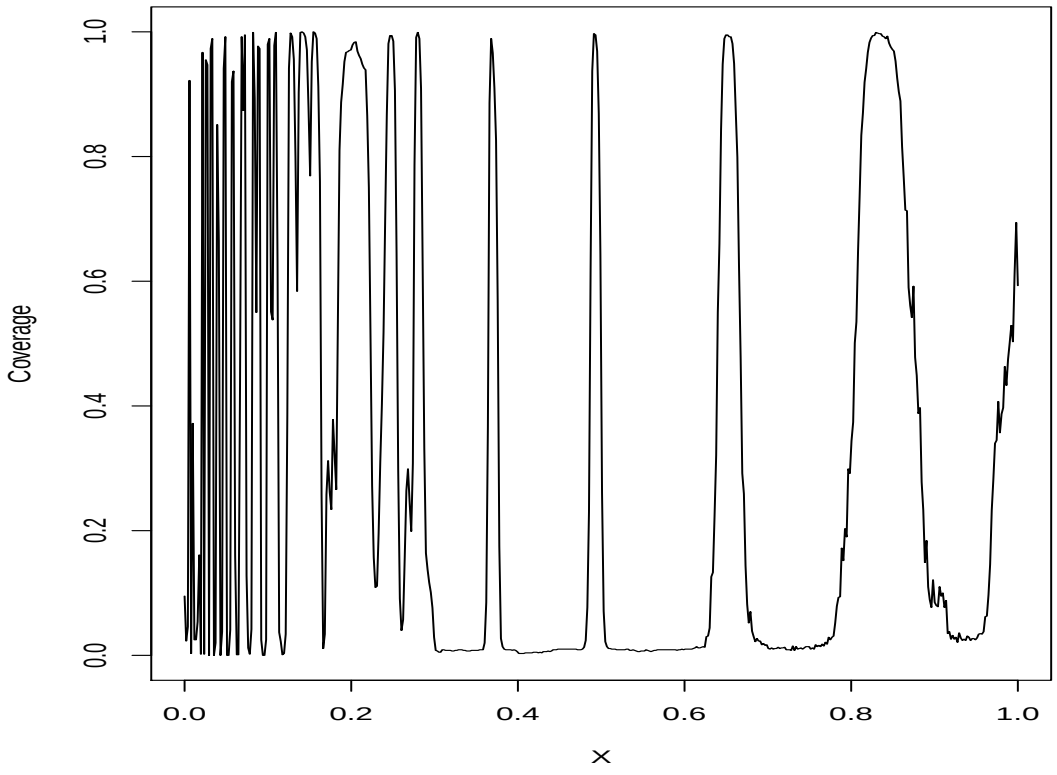


Figure 6.17: Coverage vs x, when we use the GP with uncertainty (top) and just its mean (bottom) as the long term trend; using a decimated wavelet decomposition with the normal PDF as the symmetric PDF in the prior.

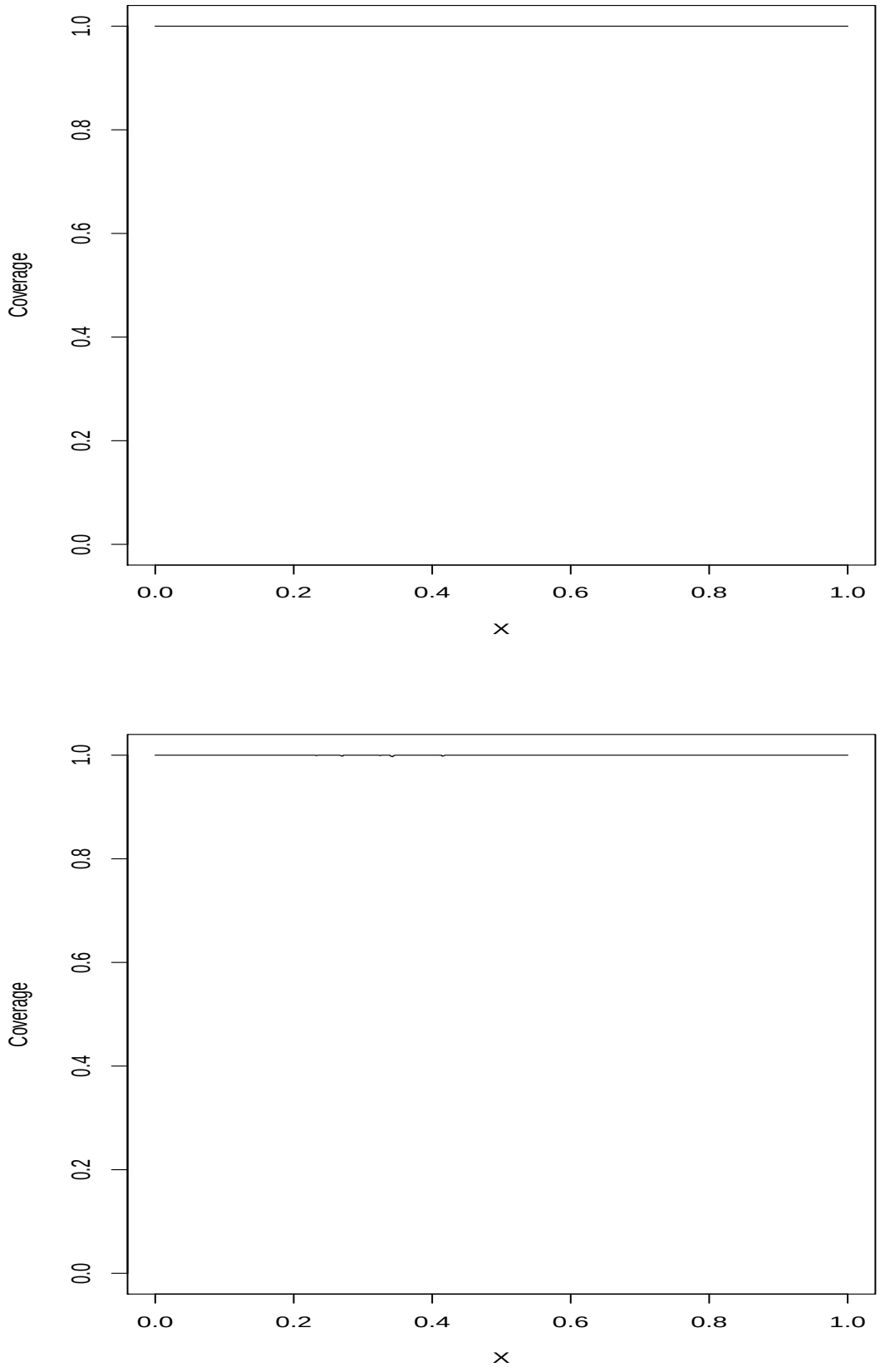
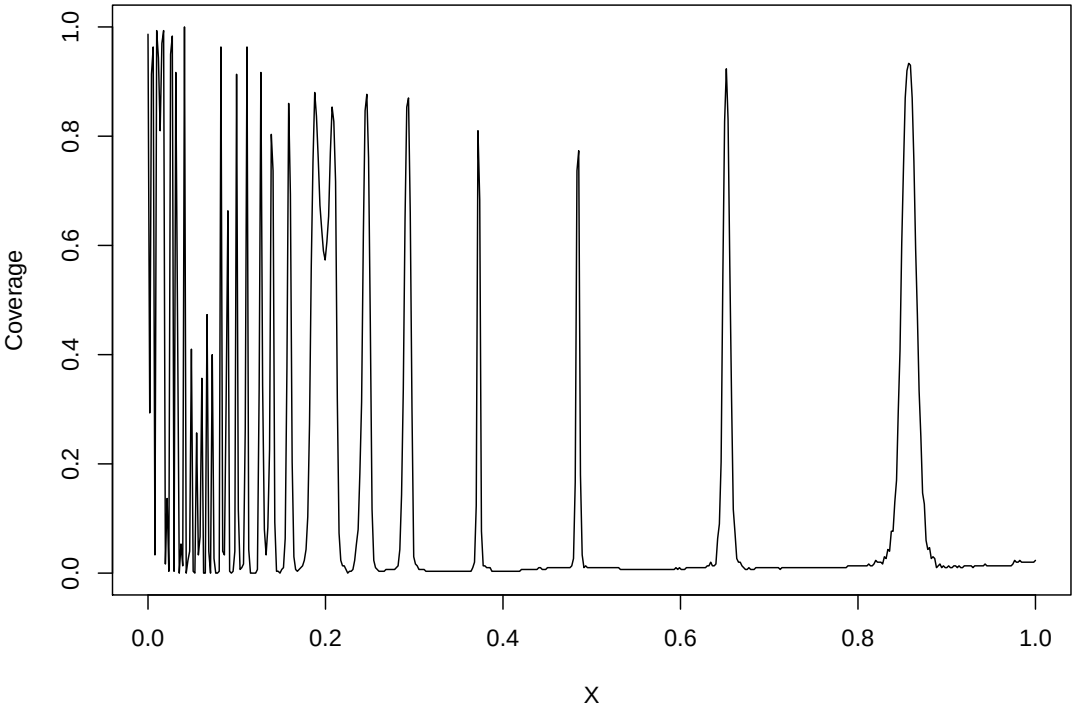


Figure 6.18: Coverage vs x, when we use the GP with uncertainty (top) and just its mean (bottom) as the long term trend; using a decimated wavelet decomposition with the Laplace PDF as the symmetric PDF in the prior.



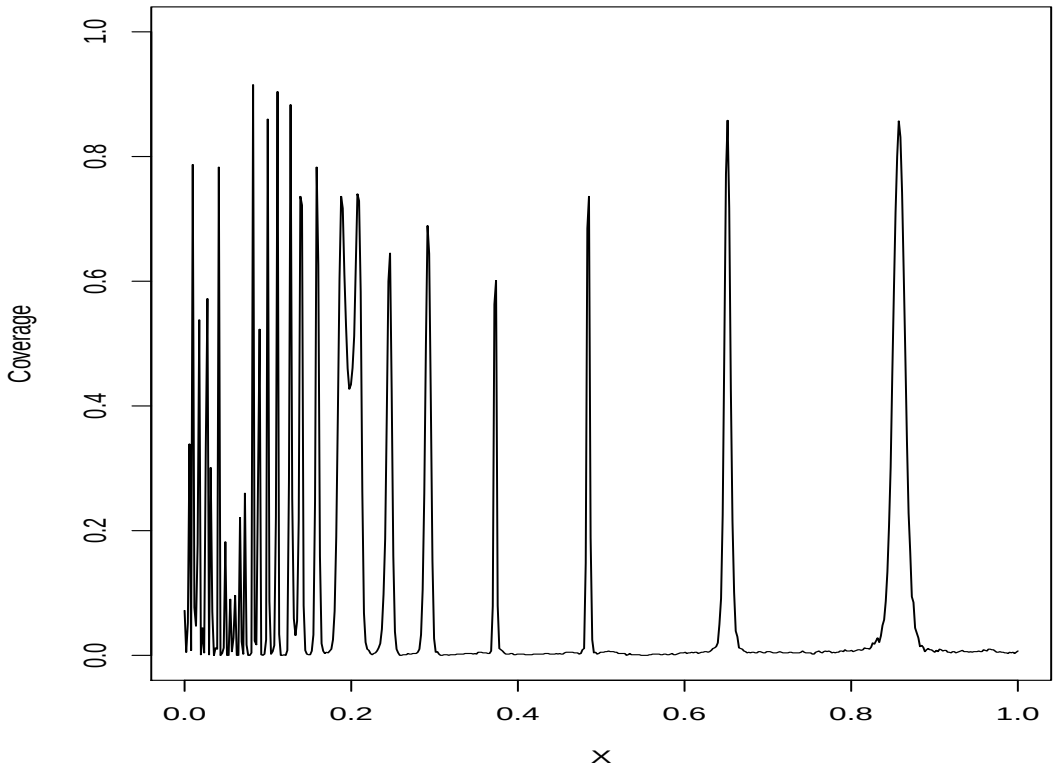


Figure 6.19: Coverage vs x, when we use the GP with uncertainty (top) and just its mean (bottom) as the long term trend; using a non-decimated wavelet decomposition with the normal PDF as the symmetric PDF in the prior.

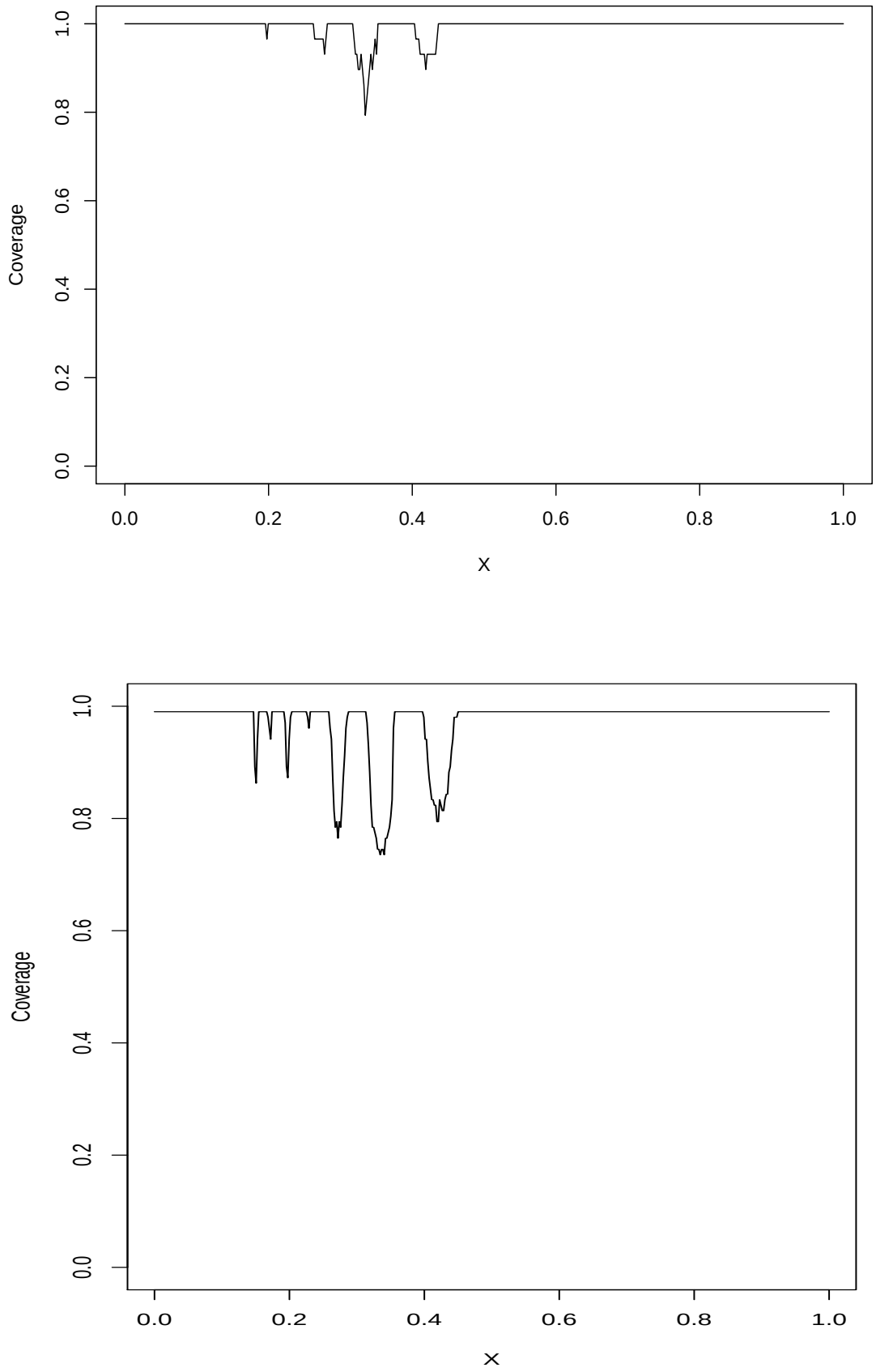


Figure 6.20: Coverage vs x, when we use the GP with uncertainty (top) and just its mean (bottom) as the long term trend; using a non-decimated wavelet decomposition with the Laplace PDF as the symmetric PDF in the prior.

6.7 Conclusions

In this chapter, a method was developed to estimate our beliefs in a one dimensional function using a Gaussian process regression and wavelet shrinkage. A Gaussian process was used to model the overall long term trend in the data, with the remaining trend modelled using a wavelet shrinkage scheme. We are able to utilise the distributions that these methods produce to estimate our uncertainty in the underlying function f. We have shown that the method can be adapted to reflect our personal prior beliefs about the form of the function. Different choices of functions for the long term trend and the different prior distributions for the wavelet coefficients can be seen, and the effect that this has on our posterior inference of the function.

In the next chapter, we explore the challenge of modelling a function that produces multiple observations from a single parameter value. This type of situation can occur, for example, when the output of our function is a time-series. This challenging situation is exasperated when the time-series is correlated. When the time-series is high dimensional, that is, when we observe a large number of time points for a single parameter value, many of the popular method become infeasible. This infeasibility can often be due to time constraints and the amount of memory storage that these methods require. Hence, there is a need to develop a method to reduce the burden on these functions to allow us to be able perform analysis in these situations. This is accomplished through the use of a wavelet decomposition as a tool for dimension reduction, with a Gaussian process used on this reduced dimensional problem.

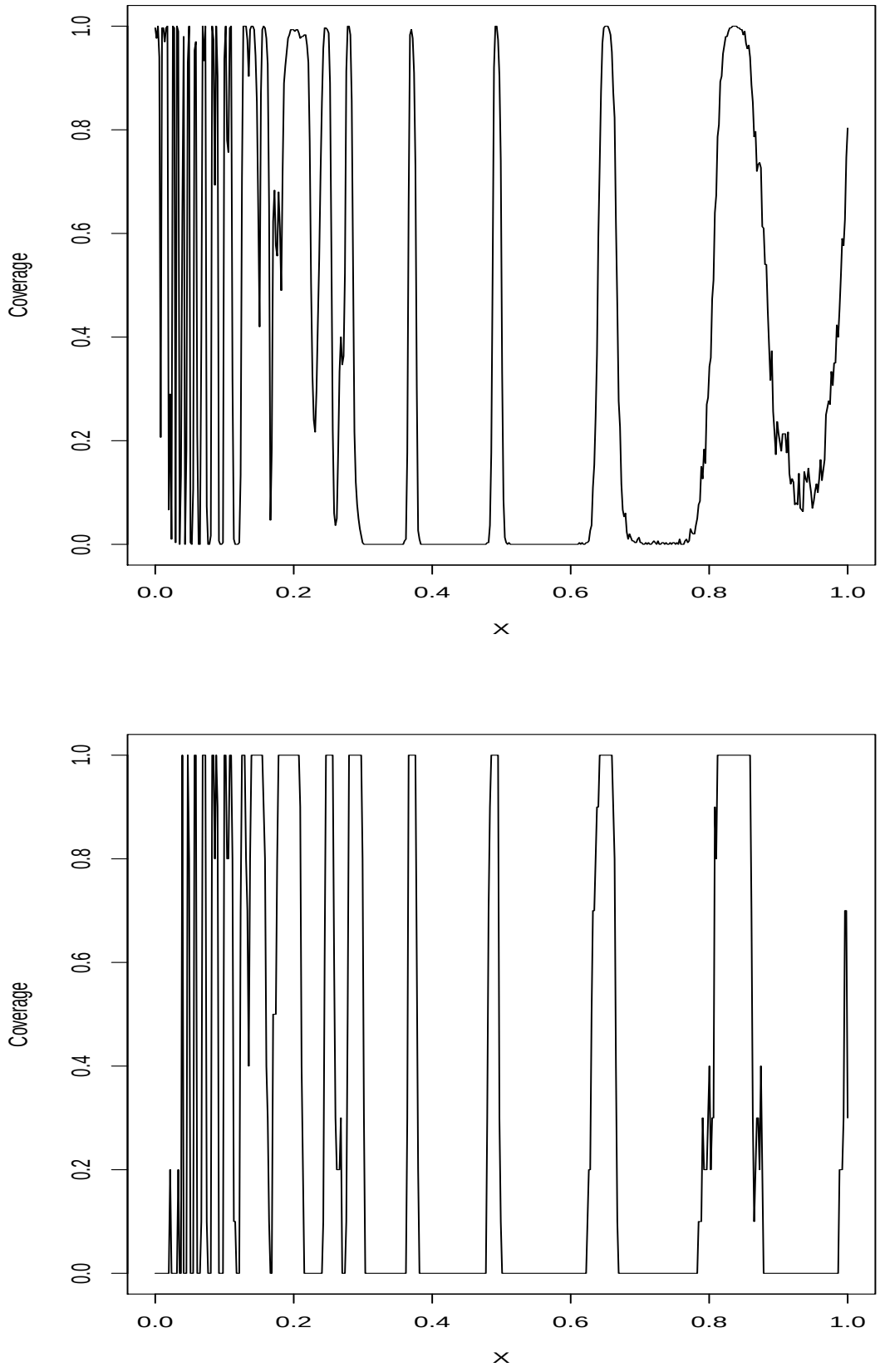


Figure 6.21: Coverage vs x, when we use a linear function with uncertainty (top) and nothing (bottom) as the long term trend; using a decimated wavelet decomposition with the normal PDF as the symmetric PDF in the prior.

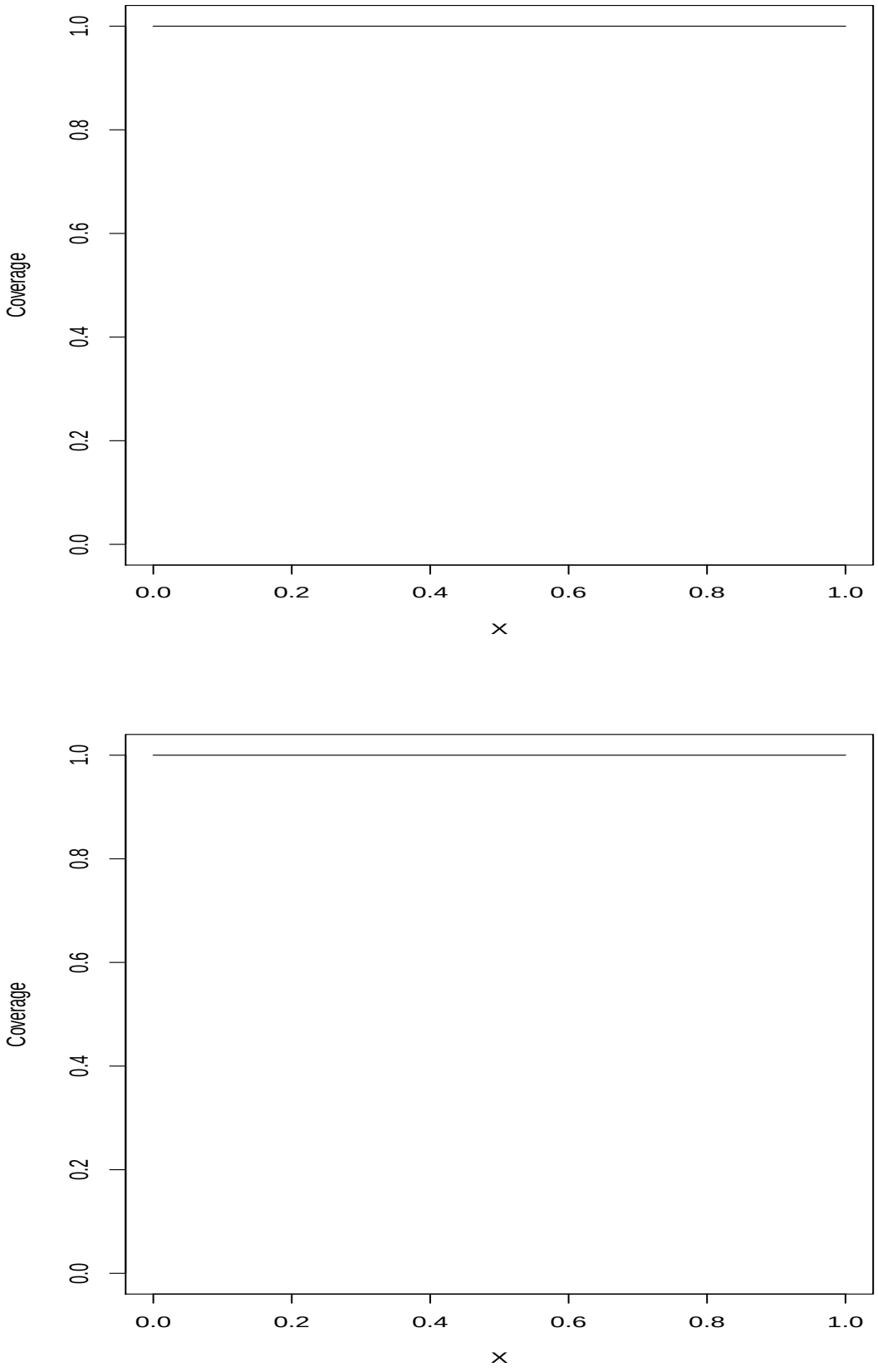


Figure 6.22: Coverage vs x, when we use a linear function with uncertainty (top) and nothing (bottom) as the long term trend; using a decimated wavelet decomposition with the Laplace PDF as the symmetric PDF in the prior.

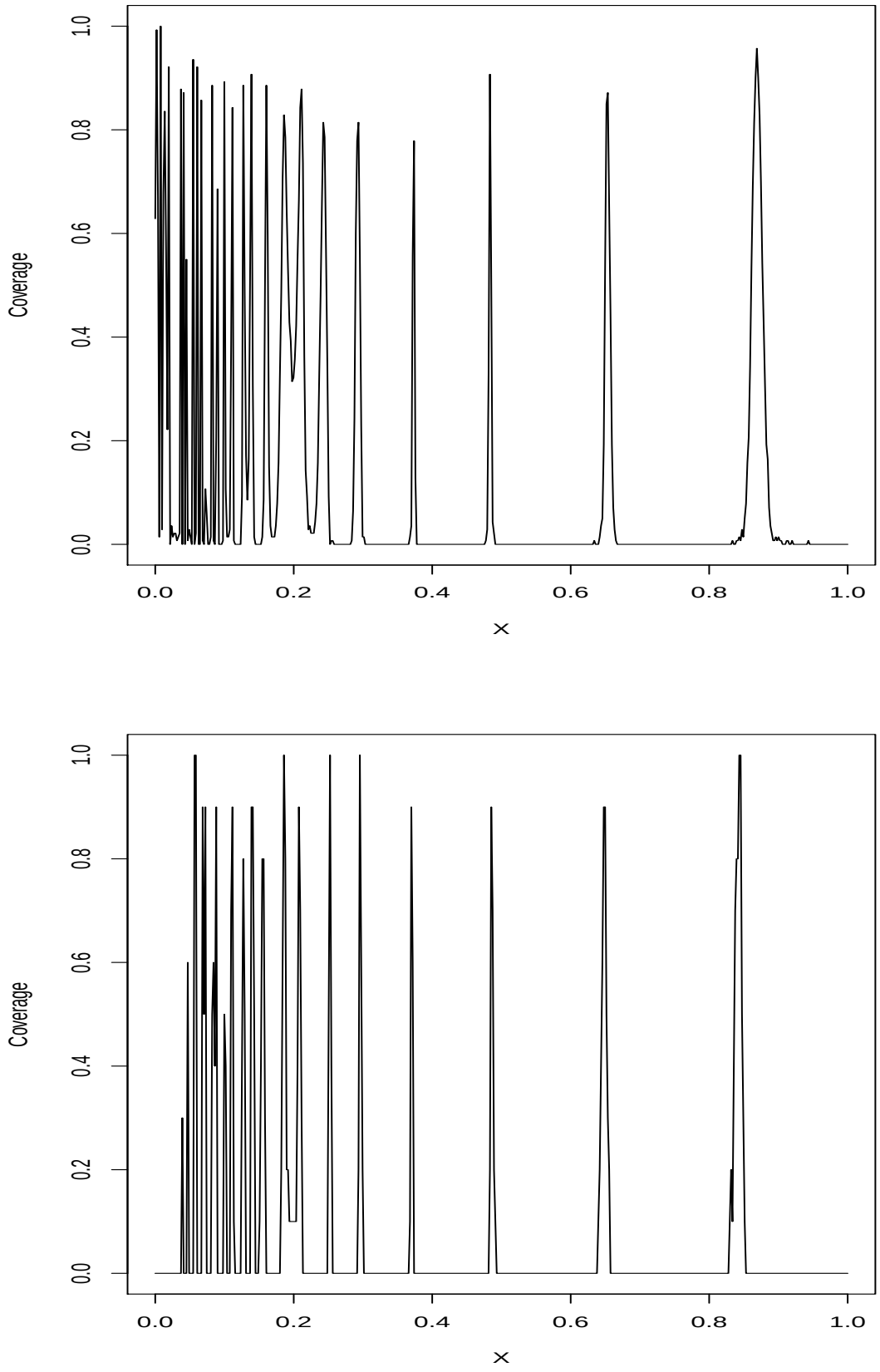


Figure 6.23: Coverage vs x, when we use a linear function with uncertainty (top) and nothing (bottom) as the long term trend; using a non-decimated wavelet decomposition with the normal PDF as the symmetric PDF in the prior.

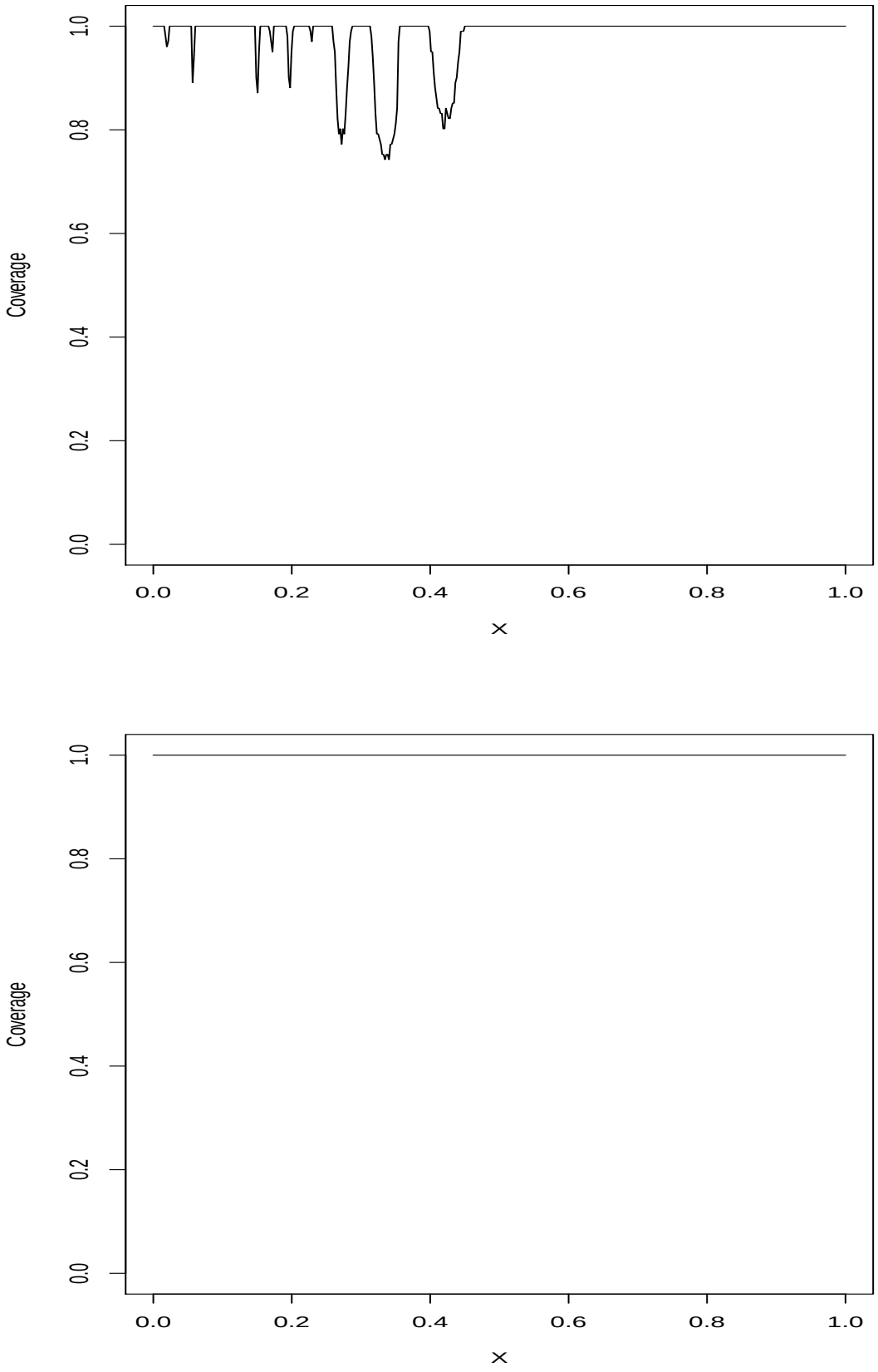


Figure 6.24: Coverage vs x, when we use a linear function with uncertainty (top) and nothing (bottom) as the long term trend; using a non-decimated wavelet decomposition with the Laplace PDF as the symmetric PDF in the prior.

Chapter 7

Dimension reduction of high-dimensional outputs

7.1 Introduction

In practice, we can often come across situations similar to those seen in Chapters 2 and 5, in which we are able to observe realisations of a function at locations $\mathbf{x}$. Unlike the cases seen in the previous chapters, the output that we observe may be multivariate (Higdon et al. 2008). That is, we have a set of n_x parameter/input values which are d-dimensional that we are interested in for our model, which we define by

$$\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_x}\} \qquad \mathbf{x}_i \in \mathcal{R}^d.$$

For any input value $\mathbf{x}_i$, the model calculates a vector of length T time series output

$$\mathbf{f}(\mathbf{x}_i) = (f_i(\mathbf{x}_1), \ldots, f_T(\mathbf{x}_i))^T. \tag{7.1}$$

If we define $\mathbf{y}_i = \mathbf{f}(\mathbf{x}_i)$, then we can see that the output of the model for our input locations is

$$\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_{n_x}). \tag{7.2}$$

Throughout the chapter, we refer to the output at a single input location as time-series data, as was the case in Conti et al. (2009); however, we are not limited to this set-up with the method suitable for situations in which we have vector outputs for a single input value. We can see that this differs from our previously explored situations as we have a vector of observations for a single set of parameter value. This poses an additional challenge as, not only must we adapt the methodology such that we can predict this vector of outputs at a given location, but we also consider the possibility that the outputs are correlated across time (Fricker et al. 2013). If our output is indeed a time series, it is easy to see (and well established in the time series literature (e.g. Chatfield 2016)) the possibility that the value

of the observation at time $t-1$ will be close to the value at time t. In this chapter, we look at a possible solution for when we we would like to represent and predict the vector output model, with particular attention paid to the additional challenges that are presented when the length of the output T is large (hence making $\mathbf{y}$ high-dimensional output).

This problem is not new, and there are a number of methods that are used in an attempt to model situations in which we have a vector output function. Three popular methods can be described, and, again, we explain the case where the indices of the f terms in equation (7.1) are time-points, however, it is emphasised we are not limited to this situation. The first solution is to model each time-point of the output independently, using the methods that were seen in Chapter 2. This solution will be referred to as independent time points (ITP). It can be seen that this solution does not take into consideration the possible relationship between the outputs, as we are building an independent model for each time-point. A counter argument is often used to justify the use of this method however; if each time-point is modeled very well independently, there may be no need to consider this additional information and complexity given that the improvements that can be made are minimal. For example, we may have the stationary time-series

$$\begin{aligned} z_1 &= 2 + x \\ z_t &= 0.8 z_{t-1} + 2 \quad t \in \{2, \ldots, 2^5\}. \end{aligned}$$

We can then perform a non-linear transformation on this time series to make a test function

$$y_t = 0.02 z_t^2 + 0.2 z_t \quad t \in \{1, \ldots, 2^5\}. \tag{7.3}$$

Using 15 random x locations between zero and 16, we can visualise the data, which can be seen in Figure 7.1. In this figure, we can see examples of what a time series test function could look like in the top picture; in the bottom picture, we see a visualisation of the data when the IGP method is used. For the IGP method, we split the data into those that have the same timepoint (or colours and shape in our visualisation), and build a GP for each of these time points.

Another popular method, is to include the time-point as an input parameter, and this solution will be referred to as time as input (TAI). That is, rather than having a vector output, as was seen in equation (7.3), we instead have a scalar input and use the index of the position of the scalar in the vector as an additional input parameter. The final method that is discussed is the Multi-output Gaussian process introduced in Conti & O'Hagan (2010), which will be referred to as multi-output emulator (MOE). The MOE uses a matrix Gaussian process, in which we introduce two new correlation terms – one that describes the correlation between different time-points in the output, and one that describes the correlation between a time point and an input parameter. In the paper, the authors show the relevant derivations for a posterior analysis when we set both of these correlations to

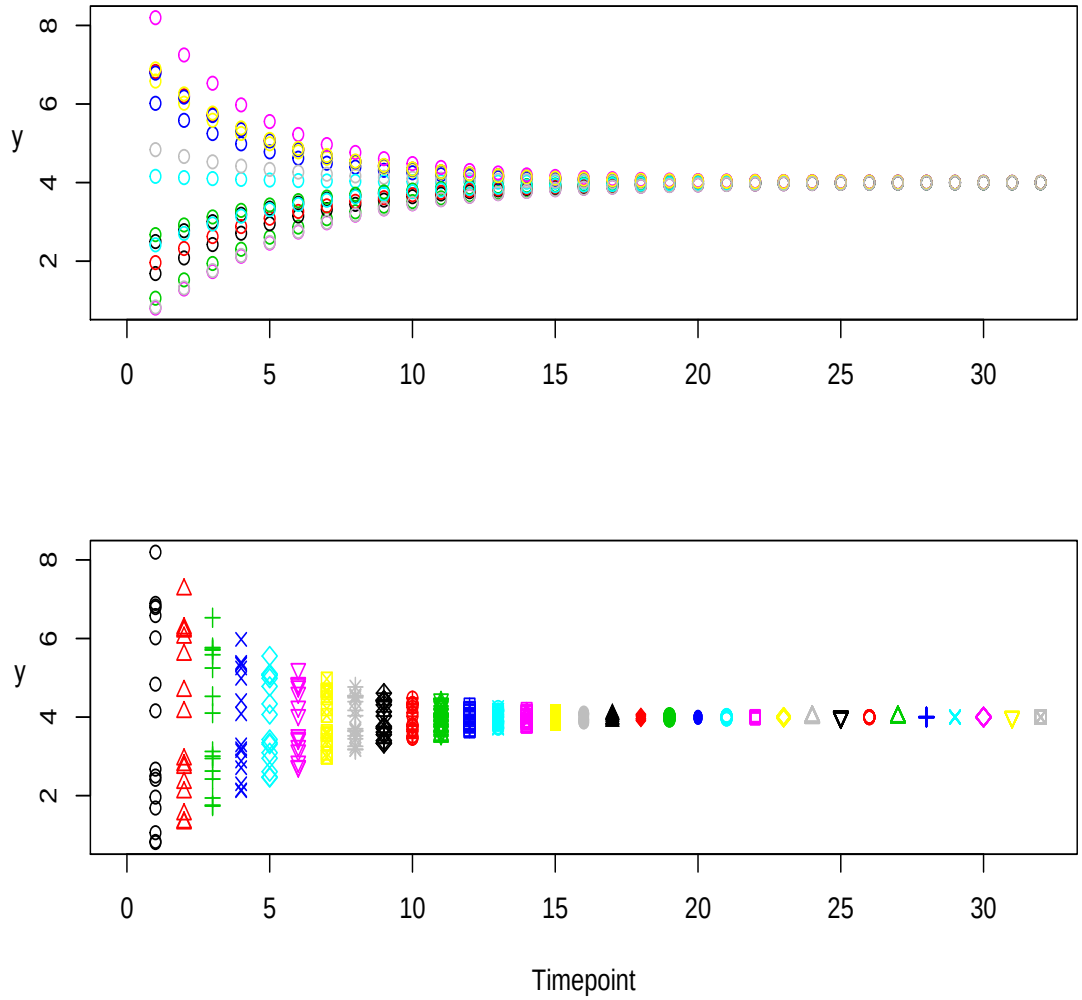


Figure 7.1: Top: The test function from equation (7.3), observed at 15 x locations; Bottom: A view of the data if we use the IGP method; each point that has the same colour and shape is data to be used in a Gaussian process.

be identity matrix, and note that using alternative choices for these correlations result in a more difficult derivation for inference. It can be seen in the methodology that both MOE and TAI could allow for correlation between the different outputs, with TAI including the time-point as an input to account for the correlation and MOE using a correlation matrix to describe the correlation. This makes both of these methods more flexible than ITP, in this respect.

When considering the parameter Σ, the covariance matrix for two different input locations at the same time point, that is, the covariance matrix for two points $\mathbf{x}_i$ and $\mathbf{x}_j$ for the same time-point t, ITP typically has the advantage over both TAI and MOE. In the case of TAI, it is easy to notice that when we have the same time-point t, the covariance between two points is a function of the inputs, resulting in the same Σ, regardless of the time-point. In the case of MOE, typically, Σ is assumed to be the same regardless of the time-point for

simplicity (Conti & O'Hagan 2010). Changing this structure and allowing Σ to change for different time-points makes the analysis much more challenging. ITP, however, allows us to build an independent Gaussian process on each time point. This provides a simple method to allow Σ to depend on the time-point, making it advantageous over MOE and TAI in this respect. These methods can also be compared by looking at the number of operations, and, hence, speed of the analysis. The longest operation is the inversion of the covariance matrix. For example, using the Gauss-Jordan elimination method, this requires $O(n^3)$ operations (Stanimirović & Petković 2013), where n is the dimension of the square matrix. It can be seen that the fastest of these solutions, in that respect, is MOE, which requires the inversion of an $(n_x \times n_x)$ matrix, where n_x is the number of input locations that we observe, and hence remains $O(n_x^3)$. The ITP requires a larger number of operations, as we are building T independent Gaussian processes, we hence have to perform an inversion on t separate $(n_x \times n_x)$ matrices, making the total order of all of the inversion $O(Tn_x^3)$. Typically, when we have $O(Tn_x^3)$, we would simply report that the number of operations is $O(n_x^3)$ as the n_x^3 term dominates the constant T. However, the T term is still reported for this chapter, as we are interested in cases in which we have high-dimensional output, such that the term n_x^3 does not dominate T and is comparable. The TAI method requires the largest number of operations out of the three methods, with the time points being included in the input and, hence, increasing the dimensionality of the input space. This increase of the input space results in the inversion of an $(Tn_x \times Tn_x)$ matrix, and hence is $O\big((Tn_x)^3\big)$. Of course, an alternative to speed up the speed the inversion of a matrix that we do not explore in this thesis is by the approximation of the inversion. We could, for example, use a spectral approximation to approximate the inversion and would reduce the number of operations that are required.

As we can see, the MOE method seems to be a good method to use, when we have the situation in which we are attempting to model a function with multiple outputs. This method is advantageous over ITP and TAI when we consider the number of operations required for the inversion of the covariance matrix, which is crucial when we are analysing the posterior distribution of the Gaussian process. However, the ITP and TAI methods are still used over the MOE method, due to the complexity of MOE, as well as some of the other positives that have been detailed. As we have seen, the ITP and TAI methods include the value of T in the order for the inversion of Σ, which, when $T > 1$, results in more operations being required for this calculation. For low values of T, this operation still tends to be relatively quick. For large values of T however, these Gaussian processes can become computationally heavy and, hence, makes these methods an unattractive solution. Therefore, there is a need to find a way to speed up these calculations. The problem is tackled in this chapter, using the data compression utility of wavelets, discussed in Chapter 3.

7.2 Using wavelets for data compression

7.2.1 Method

In Chapter 3, some properties of wavelet methodology were discussed. One of the main properties of the wavelet decomposition is its ability to provide a sparse representation: of a function using the continuous wavelet transformation, or a vector if we use the discrete wavelet transformation. That is, using a discrete wavelet decomposition, we expect the main features of a vector to be encapsulated by a few large coefficients, hence, providing a sparse representation of the vector (e.g. Abramovich et al. 1998). This property sees the wavelet decomposition used for data compression — rather than storing a dataset of size n say, we can instead store the n_d large wavelet coefficients, where $n_d < n$ (Rothwell et al. 1994).

We have a number of parameter values $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{n_x}\}$ for which we will observe the function in equation (7.1). Observing the equation at these parameter values will give us our data, $\mathbf{y}$, as was seen in equation (7.2).

For any data vector of size T, if T is dyadic, we can perform a discrete wavelet decomposition on that string of data to give us our set of wavelet coefficients. We assume that most of the information from the vector is stored within a few large coefficients, and hence if, instead of considering all of the T wavelet coefficients, we consider a subset of the largest of these coefficients, we can provide an approximation to the function. Using this subset of size $n_d < T$, we can set all other coefficients to zero. The value of n_d can be set using a fixed number of coefficients, or alternatively, n_d could be set to be (with rounding) a percentage of T. If we are to retain the largest absolute coefficients, the boundary conditions must be checked as, if there is a problem, these large coefficients will always be selected (see the discussion of the boundary problem in Chapters 3 and 6).

We are then able to fit independent Gaussian processes to each of those n_d coefficients. By doing so, we reduce the computational load of when we perform all of the inversions of the covariance matrices to $O(n_d n_x^3)$ as opposed to $O(T n_x^3)$. This gives us a possible solution if we wish to use independent Gaussian processes to model equation (7.1) when we have a large T. The TAI method could also be implemented using the wavelet coefficients, however, implementation of the wavelet transformation typically reduces the correlation between the coefficients. This may result in a non-smooth relationship between the coefficients and, hence, using a smooth covariance function between the coefficients may not be advisable. We could also use the MOE method with these coefficients, however, it will not be advantageous over using MOE on the data vector, as we would rather use the MOE method of the full vector as opposed to an approximation of the vector.

7.2.2 The accuracy of the approximation

One key consideration in the method is the number of wavelet coefficients that will be considered in our method, with a Gaussian process prior fit for those coefficients that are retained. As in previous wavelet shrinkage methods, like those discussed in Chapters 3 and 6, those coefficients corresponding to the coarsest levels are often very informative of the underlying function. In the shrinkage methods, this led to those coefficients not being included in the analysis, as we assumed that they would be signal and not be attributed to the error (e.g. Johnstone & Silverman 2005). For this method, however, we instead ensure that the coefficients are included in the analysis. This difference is due to the differing objectives, in the wavelet shrinkage we were interested in removing the noise and, hence, the coefficients were kept the same and not included in the analysis; for this method, we want to find those coefficients that contain most of the information of the function, and, hence, the coefficients are included.

Not only must we decide on the level in which all coefficients at that level and coarser will be automatically included, but we must also consider the number of the remaining coefficients that we will also include in our analysis. The more coefficients that are used for the method, the slower our computation will be. As one of the key aims of the method is to reduce the computational time required to analyse the multiple output models, we will be using only a proportion of the possible coefficients. As such, we can see visually the effect that removing wavelet coefficients (in other words, setting them to zero) when we reconstruct the function has on our estimation of a function with an example. The function that will be used for this example will be the time evolving function

$$\begin{aligned} f_1(x) &= 2x + \sin(2x^2), \\ f_t(x) &= f_{t-1}(x) + \sin(t) + 0.001t - 0.00001t^2 - 0.1\cos(x). \end{aligned} \tag{7.4}$$

We will observe the function at 256 timesteps, keeping the function fairly quick, as the utility of this example is to show the ability of the approximation. We run the function at $x = 0.25$ for the example, with the coefficients belonging to coarsest three wavelet level automatically retained. A Debauches wavelet coefficient is used with ten vanishing moments for this exploration. This was done as we expected the output function to be smooth, and, hence, a wavelet with ten vanishing moments will be able to ,odel the function with less large coefficients. To deal with the boundary conditions, a linear interpolation is subtracted from the vector for the decomposition, so that the vector has the same first and last element. When 20% of the remaining coefficients were kept, we found MSE $= 0.1014$; when 30% of the remaining coefficients were kept, we found MSE $= 0.0174$; and when 40% of the remaining coefficients were kept, we found MSE $= 0.0033$, all rounded to four decimal places.

This can be seen visually in Figure 7.2, in which we have focused on the time-points between 50 and 100. It can be seen visually that, as we would expect, as more coefficients are retained, the approximation to the function becomes more accurate. Further to this, we are able to show the approximation for an array of possible coefficients retained, and, in Figure 7.3, we plot the proportion of coefficients that are retained in our approximation, and their respective mean squared error. We can see that we have an curved reduction in MSE as more coefficients are retained, with the MSE converging to zero as we retain all of the coefficients. Therefore, we can see that there is a trade off between accuracy and speed for our method, and the percentage of coefficients that we will retain, which we will label α_l, needs to be carefully considered.

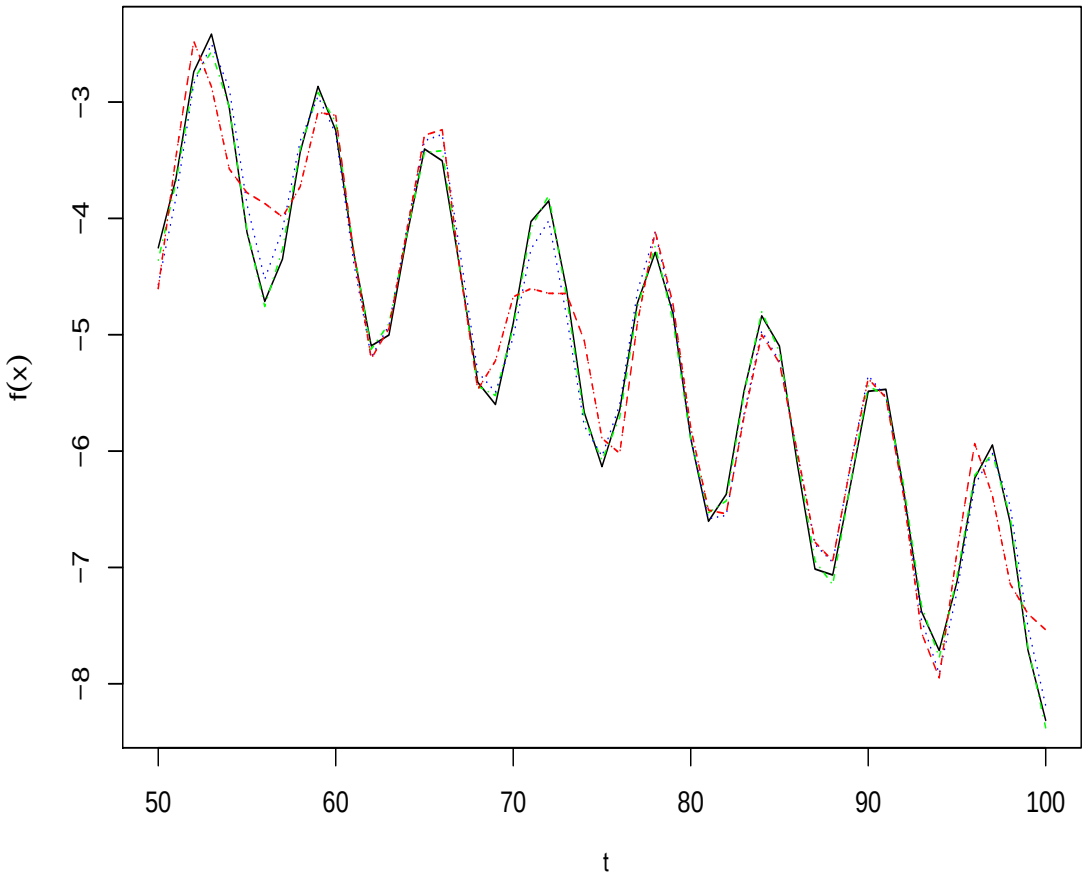


Figure 7.2: The approximation of a function using wavelets, the true function is seen in black. The coarsest 3 level's coefficients are automatically retained, and differing proportions of the remaining coefficients are also retained. We can see the function when 20% (red dashed), 30% (blue dotted), and 40% (green dashed) are retained

In this subsection, we have shown the effect that selecting a subset of the largest

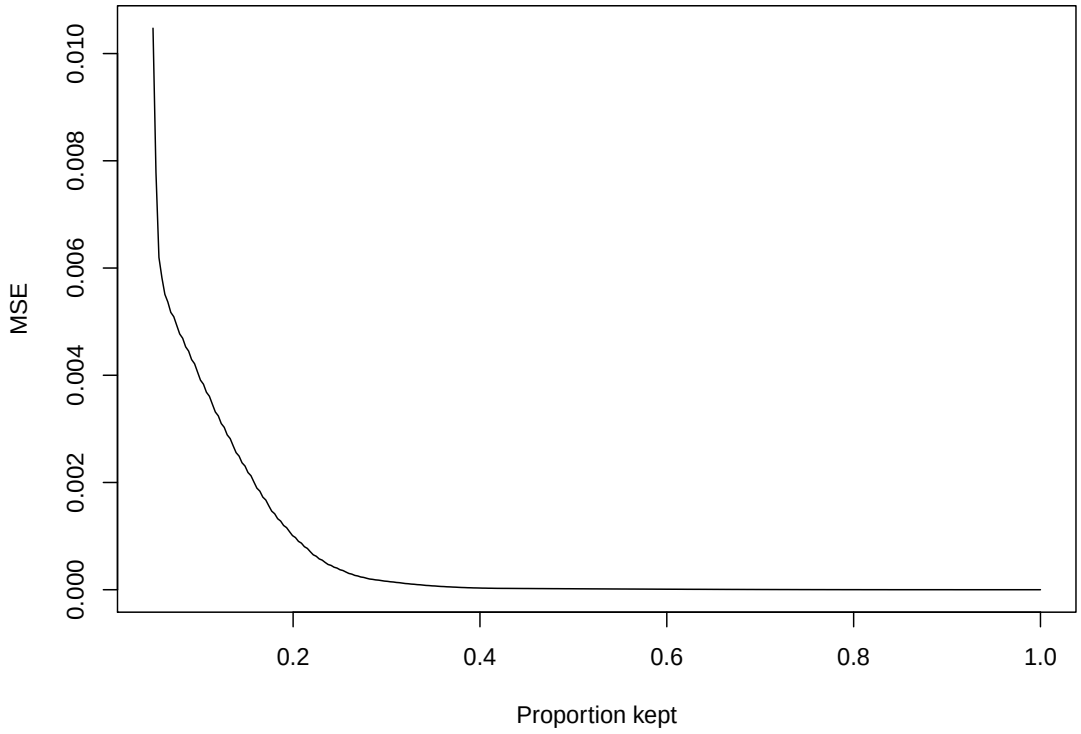


Figure 7.3: The mean squared error of the approximation to the data vector, the proportion of the wavelet coefficients retained after the third coarsest level is shown on the x-axis. We use a Daubeches wavelet with ten vanishing moments on the test function, seen in equation (7.4).

wavelet coefficients has on our representation of a vector. We must, however, consider the challenge of identifying the largest coefficients when we have more than one piece of data. When we have just one vector, it is natural to identify the largest (by some measure) coefficients by performing a discrete wavelet decomposition on that vector, and simply identify those large coefficients through the decomposition. Hence, when we have more than one data vector, we also have more than one set of coefficients, in which we must decide which are the largest/most important coefficients. One method that could be used to identify those important coefficients is by looking at the aggregate of the coefficients from the data vectors. That is, we sum the absolute value of the coefficients over all of the observed data, and use these sums to identify the largest coefficients. An alternative method for identifying the largest coefficients is to identify the largest α coefficients in each dataset, we then select to retain those coefficients that are identified as the largest coefficients the most number of times. In all of the simulations in the chapter, both of these methods were implemented and the coefficients that were identified as being the largest were identical in almost all of the cases. With this in mind, it is recommended

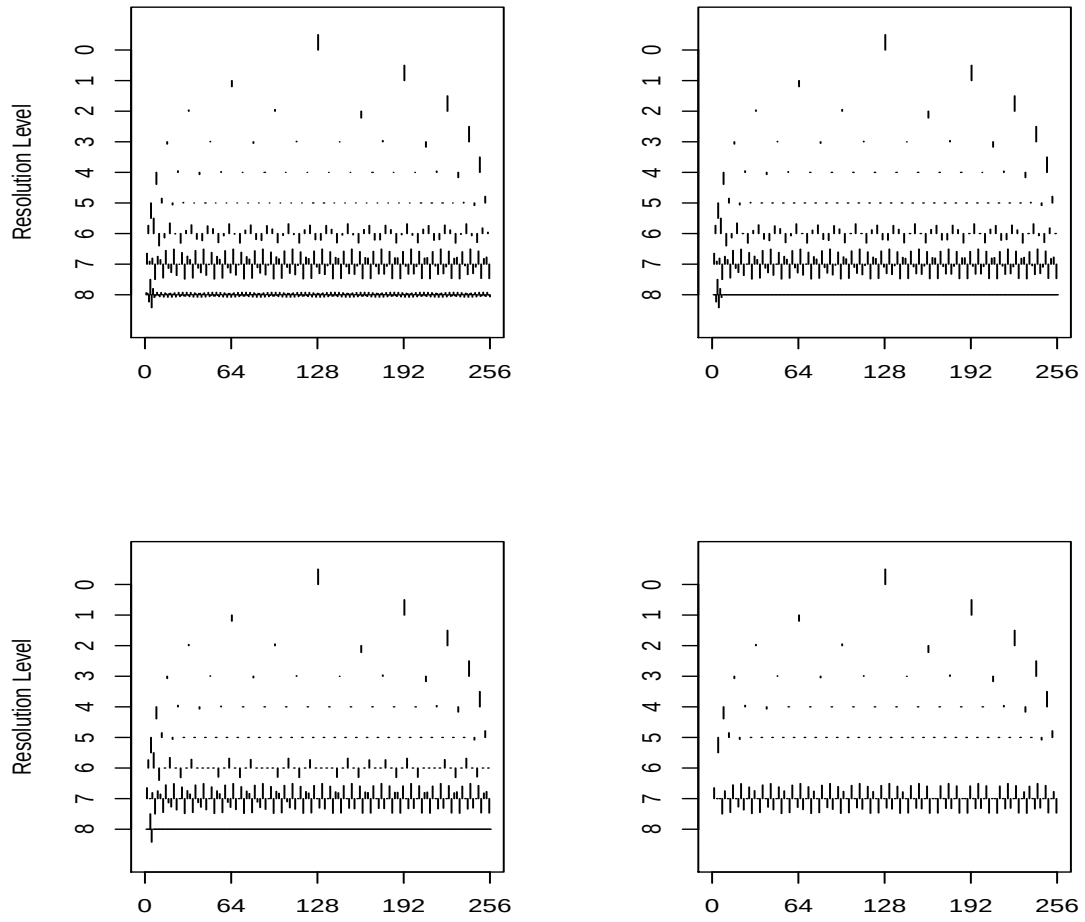


Figure 7.4: The wavelet coefficient for the different levels of approximations. The top three levels automatically retained, and different proportions of coefficients retained for the remaining coefficients; Top left: all coefficients retained; Top right: 40% of coefficients retained; Bottom left: 30% of coefficients retained; Bottom right: 20% of coefficients retained

that the first method should be used to identify the largest coefficients, due to its ease of implementation, unless there is evidence that the alternative method is more representative. Due to the dimension reduction that the use of a wavelet decomposition method we have discussed introduces, there is the possibility that other methods (such as the TAI) could also be used with this set-up. For TAI, as was discussed in Chapter 7.1, the number of operations required to invert a matrix was cubic in T (the dimension that has been reduced), and so for this to become a viable option the number of dimensions must be greatly reduced. However, in most cases in which we have a large value of T, we would require a very large proportion of the wavelet coefficients to be set to zero. When we greatly reduce the number of wavelet coefficient used, as was shown in Figure 7.3, the

approximation of the function becomes poor and so although it is possible to use TAI in theory, the poor approximation may result in our inference becoming too inaccurate. The reduction in dimension could also be used in combination with the MOE method, with the required number of operations reducing, however the complexity problem associated with the MOE remains and hence the dimension reduction does not solve this problem.

7.2.3 Algorithm

Here, in Algorithm 5, we detail the algorithm that can be followed to produce an estimate of a function that has the same form as equation (7.1) using our method. In the algorithm, we denote DWT$(\cdot)$, which is the discrete wavelet transformation and gives us our wavelet coefficients, and we also use IDWT$(\cdot)$ to denote the inverse discrete wavelet transformation.

Algorithm 5 The data suppression method for Gaussian processes

Input: $\mathbf{x}_1, \ldots, \mathbf{x}_{n_x} \quad \in \mathbb{R}^d$ - input locations.

$\mathbf{y}_i = (f_1(\mathbf{x}_i), \ldots, f_T(\mathbf{x}_i)) \qquad i = 1, \ldots, n_x$ - output of model at $\mathbf{x}_i$.

$\alpha \in (0, 1]$ - threshold for proportion of coefficient retained.

Output: $\mathbf{y}^* = \hat{\mathbf{f}}(\mathbf{x}^*)$ $\mathbf{x}^* \in \mathbb{R}^d$ - prediction at unseen location.

$\mathbf{d}_i = \text{DWT}(\mathbf{y}_i)$ - the collection of coefficients for $\mathbf{y}_i$.

Find set S s.t. $S = \{j : d_{ij}$ in coarser level than $l \mid |d_{ij}| > \alpha\}$.

$\mathbf{d}_i^* = \mathbf{d}_i[S]$.

Use a Gaussian process prior for each element of $\mathbf{d}^*$.

For new unseen location $\mathbf{x}^*$, $\hat{\mathbf{d}}(\mathbf{x}^*)$ using posterior mean of the Gaussian process.

$\mathbf{y}^* = \text{IDWT}(\hat{\mathbf{d}}(\mathbf{x}^*))$.

7.3 Examples

7.3.1 The efficiency and accuracy of the method

We explore the accuracy of the method detailed in Algorithm 5, as well as how quickly the method performs by again using the test function in equation (7.4). To do so, we can change aspects of the test function and analyse these to see how the performance is affected in different situations. To give a benchmark for the method, we compare it to the ITP method introduced in Section 7.1. The results of the TAI method are not reported due to the time constraints involved in the method; In the case in which we have $T = 2^{12}$ and $n_x = 16$, we require the creation, storing, and, most importantly, inversion of an $(262,144 \times 262,144)$ matrix. In situations in which the covariance matrix is this large,

which is the subject of this chapter, this method becomes infeasible. Here, we explore the effect that the dimension of the function has on our analysis, that is, we will look at the accuracy and speed of the method for when we have a varying number of time-points, T.

As described earlier in Section 7.2 and previously in Chapter 2, the speed of the Gaussian process suffers as the number of inputs increases, due to the inversion of the covariance matrix that is required for its calculations. Our method, the wavelet Gaussian process (WGP), and ITP both do not circumvent the requirement of inverting an $n_x \times n_x$ matrix, however it does require fewer of these inversions due to the number of coefficients that we retain being smaller than T. This means that fewer Gaussian processes are required to account for the multidimensional output. We explore the effect of having a larger, or smaller number of function runs, n_x in our analysis. The efficiency of the method will be very much dependent on this, as it dictates the speed of the individual Gaussian processes that we build. Finally, we also look at the efficiency of drawing samples from the predictive distribution at an unobserved location X, which could be used in subsequent analysis, or to estimate factors such as the expectation and uncertainty bands. In the analysis, a Debauches wavelet with ten vanishing moments is used for all of the examples, and, to handle boundary conditions, a simple linear regression in t was subtracted from the data vector for the wavelet transformation, with this subsequently added after the inverse wavelet transformation.

As our performance measures, the time taken to produce a functional output using the respective method, and the standardised mean squared error (from equation (2.13)) of the method is used. To produce more reliable results, the experiment is repeated 1,000 times, with randomised input locations for $x \in [0, 1]$, drawn from a standard uniform distribution. The locations are selected uniformly, as to represent a variety of situations in which we may have less data points around certain areas of input space that we want to predict, or conversely, many data points. The means of both the time and the SMSE are found over the 1,000 experiments and are reported in Table 7.1. In Figure 7.5, we also show an example of coefficients of the wavelet decomposition for an example that was run. We can see that the coefficients appear to be following a smooth curve, suggesting the Gaussian process is an appropriate distribution to use for the wavelet coefficients. Just one example has been shown to display the nature of the coefficients of the wavelet decomposition, and, for this decomposition, every coefficient had the same smooth characteristics. For the examples, the coefficients belonging to the coarsest three levels are automatically retained.

In Table 7.1 we can see the results of our method, and the ITP method for our analysis. It is noticeable that, for situations in which we have a low number of inputs and the dimension of the output is also low, then WGP is actually slower than that of the ITP method. It can also be seen that for these situations, the SMSE of the two methods are

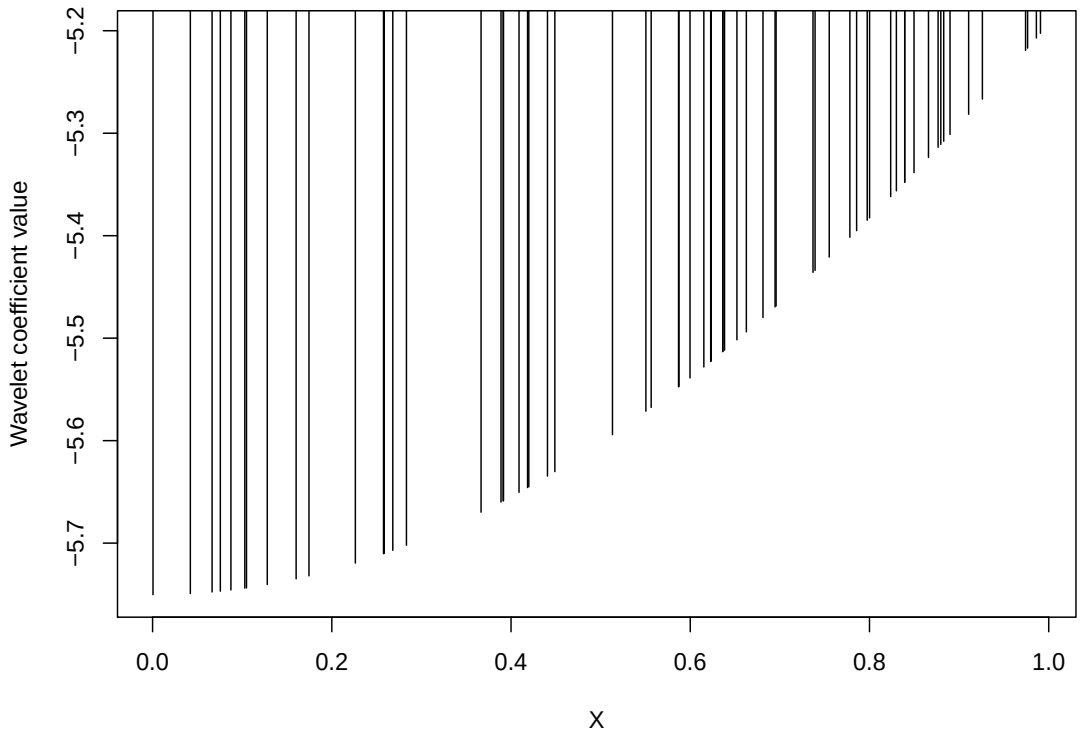


Figure 7.5: An example of the wavelet coefficients that were use to build the Gaussian process in the WGP method for different x values. In this example, we have selected to display the coefficients $d_{5,5}$, in which we have $n_x = 64$. A Daubeches wavelet with ten vanishing moments has been used to obtain these coefficients. The smoothness of the wavelet coefficients for different values of $\mathbf{x}$ can be observed here, justifying the use of a Gaussian process.

very similar. This is due to the fact that the test function does not have as many features to represent, and so, the approximation using the WGP method is more accurate. This property results in an analysis that would be very similar to using the raw data. The WGP shows an advantage in terms of efficiency over the ITP method when the number of inputs becomes larger, and the dimension of the output increases, as we would expect. This is especially noticeable when $n_x = 64$ and $n_d = 2^{12}$ in which we see times that are around three times quicker than ITP. It is also noticeable that the SMSE becomes worse for the WGP compared to the ITP method, as n_x and T become larger. This may suggest that we could include more of the wavelet coefficients to rectify this — if time allows it.

7.3.2 Array of Test Functions

Further to the example seen in Section 7.3.1, we attempt the method on a variety of different test functions, rather than simply focusing on a particular one. With this in

	n_x	n_d	Posterior samples	Time	SMSE
WGP	16	2^5	1,000	0.2324	0.0958
ITP				0.0770	0.0821
WGP			10,000	2.0340	0.3035
ITP				0.1040	0.3077
WGP		2^{12}	1,000	4.4901	19.7925
ITP				12.7708	18.6504
WGP			10,000	14.4754	20.8113
ITP				15.6968	20.8995
WGP	64	2^5	1,000	2.2104	0.1139
ITP				0.3938	0.1136
WGP			10,000	2.2414	0.1096
ITP				0.4024	0.1047
WGP		2^{12}	1,000	15.4146	10.4373
ITP				47.0927	9.3540
WGP			10,000	21.6961	10.3895
ITP				56.9586	9.6718

Table 7.1: Comparison of the results between ITP and WGP for differing numbers of inputs, samples and output dimensions.

mind, a changing test function was made to try to provide different types of test functions for the two methods. The form of the test function, which is similar to equation (7.4), is

$$\begin{aligned} f_1(x) &= \beta_1 x + \beta_2 \sin(\beta_2 x^2), \\ f_t(x) &= \beta_3 f_{t-1}(x) + \sin(t) + \beta_4 t + \beta_5 t^2 + \beta_6 \cos(x). \end{aligned} \tag{7.5}$$

To attempt to provide different challenges for the method, 1,000 test functions will be created in which, for each test function, all β coefficients will be selected at random from a distribution. Further to this, the dimensionality of the output, T, as well as the number of different inputs that we simulate from the test function, n_x, will also be drawn from a discrete random distribution. As T will be dyadic, we define $T = 2^{\beta_7}$, and the random

distribution will be assigned to β_7. The distributions are

$$\begin{aligned}
\beta_1 &\sim U(-10, 10),\\
\beta_2 &\sim U(-10, 10),\\
\beta_3 &\sim U(-1, 1),\\
\beta_4 &\sim N(0, 0.03),\\
\beta_5 &\sim U(-0.000002, 0.000002),\\
\beta_6 &\sim N(0, 3),\\
\beta_7 &\sim DU(5, 15),\\
n_x &\sim DU[10, 200],
\end{aligned}$$

where DU is the discrete uniform distribution. The beta terms have had a mixture of uniform and Gaussian distributions placed on them. The Gaussian distribution was used on β_4 to simulate situations in which we generally have a lack of linear terms, with the quadratic term often dominating the linear trend, however the standard deviation was left large enough so that we could simulate the converse situation on occasion. A Gaussian distribution was placed on β_6 such that in approximately $\frac{1}{4}$ of the cases $|\beta_6| \leq 1$, and so the $\sin$ term was more influential, with the remaining case the opposite was true. The unseen input that will be tested for each of the scenarios is $x = 0.5$. A Debauches wavelet with ten vanishing moments will be used for our wavelet decomposition for these examples.

The results of the example can be seen in Table 7.2. As stated earlier, 1,000 different scenarios were simulated, with a random draw from the parameters distribution used as inputs to the function. The results in Table 7.1 show the average of the SMSE and time taken in all of these scenarios for comparison. It can be seen that the SMSE for the two method are comparable, with the ITP method being superior in this aspect. When the results were analysed further, it was found that the WGP had a smaller SMSE than the ITP method in 32.64% of the scenarios. In each of those scenarios, the WGP was also quicker than the ITP method. This was especially prevalent when we had a large value for T, which, in terms of the speed of the method, is as we would expect due to the arguments that have been outlined earlier. This was reflected when we performed additional analysis in which the values of β_7 and n_x were fixed to be 14 and 190 respectively, the results of which can be seen in Table 7.4. When these values were fixed to be small values, $\beta_7 = 6$ and $n_x = 15$, it can be seen that the ITP method is both quicker and more accurate than WGP for these situations. This leads us to the suggestion that if we have a relatively small dimensional example, the ITP method should be used over WGP. It was also noticed in those situations that when β_4 was large, our method had a smaller SMSE than the ITP. When the converse was true, that is, when T was small and β_4 was also small, the WGP

was worse in both metrics. This is an interesting find, as it shows that in certain scenarios, the WGP is advantageous in both aspects over the ITP method (when we have a large dimensional output number and we suspect there is a reasonable sized linear term). We can see from the results that the WGP method is, on average, over four times quicker than the ITP method, which is one of the main features that we required from our method.

	Time	SMSE
WGP	28.522	8.044
ITP	130.497	7.746

Table 7.2: Showing the results of WGP and ITP using equation (7.5) with random parameter values. The SMSE reported is the mean of the 1,000 SMSE calculations.

	Time	SMSE
WGP	0.063	0.0760
ITP	0.018	0.0691

Table 7.3: Showing the results of WGP and ITP using equation (7.5) with random parameter values, but fixing $\beta_7 = 6$ and $n_x = 15$. The SMSE reported is the mean of the 1,000 SMSE calculations.

	Time	SMSE
WGP	179.117	14.072
ITP	1698.104	13.622

Table 7.4: Showing the results of WGP and ITP using equation (7.5) with random parameter values, but fixing $beta_7 = 14$ and $n_x = 190$. The SMSE reported is the mean of the 1,000 SMSE calculations.

7.3.3 Test functions with a discontinuity

Expanding on the examples seen in Section 7.3.2, we could also consider those cases in which we have a discontinuity within the test function. That is, we will again compare our method to the ITP method, using the metric of average time taken to implement the method, and SMSE. As in Section 7.3.2, 1,000 different test functions and data were simulated, with all of the parameters sampled from random distributions. The test function

for this case has the form

$$
\begin{aligned}
f_1(x) &= \beta_1 x + \beta_2 \sin(\beta_2 x^2), \\
f_t(x) &= \begin{cases} \beta_3 f_{t-1}(x) + \sin(t) + \beta_4 t + \beta_5 t^2 + \beta_6 \cos(x) + s & t = t_s, \\ \beta_3 f_{t-1}(x) + \sin(t) + \beta_4 t + \beta_5 t^2 + \beta_6 \cos(x) & \text{else.} \end{cases}
\end{aligned} \tag{7.6}
$$

We can see that this test function has a jump discontinuity of size s when $t = t_s$, hence, our smoothness property is lost for the outputs. The distributions of the parameters that will be used for our test function are as follows

$$
\begin{aligned}
\beta_1 &\sim U(-10, 10), \\
\beta_2 &\sim U(-10, 10), \\
\beta_3 &\sim U(-1, 1), \\
\beta_4 &\sim N(0, 0.03), \\
\beta_5 &\sim U(-0.000002, 0.000002), \\
\beta_6 &\sim N(0, 3), \\
\beta_7 &\sim DU(5, 13), \\
n_x &\sim DU(10, 200), \\
s &\sim N(0, 30), \\
t_s &\sim DU(2, 2^{\beta_7}).
\end{aligned}
$$

An example of one of the test functions that were used in our analysis can be seen in Figure 7.6. The results of the simulations can be seen in Table 7.5. We can see that the method performs comparably with the ITP method with regards to SMSE, with only a difference of 0.001 (rounded) between the two methods. This is encouraging and suggests that the wavelet method performs well due to its ability to capture the main features of the output vector in only a few coefficients. In most of the cases, the jump discontinuity had a considerable effect on the output of the function, and the small SMSE of our method suggests that the wavelet decomposition has modelled many of these cases well. In Table 7.5, we can also notice that the WGP, as expected, has performed the analysis in a quicker time on average than ITP. The WGP method, for this group of test functions, was shown to be approximately five times quicker than the ITP method. These simulations suggest that if we have a situation in which the output has a discontinuity, the WGP method is advantageous if time is a crucial factor, providing a comparable estimate in a much shorter time.

	Time	SMSE
WGP	15.734	1.118
ITP	78.318	1.117

Table 7.5: Showing the results of WGP and ITP using equation (7.6), with random parameter values. The SMSE and time reported is the mean of the 1,000 SMSE calculations and time taken respectively.

7.4 Conclusions

In this chapter, we have explored the use of wavelets as a dimension reduction tool for high dimensional output data. We have seen that the method performs well in terms of computing time compared to other popular methods. Throughout the chapter, we have used the Gaussian process of each of the retained coefficients in the wavelet decomposition. This was justified by the smoothness that the wavelet coefficients displayed for the observation locations. In this chapter, we have chosen to use wavelets as our tool for dimension reduction, however it should be noted that this is just a choice of basis approximation, and other methods could be used alternatively.

Not only is the method advantageous over some of the most popular method, in terms of computing time, but our method also has scope for extension to much more challenging situations. An example of this is if we believed that there was a discontinuity in the input space for some (or all) of the output time points. The Gaussian processes on the coefficients that are affected by this discontinuity could be replaced by adapted Gaussian process techniques such as the, TGP or the Voronoi Gaussian process method introduced in Chapter 5. Other methods, such as the Multi-output Gaussian process, would prove much more challenging to adapt for these situations.

For future work, as we explored in Chapter 6, a good extension to the method that could be explored is uncertainty propagation. It was seen in that chapter that a useful tool for practical purposes was the ability to not only produce an estimate for the mean of the unseen locations, but also to give insightful probability estimates and credible intervals.

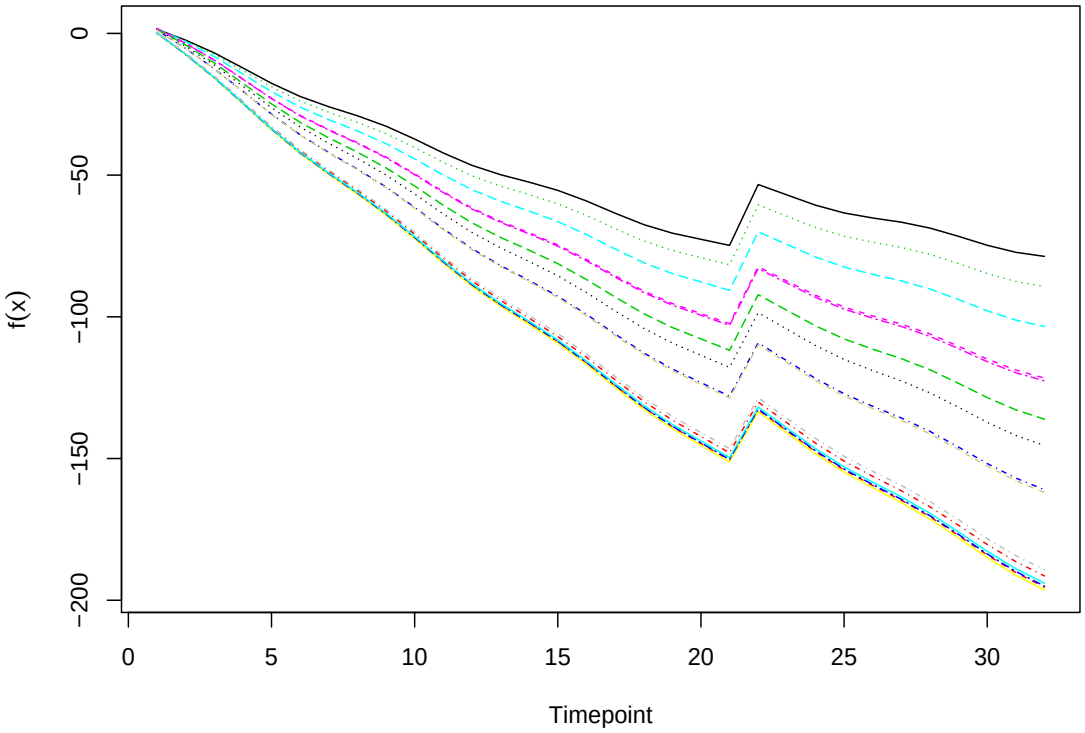


Figure 7.6: An example of a test function with a discontinuity from equation (7.6). The test function has been observed at 16 different values of x, with each colour representing an observation.

Chapter 8

Discussion

In this thesis, novel methods were introduced, and the confluence of Gaussian process emulation and wavelets was explored. Particular attention was paid to situations in which we are working with a function that contains a discontinuity. Mainly, two challenges that people are faced with were addressed in the thesis: the building of a statistical model that can be used to accurately represent our beliefs in the underlying model, and sampling further design points by utilising the information that we have from the original design points to better define the discontinuity.

In Chapter 4, we introduced a method to better define the discontinuity in the unknown function f. To achieve this, the idea was to sample new points that were close to the location of the discontinuity. In our examples, it was seen that when the initial locations of the design points were not poor, we often sampled closer to the location of the discontinuity (from above or below) than any existing design points. Sampling closer to the location of the discontinuity will give us more accurate information regarding the precise location in which this discontinuity occurs. As such, in the chapter, the metric chosen to assess the goodness of our sampling was the distance from the discontinuity.

Being able to generate samples around the location of the discontinuity allows us to gain better estimates of the features around this challenging part of space. For example, if we sample two points close together near one side of the discontinuity (in a one-dimensional sense), we will be more informed about the first derivative of the function at this location. The method seems to perform well in these aspects and samples around the location of the discontinuity. Other popular standard sampling techniques, such as selecting the location which has the largest posterior uncertainty in the Gaussian process, are not aimed for this objective and, hence, perform poorly. Using the largest posterior uncertainty location does not take fully into account the variance that we observe in the outputs at our design point locations, which is key to understanding where this discontinuity occurs. The largest uncertainty sampling method instead tends to select locations that are furthest from the existing design point, with a tendency to sample near the boundary

of the function.

Due to the neighbourhood structure induced by using the moving windowed variance, with a neighbourhood structure required due to the moving window, the method seems quite restricted to one dimension. Future work in this area would be to generalise the method. In a two-dimensional wavelet decomposition, for example, for each level of the decomposition, we have three sets of wavelet basis coefficients: two coefficients to describe the magnitude of the basis in each dimension marginally, and another to describe the joint activity (Nason 2010). There does not appear to be an obvious neighborhood structure that is comparable to the one-dimensional version described in Chapter 4.

In Chapter 5, we developed a novel method to model a function $f(\cdot)$ that may contain discontinuities. The method allows us to estimate our uncertainty in the function for both observed and unobserved input values $\mathbf{x}$. The input space of $\mathbf{x}$, $\mathcal{X}$, was partitioned into regions, with separate statistical models built in each of those regions. For our method, Voronoi tessellations are used to partition the input space. To allow more diverse shapes of regions, the tiles of the Voronoi tessellations were allowed to merge to form a region. As was explored, allowing larger regions to be formed using tiles that are created through our partitioning scheme, is advantageous in various situations, such as when one region is surrounded by another. In the chapter, another partitioning method was discussed, treed partitioning, which is used by the TGP method. Future research effort could be to explore the idea of allowing partition cells to merge to create larger regions in other partitioning methods, such as the treed partitioning, and to see what benefit this brings to the accuracy of our model.

In Chapter 5, the statistical model that was used to model each region was the Gaussian process. The decision to use the Gaussian process is due to our belief that the separate regions could be modeled as a smooth function. In a number of applications that we have explored throughout the thesis, we are limited in the number of observations that we have. In Chapter 2, it was seen that the Gaussian process prior performs well when the number of observations are limited, hence giving further justification for this model in the situations in which the number of observations are limited. It should be emphasised, however, that the decision to use a Gaussian process prior is a personal choice and the reader is free to select a model of their own choice — certainly if there is any prior knowledge that can be incorporated into this decision.

Within the prior distribution of our model in Chapter 5, a Poisson process prior is used for the locations and number of centres in the Voronoi tessellation of our partitioning. We could, however, use a more complex process than this, such as a Gibbs process (Illian et al. 2008), which includes a repulsion term. This repulsion term, as the name suggests, affects how close the centres are to each other, with a large repulsion term reducing the probability of two centres being located close to each other, compared to that of a small

repulsion term. Using a model with a repulsion term would have the benefit of additional centres having a localised effect on the model tessellation, and may help when modeling complex shapes.

One aspect of the method in Chapter 5 that could be improved upon, is the speed of our implementation. Our method's computational run time is slower than that of its main competitor, the TGP model. This computational time could be reduced by using a more powerful coding platform than R. Another strategy that could be implemented to decrease the computational time is to use a proper prior distribution for the smoothness parameter $\mathbf{b}$ and include this in our RJMCMC.

Being able to report, and visualise, the results of your statistical method in a clear and coherent manor is a critical step in any statistical analysis. One particular feature in our method that would be of interest to a model builder is the location of the discontinuity in the function. The shape, size, and location of the discontinuity could aid an expert/the model builder in their understanding as to why these discontinuities occur. As we saw in Section 5.6, this is easy to visualise and report in two dimensions. This visualisation, however, becomes much more difficult when the number of input dimensions become larger. One example of how to attempt to visualise the shape and location of the regions was shown in Figure 5.20. Future research effort could be to find, or create, new visualiation techniques so that these regions can be reported and understood more clearly.

In Section 5.5, a novel method was introduced that looks to sample new design points on the boundary of a discontinuity. The method looks at our posterior beliefs, selects the most probable model, and then looks to sample on the boundary of the region of interest. It was shown that the sampling method performed well when compared to alternative popular sampling methods. This is due to the new samples that were selected lying close to the location of the discontinuity. This area of space was often where our squared error for prediction was the largest, and, by gaining more information around these regions, as opposed to the areas where the squared error was comparatively small, we were able to model the function as a whole better. The alternative methods were not geared towards attempting to sample near this challenging part of space in particular.

In Chapter 6, a novel technique was developed that looks to quantify our uncertainty in a function $f(\cdot)$ that has a one-dimensional input. Wavelet shrinkage is a tool that can be used in these situations, but, there are multiple considerations that must be addressed. If we do not have a data vector, which is obtained by observing the function at selected locations, that is either periodic or symmetric, we will have problems with coefficients on the boundaries. In our method, to handle this, the function was split into two parts, with a long term trend function and a wavelet trend function used. By assigning a prior distribution, the Gaussian process, to the long term trend, and also considering the uncertainty that we have from the wavelet shrinkage, which was used for the wavelet trend function,

we were able to find our posterior uncertainty about the full underlying function.

This method was novel due to the uncertainty that is being taken into consideration. Previous methods that have looked at similar problems either: considered the case in which the underlying function was periodic/symmetric, and, hence, had no boundary problems (Semadeni et al. 2004), or assumed that there was no uncertainty in the long term trend function and did not consider at the full distribution of the wavelet function (Lee & Oh 2004). Neither of these methods considered the use of a Gaussian process within their respective methods.

The method introduced provides a tool for modelling a one-dimension function $f(\cdot)$ which may contain a discontinuity in the input space. We utilise the power of the Gaussian process, with regards to its ability to represent a smooth function well, as we expect the long term trend of the function to be smooth, and combine this with the ability of the wavelet shrinkage to model a function with a discontinuity. It was seen in our examples that the method was also advantageous in terms of prediction over each of the methods used individually. We also explored a possible adaptation to the method as we looked at using a linear function for our long term trend, much as was seen in Lee & Oh (2004), and examined the gain that was made by considering our uncertainty that should be inherent in this function.

In Chapter 6, we displayed the results using both a Gaussian and a Laplace distribution for the non-zero PDF in our mixture prior. This allowed us to examine the effect that our beliefs in the prior distributions of the wavelet coefficients has on our posterior inference. We saw that the Laplace distribution's tails decayed much more slowly than those of the Gaussian, and, hence, created much wider tails in our posterior distributions of the wavelet coefficients. Due to this, when the Laplace distribution was used, wider credible intervals were observed in our posterior estimate of the function f. Future research efforts could be directed towards looking at further changes of prior distribution for the wavelet coefficients, such as the use of the Cauchy PDF in our mixture prior, as was suggested by Johnstone & Silverman (2005).

In Chapter 7, a method was developed to allow us to model a function $f(\cdot)$ that produces a large number of outputs for a single input $\mathbf{x}$. As was discussed, popular methods that are used for multiple output functions are either extremely inefficient, as we saw for the TAI model, or were too complex, as we saw for MOE. The inefficiency of TAI model was due to the extremely large matrix A that not only needed to be stored, but also be inverted. The MOE method suffers due to its complexity of implementation, certainly when attempting to consider the correlations across all of the inputs and outputs.

Our method instead builds upon the idea of using an independent Gaussian process on each output point. When the number of output dimensions is large, this can be a heavy task, in terms of computation time. To reduce the computational burden, a wavelet

decomposition is used, with a subset of these coefficients set to zero, and the remaining coefficients retained for our statistical analysis. It was seen that a data vector could be approximated well by a few wavelet coefficients in a DWT. We also saw in the examples throughout the chapter, when there was smoothness in the test function, the coefficients also displayed smoothness, and, hence, a Gaussian process was appropriate.

When considering our method, it can be advantageous over other methods, such as the MOE, when we have a test function which may contain a discontinuity in the input space. Our method allows for simple and natural adaptation, in that we could easily use the Voronoi tessellation Gaussian process method discussed in Chapter 5, or, similarly, the TGP methodology in Gramacy & Lee (2008), in place of the Gaussian process for our coefficients. This would allow for discontinuities in the input space, and, hence, provide us with a method of modeling functions that contain a discontinuity in the input space and also produces a high-dimensional output. Of course, the ITP and TIA models could also be adapted to account for this type of discontinuity, however, we can again reduce the computational burden that was explored in Chapter 7 by utilising the wavelet decomposition.

One restriction of the method introduced in Chapter 7 is the data structure required to implement the methodology. To perform the DWT descried, we require the data to be equispaced and dyadic in number. The main restriction out of these two requirements is the dyadic number of points, with many situations arising in practice in which this is not the case. One element of improvement could be to investigate different basis functions for the data vectors that are more adaptable with regards to the number of datapoints required, and to see how well the method performs with this adaptation. A suggested direction for this basis would be the use of lifting as opposed to the wavelet decomposition, due to its flexibility with the number of datapoints, with the number not required to be dyadic, compared to the wavelet methodology (Sweldens 1998). Other dimension reduction techniques could also be used rather than the use of wavelets, for example, the principal components of the data could also be used in a similar manner to our method (e.g. Chatfied & Collins 2013). There could also be future research effort into finding a more rigorous method of selecting the optimal number of both vanishing moments for our wavelet decomposition, and the proportion of wavelet coefficients that are retained and used in the analysis.

Appendix A

Appendix

A.1 Derivations of densities in Chapter 6

A.1.1 Using the Gaussian PDF for the symmetric part of mixture

Using a Gaussian PDF for the non-zero symmetric mixture of our prior distribution gives us

$$\pi(d_{jk}) = (1-\omega_j)\delta(0) + \omega_j N(d_{jk}; 0, a_j^2), \tag{A.1}$$

where $N(\cdot; \mu, \sigma^2)$ is the Gaussian PDF with mean μ and variance σ^2.

Using Bayes rule with equations 6.5 and A.1, we can see that the posterior distribution will have the form

$$\begin{aligned} d_{jk}|d^*_{jk} = \int_{-\infty}^{\infty} \omega_j \frac{1}{\sqrt{2\pi a_j^2}} \exp\left\{-\frac{d_{jk}^2}{2a_j^2}\right\} \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{(d^*_{jk}-d_{jk})^2}{2\zeta^2}\right\} \\ + (1-\omega_j)\frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{d^{*2}_{jk}}{2\zeta^2}\right\} \delta(0) dd^*_{jk}. \end{aligned} \tag{A.2}$$

It is easy to see that the posterior distribution will also be a mixture distribution. To calculate the normalising constant, we can consider the two parts separately due to the linearity of integrals. The first part of the integral gives us the following

$$
\begin{aligned}
&\int_{-\infty}^{\infty} \omega_j \frac{1}{2\pi}(a_j^2\zeta^2)^{-\frac{1}{2}} \exp\left\{-\frac{d_{jk}^2}{2a_j^2} - \frac{(d_{jk}^* - d_{jk})^2}{2\zeta^2}\right\} dd_{jk} \\
=&\int_{-\infty}^{\infty} \omega_j \frac{1}{2\pi}(a_j^2\zeta^2)^{-\frac{1}{2}} \exp\left\{-\frac{d_{jk}^2}{2a_j^2} - \frac{d_{jk}^{*2} + 2d_{jk}d_{jk}^* - d_{jk}^2}{2\zeta^2}\right\} dd_{jk} \\
=&\omega_j \frac{1}{2\pi}(a_j^2\zeta^2)^{-\frac{1}{2}} \exp\left\{-\frac{d_{jk}^{*2}}{2\zeta^2}\right\} \int_{-\infty}^{\infty} \exp\left\{-\frac{\zeta^2 d_{jk}^2 + 2a_j^2 d_{jk}d_{jk}^* - a_j^2 d_{jk}^2}{2\zeta^2 a_j^2}\right\} dd_{jk} \\
=&\omega_j \frac{1}{2\pi}(a_j^2\zeta^2)^{-\frac{1}{2}} \exp\left\{-\frac{d_{jk}^{*2}}{2\zeta^2}\right\} \int_{-\infty}^{\infty} \exp\left\{-\frac{1}{2a_j^2\zeta^2}\left[(\zeta^2 + a_j^2)d_{jk}^2 - 2a_j^2 d_{jk}d_{jk}^*\right]\right\} dd_{jk} \\
=&\omega_j \frac{1}{2\pi}(a_j^2\zeta^2)^{-\frac{1}{2}} \exp\left\{-\frac{d_{jk}^{*2}}{2\zeta^2}\right\} \int_{-\infty}^{\infty} \exp\left\{-\frac{(\zeta^2 + a_j^2)}{2a_j^2\zeta^2}\left[d_{jk}^2 - \frac{2a_j^2 d_{jk}d_{jk}^*}{(\zeta^2 + a_j^2)}\right]\right\} dd_{jk} \\
=&\omega_j \frac{1}{2\pi}(a_j^2\zeta^2)^{-\frac{1}{2}} \exp\left\{-\frac{d_{jk}^{*2}}{2\zeta^2}\right\} \int_{-\infty}^{\infty} \exp\left\{-\frac{(\zeta^2 + a_j^2)}{2a_j^2\zeta^2}\left[\left(d_{jk} - \frac{a_j^2 d_{jk}d_{jk}^*}{(\zeta^2 + a_j^2)}\right)^2 - \left(\frac{a_j^2 d_{jk}d_{jk}^*}{(\zeta^2 + a_j^2)}\right)^2\right]\right\} dd_{jk} \\
=&\omega_j \frac{1}{\sqrt{2\pi}}(a_j^2\zeta^2)^{-\frac{1}{2}} \exp\left\{-\frac{d_{jk}^{*2}}{2\zeta^2}\right\} \exp\left\{\frac{a_j^2 d_{jk}d_{jk}^{*2}}{2\zeta^2(\zeta^2 + a_j^2)}\right\} \left(\frac{(\zeta^2 + a_j^2)}{a_j^2\zeta^2}\right)^{-\frac{1}{2}} \\
&\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \left(\frac{a_j^2\zeta^2}{(a_j^2 + \zeta^2)}\right)^{-\frac{1}{2}} \exp\left\{-\frac{(\zeta^2 + a_j^2)}{2a_j^2\zeta^2}\left[d_{jk} - \frac{a_j^2 d_{jk}d_{jk}^*}{(\zeta^2 + a_j^2)}\right]^2\right\} dd_{jk} \\
=&\frac{\omega_j}{\sqrt{2\pi}}(a_j^2\zeta^2)^{-\frac{1}{2}} \left(\frac{(\zeta^2 + a_j^2)}{a_j^2\zeta^2}\right)^{-\frac{1}{2}} \exp\left\{\frac{a_j^2 d_{jk}d_{jk}^{*2}}{2\zeta^2(\zeta^2 + a_j^2)} - \frac{d_{jk}^{*2}}{2\zeta^2}\right\} \\
=&\frac{\omega_j}{\sqrt{2\pi}}(\zeta^2 + a_j^2)^{-\frac{1}{2}} \exp\left\{\frac{d_{jk}^{*2}}{2\zeta^2}\left(\frac{a_j^2}{(a_j^2 + \zeta^2)} - 1\right)\right\}.
\end{aligned}
$$

The second part is much easier to calculate due to the point mass.

$$
\begin{aligned}
&\int_{-\infty}^{\infty} (1 - \omega_j)\delta(0) \frac{1}{\sqrt{2\pi\zeta^2}} \exp\{-\frac{d_{jk}^{*2}}{2\zeta^2}\} dd_{jk} \\
=&(1 - \omega_j) \frac{1}{\sqrt{2\pi\zeta^2}} \exp\{-\frac{d_{jk}^{*2}}{2\zeta^2}\} \int_{-\infty}^{\infty} \delta(0) dd_{jk} \\
=&(1 - \omega_j) \frac{1}{\sqrt{2\pi\zeta^2}} \exp\{-\frac{d_{jk}^{*2}}{2\zeta^2}\}.
\end{aligned}
$$

We now have a form for the posterior distribution for the coefficient d_{jk}, which is

$$
d_{jk}|d_{jk}^* \sim \omega_j^* N\left(\frac{a_j^2 d_{jk}^*}{a_j^2 + \zeta^2}, \frac{\zeta^2 a_j^2}{\zeta^2 + a_j^2}\right) + (1 - \omega_j^*)\delta(0),
$$

where

$$
\frac{(1 - \omega_j^*)}{\omega_j^*} = \frac{(1 - \omega_j)}{\omega_j} \frac{(a_j^2 + \zeta^2)^{\frac{1}{2}}}{\zeta^2} \exp\left\{-\frac{a_j^2 d_{jk}^{*2}}{2\zeta^2(\zeta^2 + a_j^2)}\right\}.
$$

To sample from the posterior distribution of the coefficient, we use a random sampling method such that we draw a random variable from the distribution $N\left(d_{jk}; \frac{a_j^2 d_{jk}^*}{a_j^2 + \zeta^2}, \frac{\zeta^2 a_j^2}{\zeta^2 + a_j^2}\right)$ with probability ω_j^*, and zero with probability $(1 - \omega_j^*)$.

A.1.2 Using the Laplace PDF for the symmetric part of mixture

For the case in which we use equation (6.4), our prior is

$$\pi(d_{jk}) = (1-\omega_j)\delta(0) + \omega_j L(d_{jk}; 0, a_j), \tag{A.3}$$

where $L(\cdot; \mu, a)$ is the Laplace PDF from equation (6.4) with parameters μ and a.

Using Bayes rule with equations (6.5) and (A.3), it is easy to write that the posterior distribution will have the form

$$\begin{aligned} d_{jk}|d^*_{jk} = \int_{-\infty}^{\infty} \omega_j \frac{1}{2a_j} \exp\left\{-\frac{|d_{jk}|}{a_j}\right\} \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{(d^*_{jk}-d_{jk})^2}{2\zeta^2}\right\} \\ + (1-\omega_j)\frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{d^{*2}_{jk}}{2\zeta^2}\right\} \delta(0) dd_{jk}. \end{aligned} \tag{A.4}$$

We again need to find the normalising constant of this integral, and we can do so by, similar to the proof in Section A.1.1, using the linearity of the integral to split equation (A.4) into two. Firstly, we start with the start with the second part of the equation as this contains a point mass, and, as such, the calculations will be short. We find

$$\begin{aligned} &\int_{-\infty}^{\infty} (1-\omega_j)\frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{d^{*2}_{jk}}{2\zeta^2}\right\} \delta(0) dd_{jk} \\ &= (1-\omega_j)\frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{d^{*2}_{jk}}{2\zeta^2}\right\}. \end{aligned} \tag{A.5}$$

Now, the first part of equation (A.4) is more complicated and will need to be tackled in two parts due to the two cases of in the prior in equation (6.4). The two parts will be when $d_{jk} < 0$ and when $d_{jk} \geq 0$. The former was tackled first

$$\begin{aligned} &\int_{-\infty}^{0} \omega_j \frac{1}{2a} \exp\left\{\frac{d_{jk}}{a}\right\} \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{(d^*_{jk}-d_{jk})^2}{2\zeta^2}\right\} dd_{jk} \\ &= \int_{-\infty}^{0} \frac{\omega_j}{2a} \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{\frac{2\zeta^2 d_{jk}}{2\zeta^2 a} - \frac{a}{2\zeta^2 a}\left[d^{*2}_{jk} - 2d_{jk}d^*_{jk} + d^2_{jk}\right]\right\} dd_{jk} \\ &= \int_{-\infty}^{0} \frac{\omega_j}{2a} \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{1}{2\zeta^2 a}\left[ad^{*2}_{jk} - 2d_{jk}(ad^*_{jk}+\zeta^2) + ad^2_{jk}\right]\right\} dd_{jk} \\ &= \int_{-\infty}^{0} \frac{\omega_j}{2a} \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{a}{2\zeta^2 a}\left[d^{*2}_{jk} - 2d_{jk}\left(\frac{ad^*_{jk}+\zeta^2}{a}\right) + d^2_{jk}\right]\right\} dd_{jk} \\ &= \int_{-\infty}^{0} \frac{\omega_j}{2a} \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{a}{2\zeta^2 a}\left[\left(d_{jk} - \left(\frac{ad^*_{jk}+\zeta^2}{a}\right)\right)^2 - \left(\frac{ad^*_{jk}+\zeta^2}{a}\right)^2 + d^{*2}_{jk}\right]\right\} dd_{jk} \\ &= \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{1}{2\zeta^2}\left(d_{jk} - \left(\frac{ad^*_{jk}+\zeta^2}{a}\right)\right)^2\right\} dd_{jk} \frac{\omega_j}{2a} \exp\left\{-\frac{1}{2\zeta^2}\left[d^{*2}_{jk} - \left(\frac{ad^*_{jk}+\zeta^2}{a}\right)^2\right]\right\} \\ &= \Phi\left(\frac{R^-}{\zeta}\right) \frac{\omega_j}{2a} \exp\left\{-\frac{1}{2\zeta^2}\left[d^{*2}_{jk} - \left(\frac{ad^*_{jk}+\zeta^2}{a}\right)^2\right]\right\}, \end{aligned}$$

where Φ is the CDF of the standard normal distribution and $R^- = \frac{ad_{jk}-\zeta^2}{a}$.

For the case when $d_{jk} \geq 0$, we get

$$
\begin{aligned}
&\int_0^\infty \omega_j \frac{1}{2a} \exp\left\{-\frac{d_{jk}}{a}\right\} \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{(d^*_{jk}-d_{jk})^2}{2\zeta^2}\right\} dd_{jk} \\
&= \int_{-\infty}^0 \frac{\omega_j}{2a} \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{2\zeta^2 d_{jk}}{2\zeta^2 a} - \frac{a}{2\zeta^2 a}\left[d^{*2}_{jk} - 2d_{jk}d^*_{jk} + d^2_{jk}\right]\right\} dd_{jk} \\
&= \int_{-\infty}^0 \frac{\omega_j}{2a} \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{1}{2\zeta^2 a}\left[ad^{*2}_{jk} - 2d_{jk}(ad^*_{jk}-\zeta^2) + ad^2_{jk}\right]\right\} dd_{jk} \\
&= \int_{-\infty}^0 \frac{\omega_j}{2a} \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{a}{2\zeta^2 a}\left[d^{*2}_{jk} - 2d_{jk}\left(\frac{ad^*_{jk}-\zeta^2}{a}\right) + d^2_{jk}\right]\right\} dd_{jk} \\
&= \int_{-\infty}^0 \frac{\omega_j}{2a} \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{a}{2\zeta^2 a}\left[\left(d_{jk} - \left(\frac{ad^*_{jk}-\zeta^2}{a}\right)\right)^2 - \left(\frac{ad^*_{jk}-\zeta^2}{a}\right)^2 + d^{*2}_{jk}\right]\right\} dd_{jk} \\
&= \int_{-\infty}^0 \frac{1}{\sqrt{2\pi\zeta^2}} \exp\left\{-\frac{1}{2\zeta^2}\left(d_{jk} - \left(\frac{ad^*_{jk}-\zeta^2}{a}\right)\right)^2\right\} dd_{jk} \frac{\omega_j}{2a} \exp\left\{-\frac{1}{2\zeta^2}\left[d^{*2}_{jk} - \left(\frac{ad^*_{jk}-\zeta^2}{a}\right)^2\right]\right\} \\
&= \Phi\left(\frac{R^+}{\zeta}\right) \frac{\omega_j}{2a} \exp\left\{-\frac{1}{2\zeta^2}\left[d^{*2}_{jk} - \left(\frac{ad^*_{jk}+\zeta^2}{a}\right)^2\right]\right\},
\end{aligned}
$$

where $R^+ = \frac{ad^*_{jk}-\zeta^2}{a}$.

Joining the two ω_j parts together and simplifying, we get

$$\frac{\omega_j}{2a} \exp\left\{\frac{\zeta^3}{2a^2}\right\}\left[\Phi\left(\frac{R^+}{\zeta}\right)\exp\left\{\frac{d^*}{a}\right\} + \Phi\left(\frac{R^-}{\zeta}\right)\exp\left\{\frac{-d^*}{a}\right\}\right]$$

This leaves us with a final odds ratio of

$$\frac{1-\omega^*_j}{\omega^*_j} = \frac{2(1-\omega_j)a}{\omega_j\sqrt{2\pi\zeta^2}} \exp\left\{-\left[\frac{d^*2_{jk}}{2\zeta^2} + \frac{\zeta^3}{2a}\right]\right\}\left[\Phi\left(\frac{R^+}{\zeta}\right)\exp\left\{\frac{d^*}{a}\right\} + \Phi\left(\frac{R^-}{\zeta}\right)\exp\left\{\frac{-d^*}{a}\right\}\right],$$

where ω^*_j denotes the posterior weight.

To sample from the posterior distribution of this wavelet coefficient, we must find a method to sample from the non-zero mixture element, which we are able to do using inverse sampling (Voss 2013). We can calculate the CDF of the posterior Laplace set-up using the equation seen in Johnstone & Silverman (2005), and this gives us

$$F_{post}(\theta_{jk}) = \begin{cases} \frac{\exp(a_j d^*_{jk})(1-\Phi(\theta_{jk}-d^*_{jk}-a_j))}{\exp^{-a_j d^*_{jk}}\Phi(d^*_{jk}-a_j)+\exp(a_j d^*_{jk})(1-\phi(d^*_{jk}+a_j)} & \theta_{jk} \leq 0, \\ \frac{\exp(-a_j d^*_{jk})(1-\Phi(\theta_{jk}-d^*_{jk}+a_j))}{\exp^{-a_j d^*_{jk}}\Phi(d^*_{jk}-a_j)+\exp(a_j d^*_{jk})(1-\phi(d^*_{jk}+a_j)} & \text{else,} \end{cases}$$

If we replace $F_{post}(\theta_{jk})$ with a number $u^* \in [0,1]$, knowing the inverse cumulative function, we can use the relation $F^{-1}(u^*) = x$ to make random draws from the required distribution. We do this by making random draws from the uniform distribution, and

transform these into random draws from the required distribution by inputting them into the inverse cumulative function. If this is done, we find that

$$\theta_{jk} = \begin{cases} \Phi^{-1}\left[1 - u^* e^{ad}\left[e^{-ad}\Phi(d-a) + e^{ad}(1-\Phi(d+a))\right]\right] + d - a \text{ if } u^* \geq z, \\ \Phi^{-1}\left[1 - u^* e^{-ad}\left[e^{-ad}\Phi(d-a) + e^{ad}(1-\Phi(d+a))\right]\right] + d + a \text{ else,} \end{cases} \tag{A.6}$$

where $z = \frac{\exp(a_j d^*_{jk})(1-\Phi(\theta_{jk}-d^*_{jk}-a_j))}{\exp^{-a_j d^*_{jk}}\Phi(d^*_{jk}-a_j)+\exp(a_j d^*_{jk})(1-\phi(d^*_{jk}+a_j)}$.

Using equation (A.6) allows us to make exact samples from the symmetric part of this posterior distribution. To gain a sample from the posterior distribution of the wavelet coefficient d_{jk}, similar to Section A.1.1, we draw a value u from the uniform distribution, and sample from the posterior symmetric distribution if $u < \omega^*_j$, and our sample zero if $u \geq \omega^*_j$.

Bibliography

Abramovich, F., Sapatinas, T. & Silverman, B. W. (1998), 'Wavelet thresholding via a Bayesian approach', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **60**(4), 725–749.

Aigner, M. (1999), 'A characterization of the Bell numbers', *Discrete Mathematics* **205**(1), 207–210.

Alley, R. B., Marotzke, J., Nordhaus, W. D., Overpeck, J. T., Peteet, D. M., Pielke, R. A., Pierrehumbert, R., Rhines, P., Stocker, T., Talley, L. et al. (2003), 'Abrupt climate change', *Science* **299**(5615), 2005–2010.

Andrews, L. C. & Andrews, L. C. (1992), *Special functions of mathematics for engineers*, McGraw-Hill New York.

Barber, S. & Nason, G. P. (2003), 'Simulations comparing thresholding methods using real and complex wavelets', *Research Report* **3**, 13.

Barber, S., Nason, G. P. & Silverman, B. W. (2002), 'Posterior probability intervals for wavelet thresholding', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **64**(2), 189–205.

Bastos, L. S. & O'Hagan, A. (2009), 'Diagnostics for Gaussian process emulators', *Technometrics* **51**(4), 425–438.

Chatfied, C. & Collins, A. (2013), *Introduction to Multivariate Analysis*, Springer.

Chatfield, C. (2016), *The Analysis of Time Series: An Introduction*, sixth edn, CRC press.

Chen, V. C., Tsui, K.-L., Barton, R. R. & Meckesheimer, M. (2006), 'A review on design, modeling and applications of computer experiments', *IIE transactions* **38**(4), 273–291.

Chipman, H. A., Kolaczyk, E. D. & McCulloch, R. E. (1997), 'Adaptive Bayesian wavelet shrinkage', *Journal of the American Statistical Association* **92**(440), 1413–1421.

Cleveland, W. S., Grosse, E. & Shyu, W. M. (1992), 'Local regression models', *Statistical Models in S* **2**, 309–376.

Cohen, A., Daubechies, I. & Vial, P. (1993), 'Wavelets on the interval and fast wavelet transforms', *Applied and Computational Harmonic Analysis* .

Conti, S., Gosling, J. P., Oakley, J. E. & O'Hagan, A. (2009), 'Gaussian process emulation of dynamic computer codes', *Biometrika* **96**(3), 663–676.

Conti, S. & O'Hagan, A. (2010), 'Bayesian emulation of complex multi-output and dynamic computer models', *Journal of Statistical Planning and Inference* **140**(3), 640–651.

Cressie, N. (1993), *Statistics for Spatial Data (rev.ed.)*, New York: Wiley.

Daubechies, I. (1988), 'Orthonormal bases of compactly supported wavelets', *Communications on Pure and Applied Mathematics* **41**(7), 909–996.

Daubechies, I. (1992), *Ten Lectures on Wavelets*, Vol. 61, SIAM.

Davison, A. & Mastropietro, D. (2009), 'Saddlepoint approximation for mixture models', *Biometrika* **96**(2), 479–486.

De Boor, C., De Boor, C., Mathématicien, E.-U., De Boor, C. & De Boor, C. (1978), *A Practical Guide to Splines*, Vol. 27, Springer-Verlag New York.

Donoho, D. L. (1993), 'Unconditional bases are optimal bases for data compression and for statistical estimation', *Applied and Computational Harmonic Analysis* **1**(1), 100–115.

Donoho, D. L. & Johnstone, I. M. (1995), 'Adapting to unknown smoothness via wavelet shrinkage', *Journal of the American Statistical Association* **90**(432), 1200–1224.

Donoho, D. L. & Johnstone, J. M. (1994), 'Ideal spatial adaptation by wavelet shrinkage', *Biometrika* **81**(3), 425–455.

Feingold, G., McComiskey, A., Yamaguchi, T., Johnson, J. S., Carslaw, K. S. & Schmidt, K. S. (2016), 'New approaches to quantifying aerosol influence on the cloud radiative effect', *Proceedings of the National Academy of Sciences* **113**(21), 5812–5819.

Fricker, T. E., Oakley, J. E. & Urban, N. M. (2013), 'Multivariate Gaussian process emulators with nonseparable covariance structures', *Technometrics* **55**(1), 47–56.

Gallier, J. (2008), 'Notes on convex sets, polytopes, polyhedra, combinatorial topology, Voronoi diagrams and Delaunay triangulations', *arXiv preprint arXiv:0805.0292* .

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. & Rubin, D. B. (2013), *Bayesian Data Analysis*, CRC Press.

Giunta, A., Wojtkiewicz, S. & Eldred, M. (2003), Overview of modern design of experiments methods for computational simulations, *in* '41st Aerospace Sciences Meeting and Exhibit', p. 649.

Gramacy, R. B. & Lee, H. K. H. (2008), 'Bayesian treed Gaussian process models with an application to computer modeling', *Journal of the American Statistical Association* **103**(483), 1119–1130.

Green, P. J. (1995), 'Reversible jump Markov chain Monte Carlo computation and Bayesian model determination', *Biometrika* pp. 711–732.

Haar, A. (1910), 'Zur theorie der orthogonalen funktionensysteme', *Mathematische Annalen* **69**(3), 331–371.

Haylock, R. (1997), Bayesian inference about outputs of computationally expensive algorithms with uncertainty on the inputs, PhD thesis, Department of Statistics, University of Nottingham.

Heaton, T. & Silverman, B. (2008), 'A wavelet- or lifting-scheme-based imputation method', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **70**(3), 567–587.

Higdon, D., Gattiker, J., Williams, B. & Rightley, M. (2008), 'Computer model calibration using high-dimensional output', *Journal of the American Statistical Association* **103**(482), 570–583.

Illian, J., Penttinen, A., Stoyan, H. & Stoyan, D. (2008), *Statistical Analysis and Modelling of Spatial Point Patterns*, Chichester: John Wiley & Sons.

Jaffard, S., Meyer, Y. & Ryan, R. D. (2001), *Wavelets: Tools for Science and Technology*, Vol. 69, SIAM.

Johnson, M., Moore, L. & Ylvisaker, D. (1990), 'Minimax and maximin distance designs', *Journal of Statistical Planning and Inference* **26**(2), 131 – 148.

Johnstone, I. M. & Silverman, B. W. (1997), 'Wavelet threshold estimators for data with correlated noise', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **59**(2), 319–351.

Johnstone, I. M. & Silverman, B. W. (2005), 'Ebayesthresh: R and s-plus programs for empirical Bayes thresholding', *Journal of Statistical Software* **12**, 1–38.

Johnstone, I. M., Silverman, B. W. et al. (2004), 'Needles and straw in haystacks: Empirical Bayes estimates of possibly sparse sequences', *The Annals of Statistics* **32**(4), 1594–1649.

Khairoutdinov, M. F. & Randall, D. A. (2003), 'Cloud resolving modeling of the arm summer 1997 iop: Model formulation, results, uncertainties, and sensitivities', *Journal of the Atmospheric Sciences* **60**(4), 607–625.

Kim, H., Mallick, B. & Holmes, C. (2005), 'Analyzing nonstationary spatial data using piecewise Gaussian processes', *Journal of the American Statistical Association* **100**, 653–668.

Lee, T. C. & Oh, H.-S. (2004), 'Automatic polynomial wavelet regression', *Statistics and Computing* **14**(4), 337–341.

Mallat, S. (1999), *A wavelet tour of signal processing*, Elsevier.

Mallat, S. G. (1989), 'A theory for multiresolution signal decomposition: the wavelet representation', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**(7), 674–693.

Mardia, K., Kent, J. & Bibby, J. (1979), *Multivariate Analysis*, Vol. 1, Academic Press.

Misiti, M., Misiti, Y., Oppenheim, G. & Poggi, J.-M. (2013), *Wavelets and their Applications*, John Wiley & Sons.

Montgomery, D. C., Peck, E. A. & Vining, G. G. (2012), *Introduction to Linear Regression Analysis*, Vol. 821, John Wiley & Sons.

Nason, G. (2010), *Wavelet Methods in Statistics with R*, Springer Science & Business Media.

Nason, G. P. & Silverman, B. W. (1994), 'The discrete wavelet transform in s', *Journal of Computational and Graphical Statistics* **3**(2), 163–191.

Nason, G. P., Von Sachs, R. & Kroisandt, G. (2000), 'Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **62**(2), 271–292.

National Atmospheric Deposition Program (2007), 'Ammonia monitering network', `http://nadp.sws.uiuc.edu/data/AMoN/`. Accessed: 2017-06-10.

Nunes, M. A., Knight, M. I. & Nason, G. P. (2006), 'Adaptive lifting for nonparametric regression', *Statistics and Computing* **16**(2), 143–159.

Oakley, J. & O'Hagan, A. (2002), 'Bayesian inference for the uncertainty distribution of computer model outputs', *Biometrika* **89**, 769–784.

Ogden, T. (2012), *Essential Wavelets for Statistical Applications and Data Analysis*, Springer Science & Business Media.

Oh, H.-S. & Kim, H.-M. (2008), 'Bayesian automatic polynomial wavelet regression', *Journal of Statistical Planning and Inference* **138**(8), 2303–2312.

Oh, H.-S. & Lee, T. C. (2005), 'Hybrid local polynomial wavelet shrinkage: wavelet regression with automatic boundary adjustment', *Computational Statistics & Data Analysis* **48**(4), 809–819.

O'Hagan, A. & Forster, J. J. (2004), *Kendall's Advanced Theory of Statistics, volume 2B: Bayesian Inference*, Vol. 2, Arnold.

Okabe, A., Boots, B., Sugihara, K. & Chui, S. (2000), *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, New York: Wiley.

Paciorek, C. J. & Schervish, M. J. (2006), 'Spatial modelling using a new class of nonstationary covariance functions', *Environmetrics* **17**(5), 483–506.

Park, J.-S. (1994), 'Optimal latin-hypercube designs for computer experiments', *Journal of Statistical Planning and Inference* **39**(1), 95–111.

Peck, S. J. (2010), Multiscale spatial imputation applied to crop infestation modelling, PhD thesis, University of Leeds.

Percival, D. B. & Walden, A. T. (2006), *Wavelet methods for time series analysis*, Vol. 4, Cambridge university press.

Pronzato, L. & Müller, W. G. (2012), 'Design of computer experiments: space filling and beyond', *Statistics and Computing* **22**(3), 681–701.

Quinlan, J. R. (1986), 'Induction of decision trees', *Machine Learning* **1**(1), 81–106.

Rasmussen, C. (1996), Evaluation of Gaussian processes and other methods for non-linear regression, PhD thesis, Graduate Department of Computer Science, University of Toronto.

Rasmussen, C. E. & Williams, C. K. I. (2006), *Gaussian Processes for Machine Learning*, The MIT Press.

Risser, M. D. (2016), 'Review: Nonstationary spatial modeling, with emphasis on process convolution and covariate-driven approaches', *arXiv preprint arXiv:1610.02447* .

Risser, M. D. & Calder, C. A. (2015*a*), 'Local likelihood estimation for covariance functions with spatially-varying parameters: the convospat package for r', *arXiv preprint arXiv:1507.08613* .

Risser, M. D. & Calder, C. A. (2015*b*), 'Regression-based covariance functions for non-stationary spatial modeling', *Environmetrics* **26**(4), 284–297.

Rothwell, E., Chen, K., Nyquist, D., Ross, J. & Bebermeyer, R. (1994), 'A radar target discrimination scheme using the discrete wavelet transform for reduced data storage', *IEEE Transactions on Antennas and Propagation* **42**(7), 1033–1037.

Sacks, J., Welch, W. J., Mitchell, T. J. & Wynn, H. P. (1989), 'Design and analysis of computer experiments', *Statistical science* pp. 409–423.

Santner, T., Williams, B. & Notz, W. (2003), *The Design and Analysis of Computer Experiments*, New York: Springer.

Schmidt, A. M. & O'Hagan, A. (2003), 'Bayesian inference for non-stationary spatial covariance structure via spatial deformations', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **65**(3), 743–758.

Semadeni, C., Davison, A. & Hinkley, D. (2004), 'Posterior probability intervals in bayesian wavelet estimation', *Biometrika* pp. 497–505.

Shewry, M. C. & Wynn, H. P. (1987), 'Maximum entropy sampling', *Journal of applied statistics* **14**(2), 165–170.

Stanimirović, P. S. & Petković, M. D. (2013), 'Gauss–Jordan elimination method for computing outer inverses', *Applied Mathematics and Computation* **219**(9), 4667–4679.

Stein, C. M. (1981), 'Estimation of the mean of a multivariate normal distribution', *The annals of Statistics* pp. 1135–1151.

Stein, E. M. & Weiss, G. (2016), *Introduction to Fourier analysis on Euclidean spaces (PMS-32)*, Vol. 32, Princeton university press.

Sweldens, W. (1998), 'The lifting scheme: A construction of second generation wavelets', *SIAM journal on mathematical analysis* **29**(2), 511–546.

Vidakovic, B. (1998), 'Nonlinear wavelet shrinkage with Bayes rules and Bayes factors', *Journal of the American Statistical Association* **93**(441), 173–179.

Voss, J. (2013), *An Introduction to Statistical Computing: A Simulation-Based Approach*, John Wiley & Sons.

Walter, G. G. & Shen, X. (2018), *Wavelets and other orthogonal systems*, CRC press.

Weakliem, D. L. (1999), 'A critique of the bayesian information criterion for model selection', *Sociological Methods & Research* **27**(3), 359–397.

Wood, S. A., Jiang, W. & Tanner, M. (2002), 'Bayesian mixture of splines for spatially adaptive nonparametric regression', *Biometrika* pp. 513–528.